Copy–Move Forgery Detection by Matching Triangles of Keypoints

Edoardo Ardizzone, Alessandro Bruno, and Giuseppe Mazzola

***Abstract*— Copy–move forgery is one of the most common types of tampering for digital images. Detection methods generally use block-matching approaches, which first divide the image into overlapping blocks and then extract and compare features to find similar ones, or point-based approaches, in which relevant keypoints are extracted and matched to each other to find similar areas. In this paper, we present a very novel hybrid approach, which compares triangles rather than blocks, or single points. Interest points are extracted from the image, and objects are modeled as a set of connected triangles built onto these points. Triangles are matched according to their shapes (inner angles), their content (color information), and the local feature vectors extracted onto the vertices of the triangles. Our methods are designed to be robust to geometric transformations. Results are compared with a state-of-the-art block matching method and a point-based method. Furthermore, our data set is available for use by academic researchers.**

***Index Terms*— Digital image forensics, copy-move forgery, SIFT, SURF, Harris, Delaunay triangulation.**

I. Introduction and State of the Art

DIGITAL Image Forensics deals with the problem of certifying the authenticity of a picture, or its origin. An image has always implied the truth of what it represents. The advent of digital pictures and relative ease of digital image processing makes today this authenticity uncertain. The same tools, used to crop an image, eliminate "red-eye" or simply improve an image, can also be used to doctor images with despicable intent, creating an image that is not a representation of the reality.

Digital images can be manipulated in such a perfect way that the forgery cannot be visually perceived by naked eye. Nowadays, in our society, we can come in contact with a lot of tampered images, in news report, business, law, military affairs, academic research. More particularly, tampered images could be used to distort the truth in news reports, to destroy someone's reputation and privacy, e.g. by changing a face of a person in a photo with someone else's face. Law enforcement today uses emerging technological advances in the investigation of crimes. In fact Image Forensics techniques are used mainly when an image is presented as an official proof to influence the judgment. During last decade different techniques for validating the integrity of digital images have been developed. Three are three main branches [1] in Digital Image Forensics: *Image Source Identification*, that 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.

The techniques used to verify the authenticity of an image can be further divided into two major groups: intrusive and non-intrusive. In intrusive (active) techniques, some sort of signature (watermark, extrinsic fingerprint) is embedded into a digital image, and authenticity is established by verifying if the retrieved signature matches the original one, or if it is corrupted. The use of active methods is limited, due to the inability of many digital cameras and video recorders to embed extrinsic fingerprints. Passive techniques use the intrinsic content of an image to detect if it has been tampered, without any superimposed information.

One of the main objectives of Image Forensics techniques is to understand what kind of tampering has been applied. Images can be doctored in several ways [2]: 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.

In this work we particularly intensified the study of copy-move tampering [3], that is one of the most common image manipulations. The goal of copy move forgery is to replicate a part of an image, often to hide an object, by copy-pasting a set of pixels from an area to another area of the same picture, and it is often very difficult to detect with the naked eye.

The most simple approach to detect if an image has been tampered by a copy-move forgery 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 $M \times N$, and does not work when the copy-pasted object is modified by geometric transformations (rotation, scaling, distortion) [4]. Copy move detection methods can be roughly divided into two main groups: block-matching and point-matching.

In the first family of techniques an image is divided into overlapping blocks, some features are extracted from each block, and compared to each other to find the most

Manuscript received November 14, 2014; revised March 9, 2015 and May 29, 2015; accepted May 31, 2015. Date of publication June 15, 2015; date of current version August 6, 2015. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Husrev T. Sencar.

The authors are with the Dipartimento di Ingegneria Chimica, Gestionale, Informatica e Meccanica, University of Palermo, Palermo 90128, Italy (e-mail: edoardo.ardizzone@unipa.it; alessandro.bruno15@unipa.it; giuseppe.mazzola@unipa.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2015.2445742

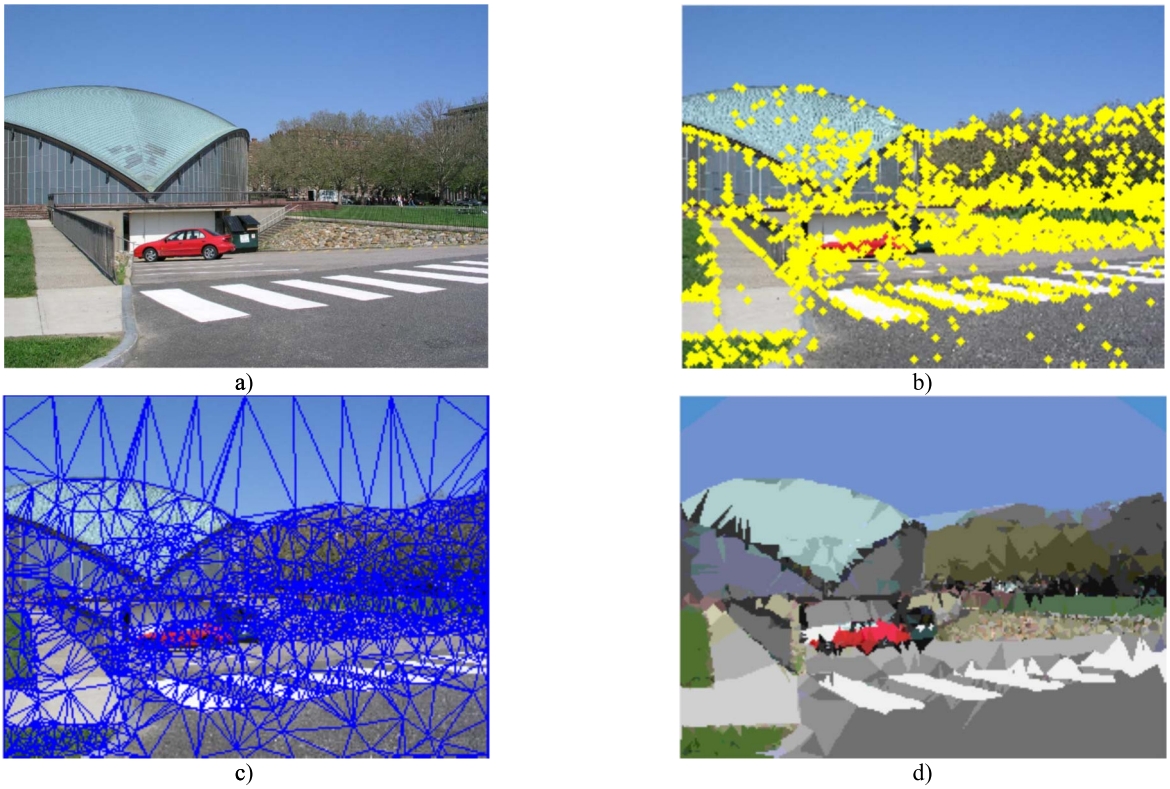

Fig. 1. Example of segmentation with our method. Original image (a), with superimposed SIFT points (b), with Delaunay triangulation (c) and the segmented image, in which each triangle is filled with its first dominant color (d).

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 have been proposed in literature to search copy alterations within a block-matching based system: Discrete Cosine Transform (DCT) coefficients [4], Principal Component Analysis (PCA) and Eigenvectors of the covariance matrix, [5], Discrete Wavelet Transform and Singular Value Decomposition [6], color information [7], Fourier Mellin descriptors [8]. In [9] Ryu et al. proposed a method which is based on Zernike moments, who showed to be robust against several attacks. In one of our previous works [10], we studied the performance of several texture descriptors in a block matching pipeline to detect copy move forgeries. Block-matching approaches, depending on the selected features, are typically robust to noise addiction, compression, filtering, but lack of robustness against geometrical attacks (rotation, scaling, distortion).

The point based approaches extract interest points and use local features, rather than blocks, to identify duplicated regions [11]. In particular SIFT [12]–[15] and SURF [15], [16] detectors are used to find points of interest, and the related local descriptors are used to find matches between these points. To eliminate the false matches, they are typically filtered by using post-processing techniques, as RANSAC. RANSAC is also used [12] to estimate the geometric transformation applied to the copy-moved area. Some of these works [12]–[14] use clustering methods to find groups of points that match, rather than single points, in order to search for the "structure" of the cloned objects. Point-based approaches proved to be robust to geometric transformation (rotation and scaling), but do not work if homogenous areas are used to hide an object, as keypoints cannot be extracted from those areas. An interesting work by Christlein et al. [18] compares and evaluates the results obtained with different approaches to the problem of copy move forgery detection.

In this paper we propose a very new approach that is based on the analysis of triangles of local keypoints. Matching between triangles is done both extracting inner features of the triangles (color), analyzing their geometrical properties (angles) and the feature of the vertices that compose the triangles (local descriptors). Our methods can be considered "hybrid", as they are halfway between the block-based and the point-based ones. The paper is organized as follows: in section 2 we describe the proposed methods; in section 3 we discuss the experimental results and we compare our methods with two reference techniques; a conclusive section ends the paper.

II. Proposed Approach

The idea behind our approach is very simple: all the objects in a scene may be represented as a set of connected triangles. This model is very used in Computer Graphics, and we decided to apply it to our set of 2D images. We first extract points of interest from an image, using three of the most common detectors (SIFT [19], SURF [20] and Harris [21]) (see fig. 1.b). A Delaunay triangulation [22] (fig. 1.c) is built onto the extracted points. Image is therefore subdivided into triangles, which include pixels with very similar features (fig. 1.d).

We decided to use the Delaunay triangulation, instead of the Voronoi tessellation, as its atomic element typically does not include edges of the objects and its content may be considered as homogenous. Furthermore, to include also the outer parts of the image, where typically no interest points are extracted, we added uniformly arbitrary points onto the borders of the image (fig. 1.c). This solution does not influence the triangle mesh construction onto the extracted keypoints, but helps us to subdivide into triangles also the parts of the image that are near the vertical or the horizontal borders.

In this paper we present two methods that use this model to represent the objects: in the first method we analyze the inner content of the triangles (color) and their geometric properties (areas and angles); the second method analyzes the properties of the vertices that form the triangles, that are the points of interest of the image.

A. Triangles Matching by Colors and Angles

We extract from each triangle the first n dominant colors: each color channel is quantized into b bins and a 3D histogram is built with the pixels of the triangle. The n most frequent values of the histogram are taken as the dominant colors of that triangle (n and b will be further discussed in section III). Each triangle is represented by 3*n values (n values per channel). For our purposes information about color frequencies is discarded. We further compute triangle areas and inner angles. Angles are taken in counterclockwise order starting from the maximum one. This solution helps us to make our method robust to affine transformations, as discussed below. The input image is finally segmented into triangles (see fig. 1.d) which are described by their dominant colors, their areas and their ordered sequence of angles. To find possible copy-moved regions, we search for similar triangles into the image, analyzing two different features: colors and angles. First, triangles are sorted according to the L1 norm of their color vectors. The sorted list of triangles is then scanned and the features of each triangle are compared to the next triangles in the list, within a fixed window (a percentage of the number of triangles). An adaptive window approach, in which triangles are compared up to those in the list which distance is below a threshold, proved to be slower than the fixed window approach, without improving results.

If both the Sum of the Absolute Deviation (SAD) of the color vectors and of the angles are below a threshold, the two triangles are considered similar. If j and k (k>j) are the indexes of the two triangles to be compared:

$$\begin{aligned} &\sum_{i=1}^{3.n} |C_i^j - C_i^k| <= TH^c \\ &\qquad\qquad (k-j) < w_s \\ &\sum_{l=1}^{3} |a_i^j - a_l^k| <= TH^a \end{aligned} \tag{1}$$

where w_s is the fixed window size (computed as a percentage of the number of triangles), C is the color vector (made of 3*n values), a_i are the angles (see fig. 2) in radians (in which angles are sorted as described above), TH^c and TH^a are two thresholds, (see section III). When comparing the sorted angles, two triangles may match even in case of rotation or scaling. In fact the inner angles of similar triangles are the same, even if one of the two triangles is rotated or scaled with respect to the other one, if the angles are taken in the right order. Thus our method is designed to find copied objects also in case of geometric transformation.

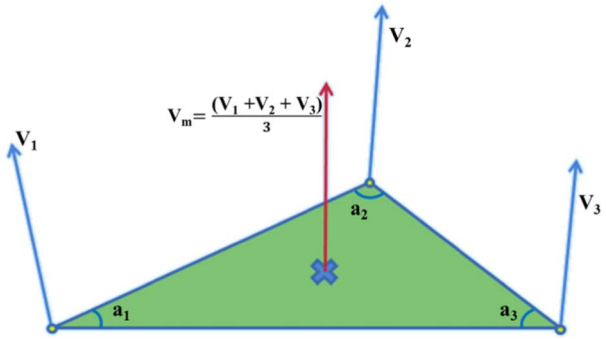

Fig. 2. A visual scheme of Delaunay Triangle of interest points. Each Triangle is represented by Dominant Color, Inner Angles (α_1, α_2, α_3), Descriptor Vectors (V_1, V_2, V_3) and Mean Vector (V_m).

To reduce false positives, we compare two triangles j and k only if the ratio between their areas:

$$r_A = \frac{\min(A_j, A_k)}{\max(A_j, A_k)} >= 0.25 \tag{2}$$

This solution limits the maximum detectable scale (to 2), but deletes the 20-25% of wrong matches. After the matching process, we have a list of pairs of triangles. To further delete false matches, we compute the centroids of these triangles, and we apply RANSAC (RANdom SAmple Consensus [23]) to the set of matching centroids, to select a set of inliers that are compatible with a homographic transformation. If less than 4 matches are found, RANSAC cannot be applied, and the match is considered "not reliable".

B. Triangles Matching by Mean Vertex Descriptors

For each triangle, we compute the Mean Vertex Descriptor (MVD) as the average value of the feature vectors (see fig. 2) extracted onto the geometric vertices of the triangles. For each triangle the Mean Vertex Descriptor V_{mi} is obtained as:

$$V_{mi} = \frac{V_{1i} + V_{2i} + V_{3i}}{3} \tag{3}$$

where V_j =1...3, i=1…N is the feature vector extracted onto the geometric vertices of the triangles and N is the number of the Delaunay Triangles inside of the image.

In this case we consider only the SIFT and the SURF algorithms, as there is no standard descriptor associated to the Harris corner points. The mean vector is a n-dimensional vector, where n is equal to 128 if interest points are extracted by the SIFT algorithm and is equal to 64 if interest points are extracted by the SURF algorithm.

To find possible tampered regions, we first sort the triangles according to the L1 norm of their MVDs and the MVD of each triangle is compared to the next ones in the list, within a fixed

window of size w_s (a percentage of the number of triangles). Also in this case a fixed window approach is preferable to an adaptive window one. Two triangles match if the L1 distance of their corresponding MVD is lower than a threshold.

If j and k (k>j) are the indexes of two triangles to be compared and V_{mj}, V_{mk} are the corresponding MVDs:

$$|V_{mj} - V_{mk}| \leq TH^v \quad (k - j) < w_s \tag{4}$$

where TH^v is a threshold, w_s is the fixed window size (see eq.1). As discussed in the section II.A, a RANSAC filter is applied to the geometric centroids of the matching triangles, to filter out false matches.

III. Experimentation and Evaluation

For our experiments we created our own dataset, which is available at our group website [24] (or by contacting the authors by email) and we mainly focused on the results at pixel level. We also evaluated the results onto the dataset created by Christlein et al. [18]. With respect of this one, our dataset investigates a wider range of possible geometric attacks. In the next sections we describe our dataset, discuss the reference methods we used for comparison and the evaluation metrics. Finally, the experimental results are presented.

A. Dataset

It is made of medium sized images (almost all 1000×700 or 700×1000) and it is subdivided into several datasets. The first dataset D0 is made of 50 not compressed images with simply translated copies. We used this dataset for tuning the parameters of our method. For the other two groups of images (D1, D2) we selected 20 not compressed images, representing simple scenes (single object, simple background), rather than complex scenes, as we are interested in studying primarily the robustness against some specific attacks. Single subject image are very common in artistic digital photography, whenever a photographer wants to highlight a detail of scene, and are used also for visual saliency analysis studies [25], [26].

The dataset D1 has been created by copy-pasting objects after rotation, D2 applying scaling to the copies. Each dataset has been further subdivided into subsets. The first subset D1.1 has been created applying to the copies 11 different types of rotation around the angle zero in the range of $[-25°, 25°]$ with step 5°. The second subset D1.2 is created by rotating the copies by 12 different angles in the range of [0°, 360°] with a step of 30°. The third subset D1.3 is built by rotating the copies by 11 different angles in the range of $[-5°, 5°]$ with a step of 1°. D1 is then composed by 680 images (with some repetitions) The subset D2.1 is obtained by scaling the copies by 8 different scaling factors in the range of [0.25, 2] with step 0.25. In D2.2 copies are scaled by 11 scaling factors in the range of [0.75, 1.25] with step 0.05. D2 is then composed of 380 images (with some intersections). We furthermore tested our approaches onto the 50 original images of the dataset D0 without tampering (subset D3), to study the ability to discriminate between tampered and not tampered images.

B. Reference Methods

Our proposed methods are hybrid as they are not fully point based and they share some aspects of the block based ones, then in our experiments we decide to compare our results with two different techniques: a block based approach and a point based method. The first technique [9] is based on the Zernike moments (in the rest of the paper simply indicated as ZERNIKE), that according to [18], is the most robust approach to several attacks. The reference point based approach [11] extracts the interest points of the image, compares each keypoint with all the other keypoints and then filters out the false positives by using RANSAC. In our experiments we tested two different versions of this algorithm (SIFT and SURF), which will be indicated in the rest of the paper as SIFT Point and SURF Point. The methods described in section II.A will be indicated in the rest of the paper, as SIFT, SURF, Harris Angle. The method described in section II.B will be indicated as SIFT, SURF Vertex.

C. Evaluation at Pixel Level

To evaluate results at a pixel level, we used three different metrics: Recall, Precision and Link Precision. To compute the first two metrics, we saved the source and the destination areas of every copy moves as binary masks, of the whole dataset. The "reference area" A_R is our groundtruth. The detected area A_D is the output mask of the methods, created as a binary mask in which all the pixels that are inside the matching triangles are set to 1 (see fig. 3). We compute the Precision and the Recall of the results as follows:

$$R = \frac{n(A_D \cap A_R)}{n(A_R)} \tag{5}$$

$$P = \frac{n(A_D \cap A_R)}{n(A_R)} \tag{6}$$

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 we have no information about pixels outside A_R; if it tends to 0 A_D and A_R have a smaller intersection;
- 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.

The Link Precision is computed as the ratio of the number of correct matches and the total number of found matches, that are the "links" between the matching triangles. The first two metrics are used to compare our results with ZERNIKE, while Link Precision with SIFT and SURF point, as recall and precision cannot be computed in these cases.

First, we tried to find the best settings for our method, within the D0 dataset, by tuning some of the parameters:

- the number of bins to quantize color channel b=8 (fixed);
- the number of dominant colors (n=1..4);

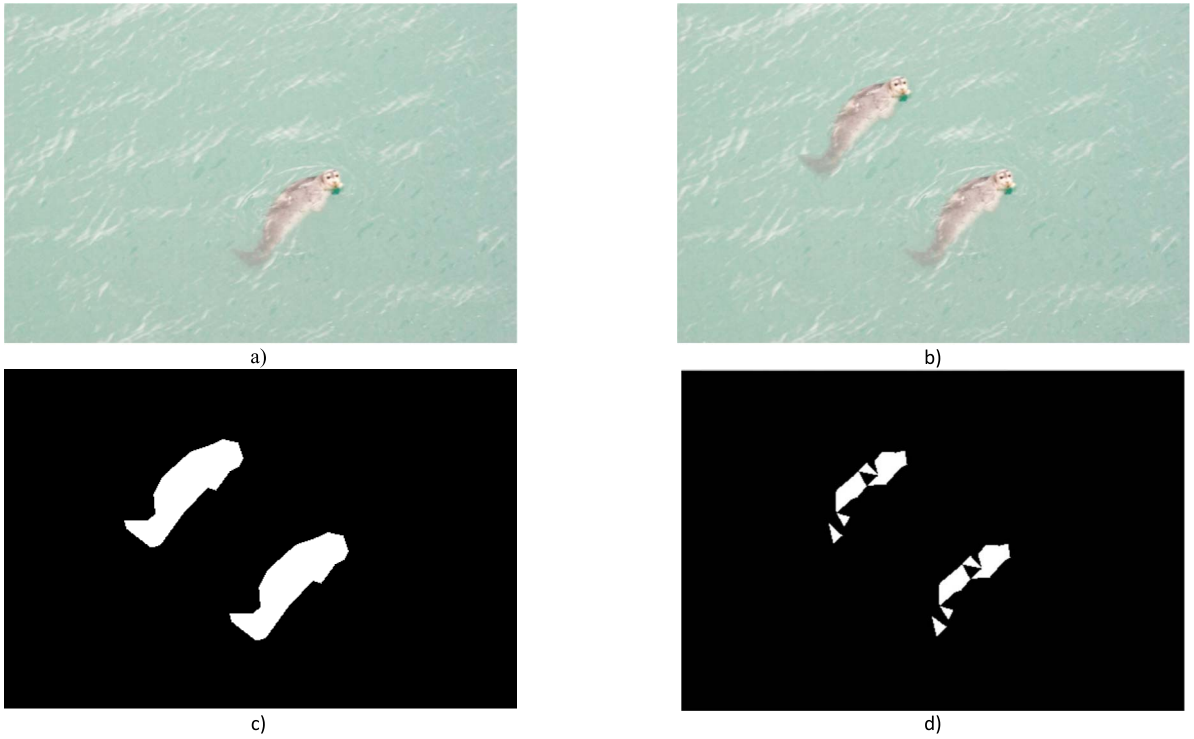

Fig. 3. A visual example of results evaluation method. Original image (a), tampered image (b), reference mask A_R (c) and output mask A_D with our method (d) for a simply translated copy.

- the color TH^c and the angle TH^a thresholds;
- the vertex threshold TH^v;
- the size of the fixed window $w_s=N_T/50$, where N_T is the number of triangles in an image.

We decided to compare only triangles that have exactly the same dominant colors ($TH^c= 0$), otherwise we measured too many false positives; w_s does not influence very much the results, then we decided to fix it to a value that is a good trade-off between precision and execution time. With respect to the number of colors, a good tradeoff between Precision and Recall is achieved with n°colors=4. The Link Precision is in practice independent of this parameter. In our experiment the optimal threshold value was 0.25 for both TH^a and TH^v and for all our methods.

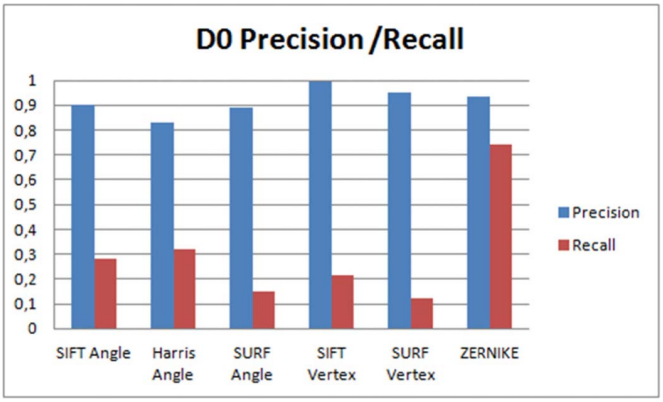

Fig. 4. Precision and Recall within the D0 Dataset.

D. Results at Pixel Level

Figure 4 shows the results, in terms of Precision and Recall, obtained within the D0 dataset with the optimal configuration of the parameters, after tuning them. Note that in our results we have a very high Precision (nearly the 100% with SIFT Vertex), and a low Recall value (around 30% for Harris Angle). This means that our method is very accurate in finding matches between triangles (very few false positives) but is not able to cover all the part of the copy-moved area (false negatives). Nevertheless in our tests we observed that copies, even when parts of the objects are not detected, are easily identifiable and distinguishable (see fig. 3.d). The ZERNIKE method, within the D0 dataset, has a very high Precision (94%, which however is lower than that of SIFT Vertex) and a high Recall (75%). Block based methods typically use a sliding "overlapping" window approach, i.e. blocks are analyzed shifting the window by 1 pixel per each iteration, while our methods search for non overlapping triangles. Therefore we expected block based methods to achieve higher recall values.

Fig. 5 shows results against rotation, with respect to ZERNIKE. Note that, in case of no rotation, while the precision of all the methods is comparable, the recall of the ZERNIKE method is much higher. This is not true if we apply rotation to the copies. Within the dataset D1.1 the precision and the recall of the ZERNIKE method drastically decrease outside $[-5°, 5°]$. The Vertex methods are, in practice, invariant to rotation (above all SIFT Vertex), while Angle methods have lower performances. Within the dataset D1.2, that includes all the possible rotations from 0° to 360° with a step of 30°, ZERNIKE method, as expected, does not

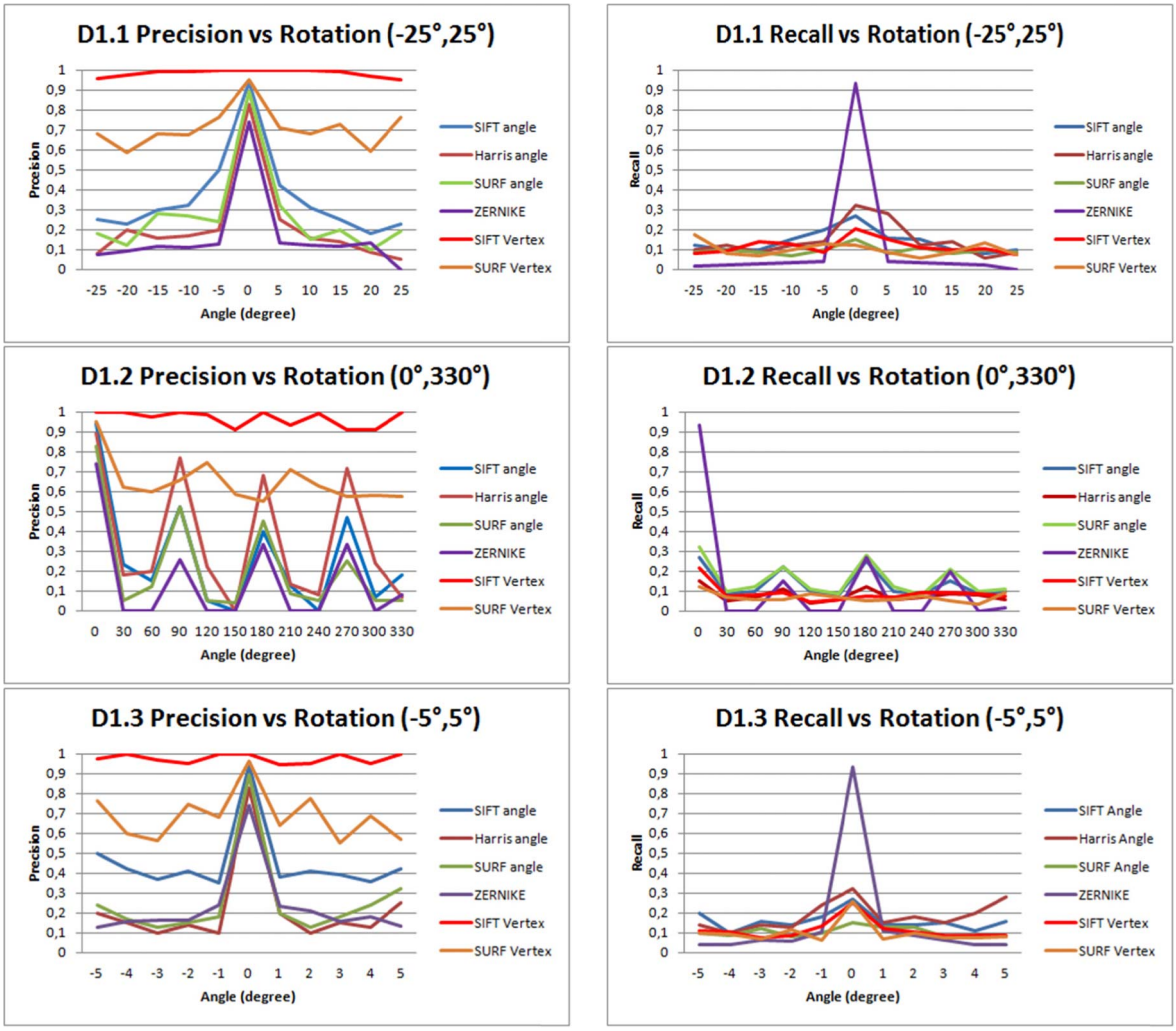

Fig. 5. Precision (left) and Recall (right) versus rotation with different angles.

work very well, neither in terms of precision nor of recall. Angle methods reach the highest values when the rotation angle is multiple of 90°. In these cases, in fact, triangles are less distorted and we can achieve a more accurate match. Vertex methods are invariant to rotation also in this case. Within the dataset D1.3 (rotation between $-5°$ and $5°$) the results are similar to those of the dataset D1.1. The robustness of our SIFT method against rotation derives to the intrinsic invariance to rotation of the SIFT keypoint extraction algorithm, therefore the SIFT Vertex method is preferable in these cases. SURF algorithm works very well only in case of small rotations.

In case of scaling (fig. 6), while the ZERNIKE method fails when the copy is resized by a factor higher than 5%, for the same reasons discussed for the rotation case, Angle methods achieve acceptable results up to a resize factor of 25%, while Vertex methods achieve the best results, especially in case of magnification. In summary, our methods give very good results in term of Precision (above all the SIFT Vertex), better than the block-based one, also when no geometrical transformations are applied. In terms of Recall, our methods try to estimate the area that has been copy-pasted, as block-based methods do, but as it is not a sliding window approach, only part of the copied area is detected. Nevertheless, block methods do not work in case of transformations, neither in terms of recall nor of precision, while our method is able to give as output some information about the copied pixels, with very few false positives.

Finally, ZERNIKE method does not achieve very good results, except for simple translated copies. This is more evident as, in our dataset, image backgrounds are regular, then block based approaches are more likely to find a lot of false matches outside the copied areas, resulting in a very lower precision. Furthermore, if there are a lot of false matches, blocks from the not tampered areas may be not filtered out by the post processing step, thus also the recall decreases.

When comparing our methods with the reference point-based approach (fig. 7), in terms of Link Precision, we note that the SIFT Vertex method achieves the same performances than the two reference methods (SIFT Point and SURF Point), for all the considered transformations, while the SURF Vertex and the Angle methods have worse results (only SIFT Angle results, between the Angle methods, are shown for the clarity of the figure).

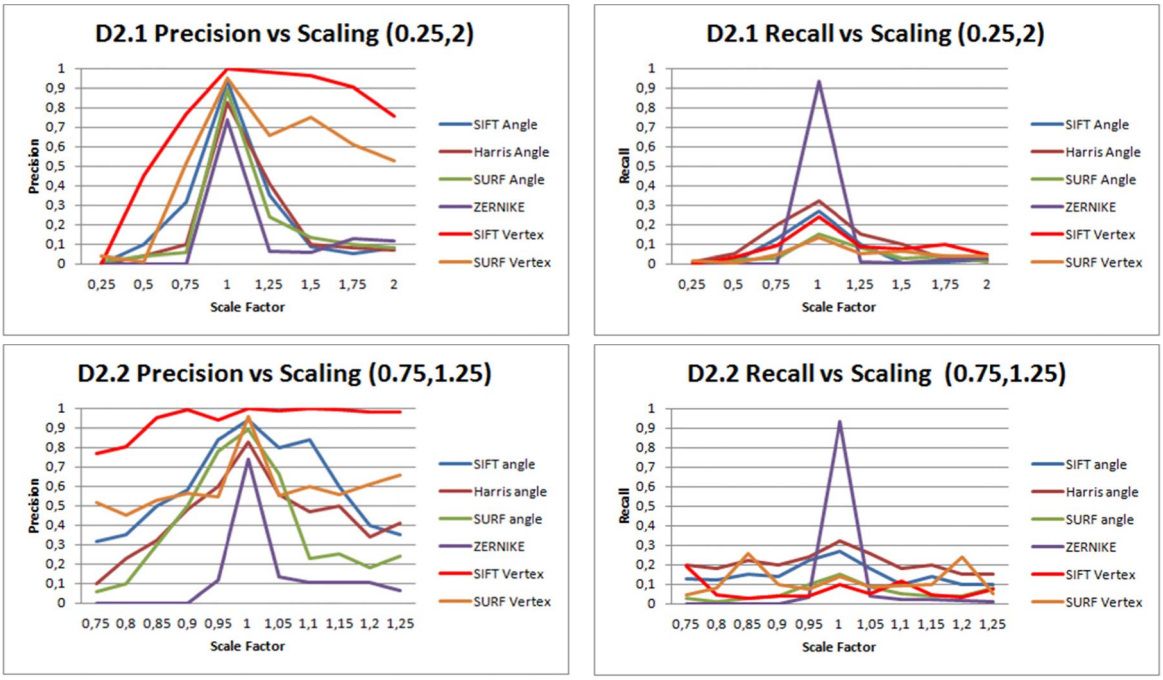

Fig. 6. Precision (left) and Recall (right) versus scaling with different scaling factors.

Finally, we repeated the tests for the images in the dataset with simply translated copies applying JPEG compression, and results did not sensibly change up to a compression factor of 50. In fact JPEG compression slightly affects edges and corners of an image, that are typically the points detected by SIFT, SURF and Harris algorithms as points of interest.

In terms of efficiency the feature extraction step takes almost the same time (few seconds) for all the methods (SURF is a little bit faster, ZERNIKE is the slowest), while time for the matching step (lexicographical sorting) is much higher for the ZERNIKE method (a tenth of minutes versus few seconds) as the number of blocks to be analyzed is much higher than the number of triangles, and it is lower for the Point methods. Our methods are, on the whole, slightly slower than the point based, as some seconds are spent to build the Delaunay triangulation. However, our method is still very efficient, as the time spent to analyze a single image, in our dataset, is always less than 10 seconds. Note that the execution time of our method depends on the number of triangles, i.e. the number of extracted keypoints. In figure 8 we show some visual examples of our results within the different datasets.

E. Results at Image Level

Even if we focused on the analysis of the results at pixel level, we also evaluated the results at image level, using the datasets D0 and D3.

The first one has been used to test the robustness of our method against false negatives, i.e. images with copy-moved parts that are not detected as tampered. All the tested methods achieved perfect performances, as no tampered images have been detected as original.

To measure the robustness against false positives, i.e. not tampered images detected as tampered, we used the dataset D3, made of the original 50 images of dataset D0. Test showed that our methods obtain very good results (no false positives). The reference point-based method is less robust (10% of incorrect detections). ZERNIKE has the worst results, in fact it detected 50% of false positives, but a good trade-off between false positives and false negatives strongly depends on the tuning of the parameters. This is due, as discussed in the previous section, to the fact that the images in our dataset have very regular backgrounds, then block-based methods are more likely to find a lot of matches also outside the tampered areas.

F. Further Experiments

We decided to test our method also within another publicly available dataset, proposed by Christlein et al. [18]. This dataset (D4) has been created starting from 48 images, containing complex scenes, which had been tampered by simple translated copies, rotated copies (within the range [0, 10] with step 2) and scaled copies (within the range [0.91, 1.09], step 0.02).

Table I and II show the results of our best method (SIFT vertex) within D4, in terms of recall, precision and link precision. The first remark is that the results of our method within D4 are worse with respect to those obtained within our dataset. This can be explained as the images in D4 are more complex and full of details, resulting in a much higher number of interest points, and a higher number of (smaller) triangles. Then, the probability to find matches outside the copied areas is higher, and the precision decreases. Furthermore, when our method finds good matches, they are typically between very small triangles. Therefore the part of the copy-pasted area which is detected is smaller, and also the recall decreases. The link precision decreases for the same reasons. At last,

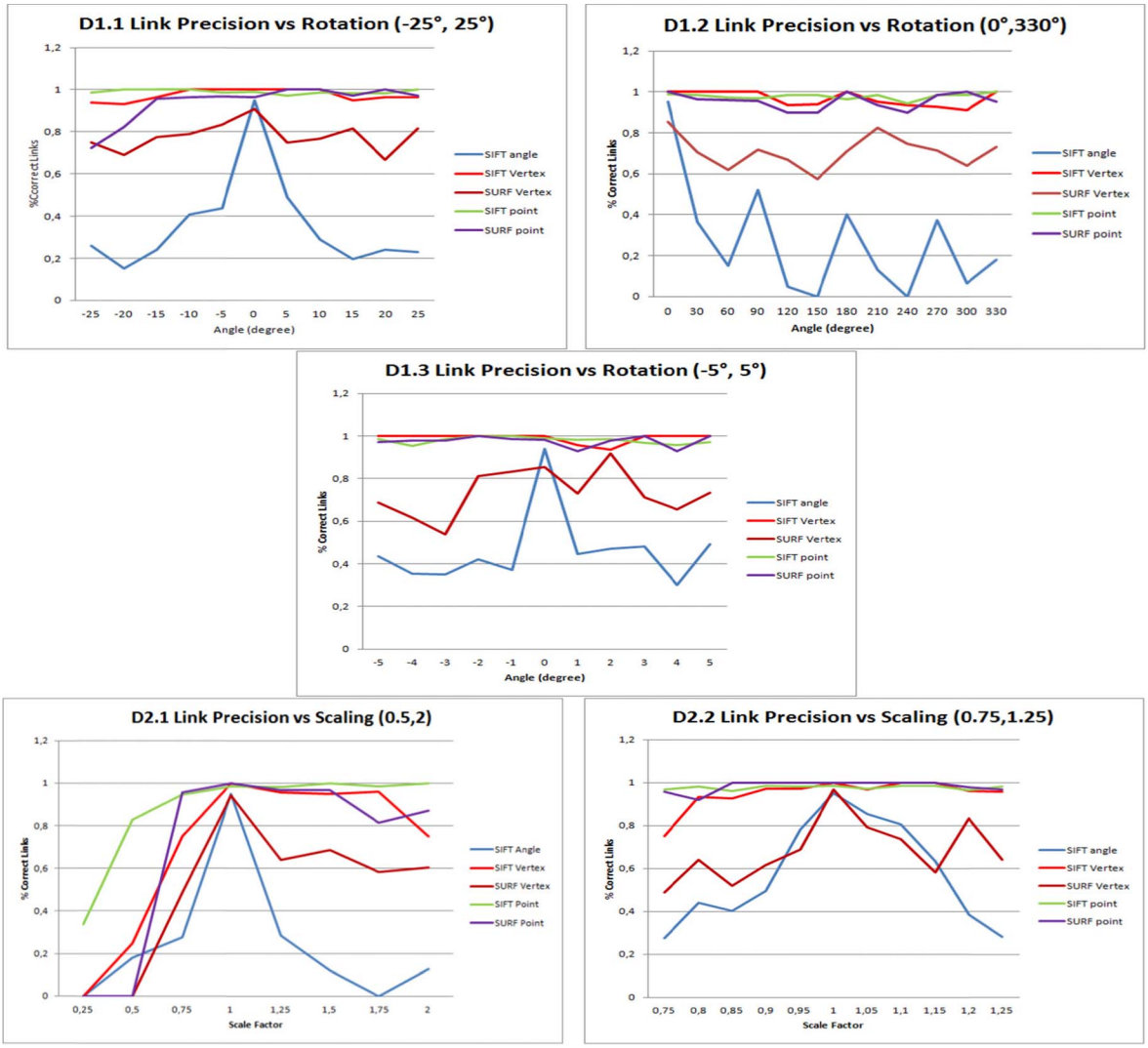

Fig. 7. Link Precision versus rotation (first two rows) and scaling (last row).

TABLE I
RESULTS WITHIN DATASET D4 VERSUS ROTATION

r	recall	precision	link_precision
0	0,052	0,617	0,674
2	0,005	0,393	0,433
4	0,004	0,382	0,425
6	0,004	0,425	0,468
8	0,004	0,382	0,419
10	0,004	0,373	0,412

the time spent to analyze an image increases, as it depends on the number of detected keypoints.

At image level we did not measure any false negatives, i.e. all the tampered images have been correctly classified. Some false positives (10% ca) are detected when our method is applied to the original images of the D4 dataset. It depends on the very high number of extracted keypoints and triangles. The results of many other approaches within this dataset are available in literature.

TABLE II
RESULTS WITHIN DATASET D4 VERSUS SCALING

s	recall	precision	link_precision
0,91	0,003	0,342	0,429
0,93	0,002	0,392	0,453
0,95	0,003	0,396	0,472
0,97	0,004	0,437	0,495
0,99	0,005	0,421	0,467
1	0,052	0,617	0,674
1,01	0,006	0,451	0,494
1,03	0,004	0,395	0,461
1,05	0,004	0,429	0,481
1,07	0,003	0,335	0,427
1,09	0,003	0,398	0,440

Finally, our proposed method performs better in case of simple scenes, as those in our dataset, where block based methods (above all) have worse results.

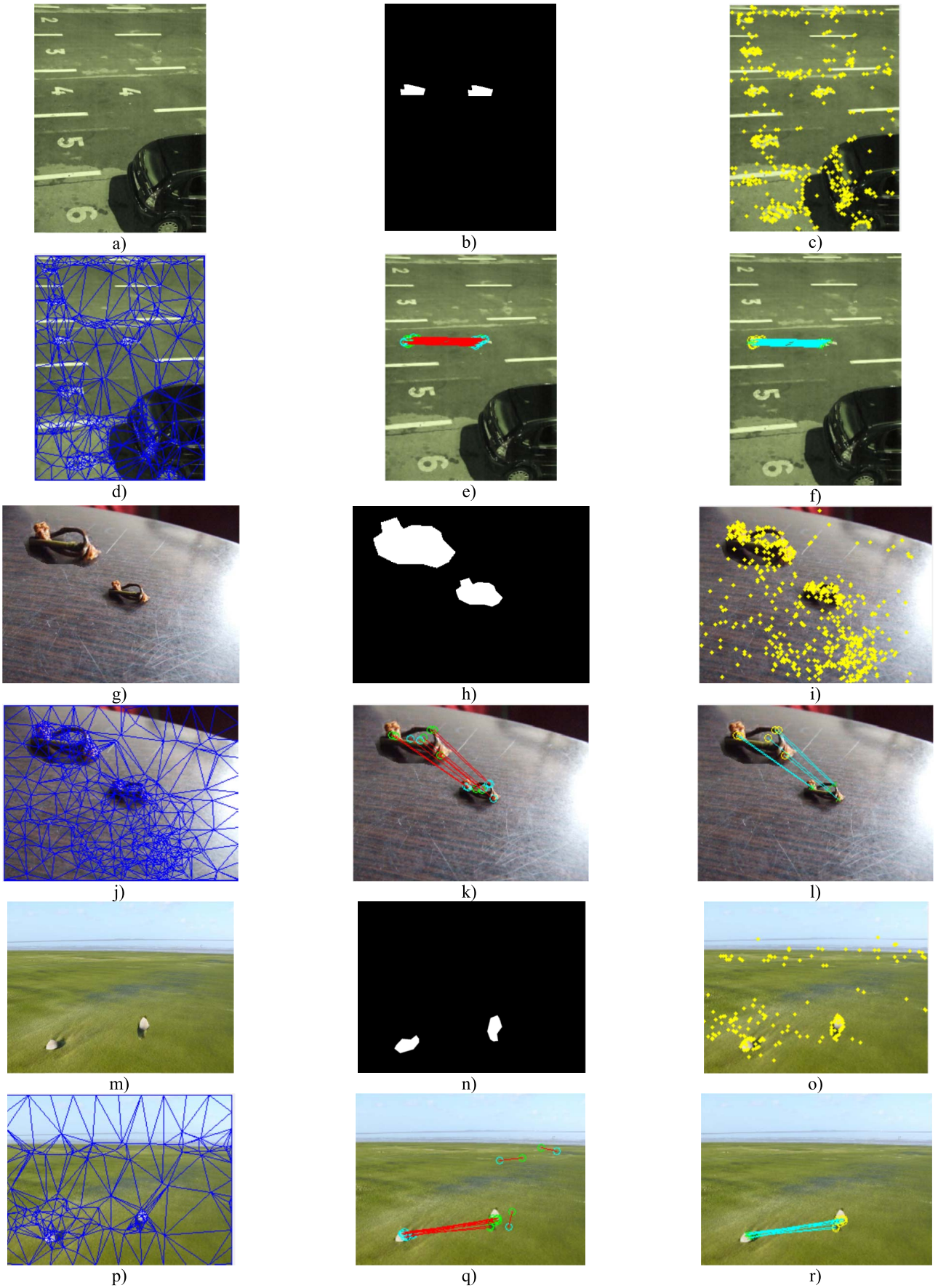

Fig. 8. Some visual results, with the SIFT Vertex method. Input Image (a, g, m), reference mask (b, h, n), SIFT points (c, i, o), Dealunay triangulation (d, j, p), matches before (e, k, q) and after RANSAC (f, l, r). The first example is from D0 (no geometric transformation, the second from D2.1 (scaling 1.75), the third from D.1.2 (rotation 240).

IV. Conclusion

Block matching methods are preferable in case of pure translation, as they reach pixel-wise precision, they give information about the copied pixels, they work also in case of homogeneous areas, when point-based cannot be used, but are extremely slow, and do not work well in case of geometric transformations. They have better performance in case of complex scenes, while for images with regular background they typically find a lot of false matches.

Point based methods achieve very good results, also in cases of geometrical transformations. They give only information about single (or groups of) points that are part of the copy-pasted area, but nothing about the pixels inside the copied areas. Therefore they can be used, with very good results, when tampering recognition is the goal, but cannot be used to detect the copy-pasted areas, unless a proper post-processing technique is used.

Our proposed methods are halfway between block and point based methods, and aim to analyze the structure of the objects in the scene, represented as a mesh of triangles. This is the major novelty of our work. Our methods can be used as well for copy-move recognition and detection, as they are able to find the presence of copy-moved areas and to expose parts of them. With respect to block based methods, our methods can find, with a very high precision, the tampered areas of the images, also in case of geometric transformations, but they are able to recover only parts of the pixels of the region, that are in most cases enough to detect the shape of the copied objects. On the other hand, our methods are two order of magnitude faster than block based ones. In comparison with point based ones, our methods have more or less the same performances, in terms of link precision, but have a lower number of false positives at the image level. This can be explained as we imposed tighter constraints with respect of the point based algorithm. In fact we search for structures that matches (e.g. triplets of points in case of SURF and SIFT vertex) rather than single points.

Our methods perform better in case of simple scenes, as the number of keypoints, and of triangles, is lower. In case of complex scenes the high number of detected triangles influences the matching process, resulting in worse performances. As well as the keypoint based approaches, our methods cannot be used if no interest points are detected, e.g. if homogeneous areas are used to hide object in a scene.

Furthermore the proposed methods can be used in the future to find copies also in case of some other type of transformations, e.g. anisotropic deformations, as we divide the object into its atomic elements, and each of them can be separately analyzed. We plan also to develop some post-processing techniques, to recover some missing matches, e.g. filling the holes between triangles, and to increase the recall of the methods.

References

[1] H. T. Sencar and N. Memon, "Overview of state-of-the-art in digital image forensics," *Algorithms, Archit. Inf. Syst. Secur.*, vol. 3, pp. 325–348, Dec. 2008.

[2] H. Farid, "Image forgery detection," *IEEE Signal Process. Mag.*, vol. 26, no. 2, pp. 16–25, Mar. 2009.

[3] B. L. Shivakumar and S. S. Baboo, "Detecting copy-move forgery in digital images: A survey and analysis of current methods," *Global J. Comput. Sci. Technol.*, vol. 10, no. 7, pp. 61–65, 2010.

[4] J. Fridrich, D. Soukal, and A. J. Lukáš, "Detection of copy-move forgery in digital images," in *Proc. Digit. Forensic Res. Workshop*, Cleveland, OH, USA, Aug. 2003, pp. 342–358.

[5] A. C. Popescu and H. Farid, "Exposing digital forgeries by detecting duplicated image regions," Dept. Comput. Sci., Dartmouth College, Hanover, NH, USA, Tech. Rep. TR2004-515, 2004.

[6] G. Li, Q. Wu, D. Tu, and S. Sun, "A sorted neighborhood approach for detecting duplicated regions in image forgeries based on DWT and SVD," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2007, pp. 1750–1753.

[7] W. Luo, J. Huang, and G. Qiu, "Robust detection of region-duplication forgery in digital image," in *Proc. 18th Int. Conf. Pattern Recognit.*, 2006, pp. 746–749.

[8] W. Li and N. Yu, "Rotation robust detection of copy-move forgery," in *Proc. 17th IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2010, pp. 2113–2116.

[9] S.-J. Ryu, M.-J. Lee, and H.-K. Lee, "Detection of copy-rotate-move forgery using Zernike moments," in *Proc. Inf. Hiding Conf.*, Jun. 2010, pp. 51–65.

[10] E. Ardizzone, A. Bruno, and G. Mazzola, "Copy-move forgery detection via texture description," in *Proc. 2nd ACM Workshop Multimedia Forensics, Secur. Intell. (MiFor)*, 2010, pp. 59–64.

[11] X. Pan and S. Lyu, "Region duplication detection using image feature matching," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 4, pp. 857–867, Dec. 2010.

[12] I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo, and G. Serra, "A SIFT-based forensic method for copy–move attack detection and transformation recovery," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 3, pp. 1099–1110, Sep. 2011.

[13] E. Ardizzone, A. Bruno, and G. Mazzola, "Detecting multiple copies in tampered images," in *Proc. 17th IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2010, pp. 2117–2120.

[14] I. Amerini, L. Ballan, R. Caldelli, A. D. Bimbo, L. D. Tongo, and G. Serra, "Copy-move forgery detection and localization by means of robust clustering with J-linkage," *Signal Process., Image Commun.*, vol. 28, no. 6, pp. 659–669, Jul. 2013.

[15] L. Jing and C. Shao, "Image copy-move forgery detecting based on local invariant feature," *J. Multimedia*, vol. 7, no. 1, pp. 90–97, Feb. 2012.

[16] X. Bo, W. Junwen, L. Guangjie, and D. Yuewei, "Image copy-move forgery detection based on SURF," in *Proc. Int. Conf. Multimedia Inf. Netw. Secur. (MINES)*, Nov. 2010, pp. 889–892.

[17] B. L. Shivakumar and S. S. Baboo, "Detection of region duplication forgery in digital images using SURF," *Int. J. Comput. Sci. Issues*, vol. 8, no. 4, pp. 199–205, 2011.

[18] V. Christlein, C. Riess, J. Jordan, C. Riess, and E. Angelopoulou, "An evaluation of popular copy-move forgery detection approaches," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 6, pp. 1841–1854, Dec. 2012.

[19] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[20] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2006, pp. 404–417.

[21] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. 4th Alvey Vis. Conf.*, vol. 15. 1988, p. 50.

[22] R. Dyer, H. Zhang, and T. Möller, "A survey of Delaunay structures for surface representation," GrUVi Lab, Burnaby, BC, Canada, School Comput. Sci., Tech. Rep. TR 2009-01, 2009.

[23] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

[24] [Online]. Available: http://www.dicgim.unipa.it/cvip/, accessed Aug. 2015.

[25] T. Liu *et al.*, "Learning to detect a salient object," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 2, pp. 353–367, Feb. 2011.

[26] E. Ardizzone, A. Bruno, and G. Mazzola, "Saliency based image cropping," in *Image Analysis and Processing—ICIAP*. Berlin, Germany: Springer-Verlag, 2013, pp. 773–782.

Edoardo Ardizzone is a Full Professor of Computer Systems with the Department of Chemical, Management, Computer and Mechanical Engineering, University of Palermo, Italy. He currently teaches Image Processing at the Graduate Course of Computer Engineering, University of Palermo. He has authored or coauthored over 150 scientific papers. He has been responsible of research units in Palermo involved in research projects funded by CNR and MIUR in the areas of computer vision, of automatic indexing and retrieval for image and video databases, and intelligent tools for image and video management in multimedia applications. His current research interests include image processing and analysis, medical imaging, image restoration, and content-based image and video retrieval. He is a member of GIRPR, the association of Italian researchers in pattern recognition and image analysis.

Alessandro Bruno received the master's degree in computer engineering and the Ph.D. degree in computer engineering from the University of Palermo, in 2008 and 2012, respectively. Since 2012, he has held a postdoctoral position with the Computer Vision and Image Processing Group, University of Palermo. His main research interests are in computer vision and image processing, including visual saliency, image and video forensics, object recognition and modeling using images and video, object tracking, and biomedical imaging.

Giuseppe Mazzola received the M.Sc. degree in electronic engineering from the Università degli Studi di Palermo, Palermo, Italy, in 2003, and the Ph.D. degree in computer engineering in 2008. Since then, it has been involved, as a consultant, in several Italian research projects. He is currently a Temporary Research Fellow with the Dipartimento di Ingegneria Chimica Gestionale Informatica e Meccanica, University of Palermo. His research interests include digital image forensics, image restoration, visual saliency, content-based multimedia retrieval, and geographical information retrieval.