Copy-Move Forgery Detection via Texture Description

Edoardo Ardizzone Alessandro Bruno Giuseppe Mazzola
Dipartimento di Ingegneria Informatica (DINFO), Università degli Studi di Palermo
Viale delle Scienze building 6, 90128, Palermo, Italy
phone: +390917028521, fax: + 390916598043,
ardizzon@unipa.it bruno@dinfo.unipa.it mazzola@dinfo.unipa.it
web: http://www.dinfo.unipa.it

ABSTRACT
Copy-move forgery is one of the most common type of tampering in digital images. Copy-moves are parts of the image that are copied and pasted onto another part of the same image. Detection methods in general use block-matching methods, which first divide the image into overlapping blocks and then extract features from each block, assuming similar blocks will yield similar features. In this paper we present a block-based approach which exploits texture as feature to be extracted from blocks. Our goal is to study if texture is well suited for the specific application, and to compare performance of several texture descriptors. Tests have been made on both uncompressed and JPEG compressed images.

Categories and Subject Descriptors
I.4.7 [**Image Processing and Computer Vision**]: Feature Measurement - *Texture*

General Terms
Algorithms, Performance, Experimentation.

Keywords
Image Forensics, Copy-Move Forgery, Texture descriptors.

1. INTRODUCTION
In today's digital age, the creation and manipulation of digital images, by using commercial software tools, is very simple and diffuse. As a result, it is very difficult to verify the authenticity and the integrity of a digital image. Digital Image Forensics deals with the problem of certifying the authenticity of a picture, or its origin, without explicit a priori information, e.g. using watermarks. Nevertheless the need of digital techniques for image authentication has been widely recognized, Digital Image Forensics is still a new research field in the Image Processing area. For a complete overview of the state of the art in Image Forensics see [1].

Three are the main branches in this research field:

- Image Source Identification, which aims to identify which device was used to capture an image (model or exemplar of scanner, of digital camera);
- Discrimination of computer generated images, to detect if an image is natural or synthetic;
- Image Forgery detection, to discern if an image has been intentionally modified by human intervention.

In our work we investigated the problem of detecting image forgeries. Images can be doctored in several ways: photo-compositing, re-touching, enhancing are only some examples of typical image alterations. Although many tampering operations generate no visual artifacts in the image, they will nevertheless affect its inherent statistics. See [2] for a classification of the most common tampering operations applied to a digital image. In particular we dealt with the problem of detecting copy-move forgeries[3], i.e. parts of an image that are copied and pasted onto the same image. The goal of our work is to test the ability of some standard texture descriptors to find similar areas into an image, as possible copy-move alterations. Both precision and efficiency are tested within our experiments.

The paper is organized as follows: section 2 discuss some state-of-the-art techniques: section 3 presents the overall proposed method; section 4 describes the tested texture descriptors; section 5 discuss the evaluation metrics we used in our experiments, which are presented and discussed in section 5; a conclusive section ends the paper.

2. COPY-MOVE FORGERY
Copy-Move[3] is one of the most common forgery used for to alter the content of an image, typically to delete objects from the scene, and to substitute information with some other from the same image. The most simple approach to solve this problem is exhaustive search, i.e. to compare an image with every cyclic-shifted version of itself. However, this approach is computationally very expensive and takes $(MN)^2$ steps for an image of size MxN. The most common approaches in literature are block-matching based. First, an image is divided into overlapping blocks. Then, some features are extracted from each block, and compared to find the most similar ones. At last, results are analyzed and decision is made only if there are several pairs of similar image blocks within the same distance. Several different features has been proposed in literature to search copy alterations within a block-matching based system. In [4], the authors proposed to use quantized Discrete Cosine Transform (DCT) coefficients for this purpose. The advantage is that the signal energy is concentrated on the first few coefficients, and operations such as noise addition, compression, and retouching should not affect these coefficients. Nevertheless, it does not work if duplicated regions are processed by geometrical transformation. Later on, Popescu et al.[5] used Principal Component Analysis (PCA) and Eigenvectors of the covariance matrix for an alternative representation of the blocks. The approach proved to

MiFor'10, October 29, 2010, Firenze, Italy.

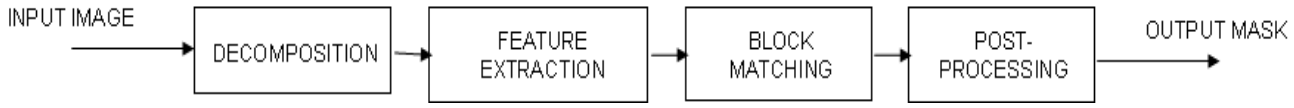


Fig. 1 Scheme of the overall method.

to be robust to compression and noise addition, however re-sampling (scaling, rotation) affects eigenvalues. In [6] Li et al. proposed to decompose the image using discrete wavelet transform (DWT) and singular value decomposition (SVD). Results are similar to those of [5], since SVD and PCA are very similar methods. In [7], Luo et al. proposed to use color information to represent blocks: average value of red, blue and green color components of the whole block and of 4 sub-blocks, divided according to different directions. Experimental results showed that this method was very robust to JPEG compression, Gaussian blurring and additive noise.

In this paper we exploit texture as feature to be extracted from blocks to be matched. The goal is to study if texture is suited for the specific application, and to compare performance of several texture descriptors.

3. OVERALL METHOD

The scheme of the overall approach is shown in fig.1.

Input image (size M x N) is first grayscaled, as in this work we are not interested in studying the influence of color properties. Image is then decomposed into overlapping square blocks (size B x B). The number of blocks to be analyzed is nB= (M-B+1) x (N-B+1).

We extract texture features from each block: 5 different texture descriptors are tested (see section 4). Each feature is represented as a vector.

Blocks are sorted according to vector's component which has the maximum variance along all the blocks. The sorted list of the blocks is then scanned in order to find similar blocks. Features from each block are compared to those of the next blocks in the sorted list, according two possible approaches:

- Fixed Window: WS =nB*P where P is a fixed parameter, that indicates the percentage of following blocks in the list to be considered.
- Adaptive window: blocks are compared up to those in the list which distance (for the key component) is below a threshold.

Even if the second approach is adaptive, the first one is faster, since matching time for matching is constant for all the blocks. Furthermore no significant differences in accuracy has been measured. Therefore we preferred the fixed window solution setting the value of WS as 1/1000 of the number of blocks.

As similarity criterion we used the relative error:

$$\left| \frac{V_i^j - V_i^k}{V_i^j} \right| < TH \qquad \forall i = 1..n \qquad (1)$$

between the corresponding components V_i^j and V_i^k of the feature vectors (size n) extracted from two blocks B_j and B_k. To make it symmetrical, instead of using (1), the relative error is computed as the ratio of the absolute error and the minimum value of the two components:

$$\left| \frac{V_i^j - V_i^k}{\min\left(V_i^j, V_i^k\right)} \right| < TH \qquad \forall i = 1..n \qquad (2)$$

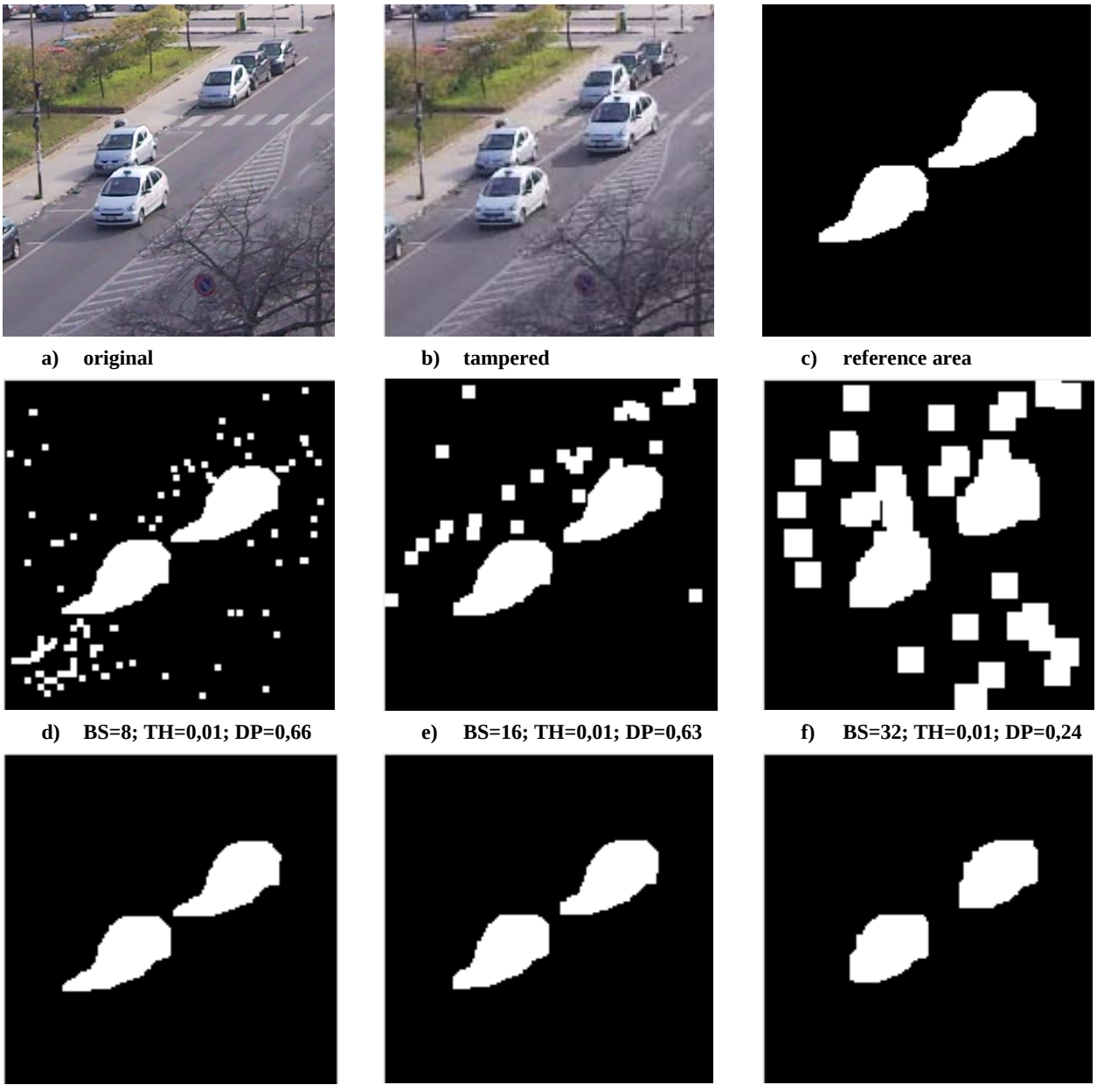
If all the relative errors are below a percentage *TH*, the two blocks are considered as candidate forgeries. We choose the relative error of each component, rather than Euclidean distance of the vectors, as in some cases, components have different order of magnitudes.

Then we compute the distance d_{jk} between spatial coordinates of matching blocks. Matching blocks that overlaps are considered false positives and discarded. To further reduce false positives, we select only pairs of blocks which are at the most frequent distance (MFD processing). In fact in a tampered image a copy is the translated version of a set of blocks from elsewhere into the image, and the same translation function is applied to all the copied blocks. Furthermore, in case of forgeries, duplicated areas are composed by several matching blocks. Then, among candidates, isolated blocks, that are not connected to any other blocks, are deleted from the output mask (IB processing). This general approach has been applied extracting 5 different texture descriptors.

4. TEXTURE DESCRIPTORS

Texture is one of the most studied image features in Computer Vision, Image Processing and Computer Graphics applications. It can be considered as a measure of the perceived image surface variations. In literature there are several works which propose[8] and evaluate[9,10] texture descriptors, for many different applications. For our purpose we test 5 different standard texture descriptors:

- Statistical: mean, standard deviation, skewness and kurtosis of the pixels grey values. Output is a 4-dimensional vector.
- Edge Histogram[11]: in our simplified version, we filter blocks with 4 directional (vertical, horizontal, 45, 135) and a non-directional Roberts-like operators. Mean of the filtered blocks are considered as descriptors. Descriptors is a 5-dimensional feature vector.
- Tamura[12] descriptors: Contrast, Coarseness and Directionality properties from the Tamura set of features. Output a 3-dimensional feature vector.
- Gabor[13] descriptors: a bank of Gabor filters (2 scale and 4 orientations) is applied to blocks. Mean and standard deviation of the coefficients from each sub-band are calculated to form a 16- dimensional texture feature.
- Haralick[14] Descriptors: The Haralick descriptors are based on statistical moments and obtained from co-occurrence matrix. We use as descriptors correlation, energy, contrast, homogeneity, resulting in 4-dimensional vector.



a) original **b) tampered** **c) reference area**

d) BS=8; TH=0,01; DP=0,66 **e) BS=16; TH=0,01; DP=0,63** **f) BS=32; TH=0,01; DP=0,24**

g) BS=8; TH=0,01; DP=0,95 **h) BS=8; TH=0,01; DP=0,94** **i) BS=8; TH=0,01; DP=0,84**

Fig. 2 Example results: Statistical Descriptor. After matching (d-f) and after post-processing (MFD and IB processing) (g-i). BS=block size, TH=threshold, DP=Detection Precision

5. EVALUATION METRICS

To evaluate the precision of the method, the source and the destination areas of every copy moves are saved as a binary mask: we called this area A_R (reference area, fig. 2.c). The result of the detection method is an output mask A_D (detected area, fig.2.d-i)

Better results as much A_D is similar to A_R. In particular we measured detection precision (DP) as follows:

$$R = \frac{n(A_D \cap A_R)}{n(A_R)} \quad (3)$$

$$P = \frac{n(A_D \cap A_R)}{n(A_D)} \quad (4)$$

$$DP = R \cdot P \quad (5)$$

where:

- *R* is the recall, i.e the ratio of the number of pixels in the intersection of the detected area A_D and the reference area A_R, and the number of pixels in A_R. When it tends to 1, A_D covers the whole A_R, but nothing can be said about pixels outside A_R; if it tends to 0 A_D and A_R have smaller intersection;

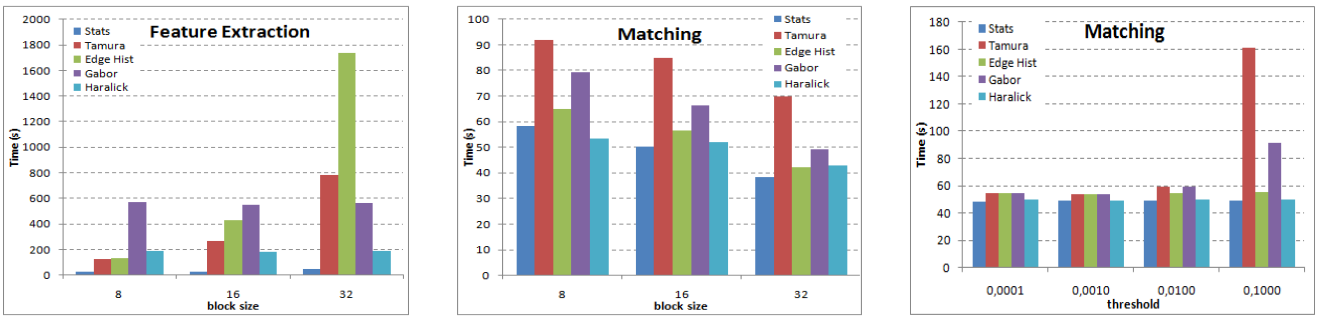


Fig.3 Average Execution Time: Feature Extraction vs block size (left); Matching vs block size (center) and vs threshold (right).

- *P* is the precision, i.e the ratio of the number of pixels in the intersection of the detected area A_D and the reference area A_R, and the number of pixels in A_D. When *P* tends to 0, the whole detected area has no intersection with the reference. If it tends to 1, fewer pixels of A_D are labeled outside A_R. Nevertheless this parameter will not assure that the whole reference area has been covered.

DP combines these two parameters: *DP* is high if A_D both covers A_R and has few outliers, and it is low if A_D and A_R are only partially overlapped or, when A_R is well covered, if A_D encloses many pixels which are not in A_R.

Varying block size two overlapping effects are observed:

- If the block size is larger than the tampered areas, $R \rightarrow 0$. In fact, whatever block we consider it will contain pixels which do not belong to the tampered area. Therefore no matches can be found.

- If block size is small $P \rightarrow 0$, because the probability to have natural similarities in an image increases, so that the number of false positives.

6. EXPERIMENTAL RESULTS

The overall method has been implemented in Matlab and executed on an Intel Core i7 PC (4 CPU, 1.6 GHz per processor, 4 GB RAM). We exploit the Matlab parallel library to make 4 workers run simultaneously.

Our dataset is made of 20 400x400 tampered uncompressed images, into which areas are copied and pasted onto other parts of the same image. For each test we measured both accuracy and execution time. Tests are also repeated for JPEG-compressed versions of the images (70% quality) to study how compression influences the detection ability of the analyzed descriptors.

Fig.2 shows some results obtained with the Statistical descriptor, varying block size and with a fixed threshold value. Note that after matching (2.d-f), DP decreases when block size increases, since larger false positives are detected, and the method does not correctly detect contours of the copies, in correspondence of fine details of the reference area. After post-processing (2.g-i), results with block size 8 and 16 are very similar, while 32 is a too larger value for this input image, and object contour is roughly detected.

Figure 3 shows average execution time for two of the steps of the method: feature extraction and matching (post-processing time is negligible with respect of matching).

With regards of feature extraction:

- Statistical descriptor is the fastest, and time slightly increases with block size.

- Edge Histogram and Tamura descriptors are quite fast for small blocks while time strongly increases with block size.

- Gabor and Haralick feature extraction time does not depend on block size.

Regarding matching time, we show results (fig.3) versus block size (results averaged on thresholds) and versus threshold value (averaged on block sizes):

- It loosely depends on block size (it slightly decrease as the number of blocks decrease)

- Processing time increases when threshold value is higher, as the number of matches increases, particularly for Tamura and Gabor.

Figure 4 reports results about average detection precision results achieved with the 5 tested descriptors, for uncompressed (left) and JPEG-compressed(right) images.

Results after matching (purple, light blue lines and red in fig. 4), before post processing, show that:

- Statistical, Edge Histogram and Gabor descriptors give similar results: precision is very high (90%) for lower values of the threshold *Th* while it decreases when for higher values of *Th*, as some false positives are detected.

- Tamura and Haralick give bad results (acceptable results only for Tamura with small block size)

After post-processing (dark blue, yellow and violet):

- For small blocks all the methods give similar results, very high precision (even about 95%). Only Haralick gives a lower average precision (85%).

- Statistical, Haralick and Edge Histogram precisions are practically independent of threshold value, within the tested range of values.

- Gabor and Tamura precisions decrease for higher threshold values, as some false positives are still detected.

For JPEG-compressed images (quality 70%):

- After matching, before post-processing, no good results with any of the descriptors, within the selected range of thresholds.

- After post-processing, we have bad results with lower values of the threshold, and some results are achieved only with 0,1. Acceptable results with Statistical (80%) and Edge Histogram (70% with block size 16 and 32).

-Higher values of the threshold do not improve precision, as the number of false positives strongly increases.

- For higher compression ratio, (e.g. 50%) the method does not give acceptable results, not even using best descriptors.

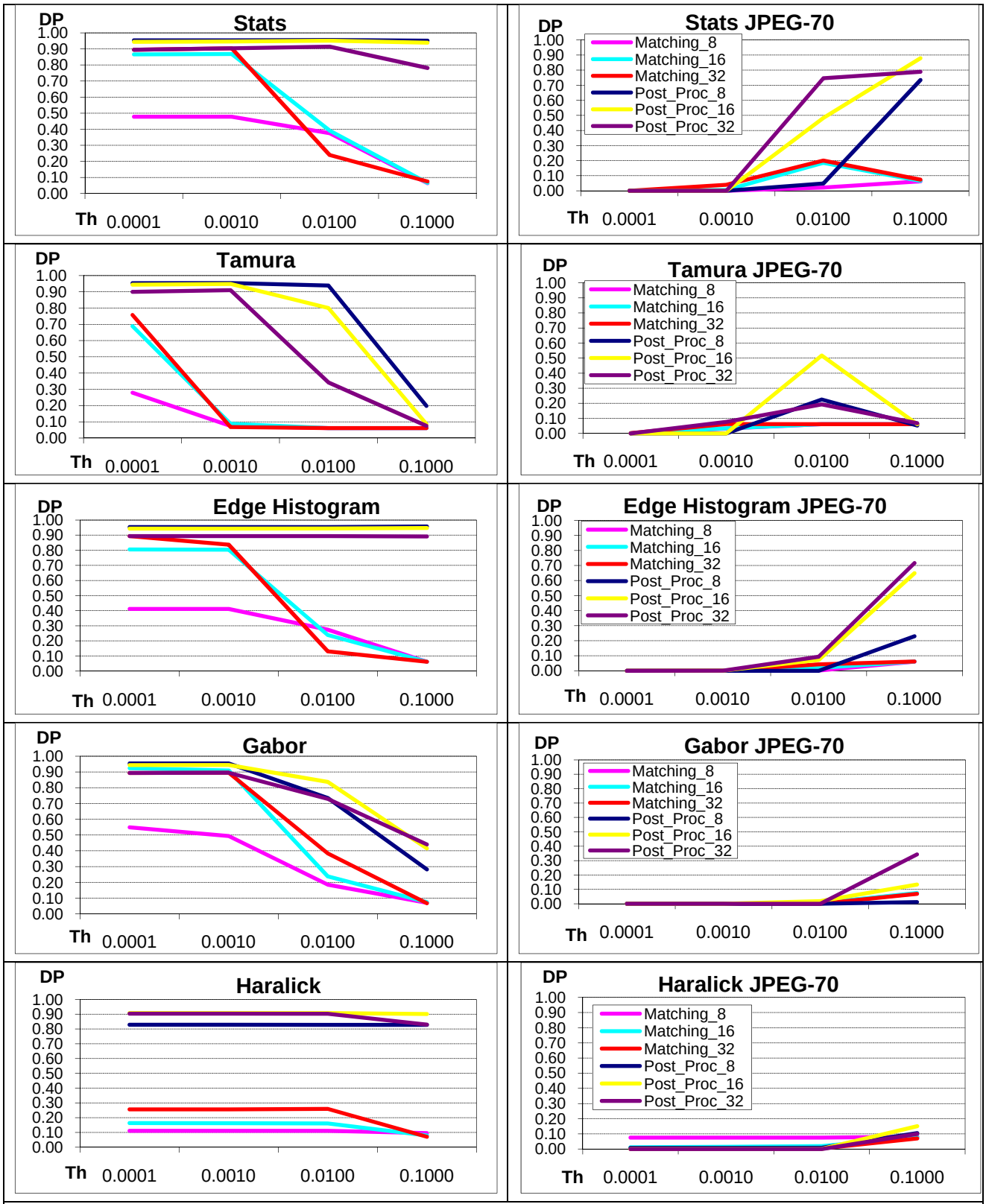


Fig 4 Average Detection Precision (DP) vs Threshold (Th). Lines represents results measured after matching and after post-processing with different values of block sizes (8,16,32). (for clarity, table callouts are shown only on the right).

7. CONCLUSIONS

In this paper we studied the ability of some standard texture descriptors to detect copies in tampered images. We used a common framework to test descriptors: a block matching approach and a post-processing step, to filter out false positives.

Experiments showed that the simplest descriptor (the Statistical) is that giving the best results in terms of precision versus execution time. Edge Histogram gives good results too, in case of small block size.

We tested our system also on JPEG compressed images and we observed that Statistical descriptor and Edge Histogram give still the best results, but setting higher values for the threshold parameter in the matching process.

In our experiments we intentionally ignored color properties, as our goal was to test only texture as relevant feature for our application. Furthermore, block matching methods are not applicable when copies are processed by geometrical transformations. In our future works we plan to compare results achieved with texture descriptors, with those obtained using other image features (color, shape), and to combine them within a single framework.

REFERENCES

[1] Sencar, T., Memon N. 2008, Overview of State-of-the-art in Digital Image Forensics. World Scientific Press

[2] Farid H. 2009, A Survey of Image Forgery Detection, IEEE Signal Processing Magazine, 26(2):16-25

[3] Bayram, S., Sencar, T., and Memon N. 2008, A Survey of Copy-Move Forgery Detection Techniques, IEEE Western New York Image Processing Workshop, Sept. 2008, NY

[4] Fridrich, J., Soukal, D., and Luk, J., 2003, Detection of Copymove Forgery in Digital Images, Proc. Digital Forensic Research Workshop, *Cleveland, OH*, August 2003.

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

[6] Li, G., Wu, Q., Dan Tu and Sun S., 2007, A Sorted Neighborhood Approach for Detecting Duplicated Regions in Image Forgeries Based on DWT and SVD, Proceedings of the 2007 IEEE International Conference on Multimedia and Expo, ICME 2007, 1750-1753.

[7] Luo, W., Huang, J., and Qiu, G. 2006. Robust Detection of Region-Duplication Forgery in Digital Image. In Proceedings of the 18th international Conference on Pattern Recognition - Volume 04, 746-749.

[8] T. Sikora, 2001, The MPEG-7 visual standard for content description—an overview, IEEE Trans. Circuits Syst. Video Technol. 11 (6), pp. 696–702.

[9] Feng Xu, Yu-Jin Zhang, 2006, Evaluation and comparison of texture descriptors proposed in MPEG-7, Journal of Visual Communication and Image Representation, Volume 17, Issue 4, August 2006, Pages 701-716

[10] Howarth P. and Ruger S., 2004), Evaluation of Texture Features for Content-Based Image Retrieval, Proc. of CIVR 2004, LNCS 3315, pp. 326-334

[11] Won, C.S., Park, D.K., 2002, Efficient Use of MPEG-7 Edge Histogram Descriptor. ETRI. Journal 24, 23–30.

[12] H. Tamura, S. Mori, T. Yamawaki, Texture features corresponding to visual perception, IEEE Trans. Syst. Man Cybern. 8 (6) (1978) 460–473.

[13] Jain A.K. and Farrokhia F. 1991, Unsupervised Texture Segmentation Using Gabor Filters, Pattern Recognition, Vol. 24, No. 12, pp. 1167-86.

[14] Robert M. Haralick, 1979, Statistical and structural approaches to texture, Proc. IEEE, vol. 67, no. 5, pp. 786-804.