Università degli Studi di Firenze
Corso di Dottorato in Informatica, Sistemi e Telecomunicazioni
Dipartimento di Ingegneria dell'Informazione (DINFO)
ING-INF/03

Image Evolution Analysis Through Forensic Techniques

Candidate
Pasquale Ferrara

Supervisors
Prof. Alessandro Piva

Prof. Enrico Del Re

PhD Coordinator
Prof. Luigi Chisci

ciclo XXVII, 2012-2014

Università degli Studi di Firenze, Dipartimento di Ingegneria
dell'Informazione (DINFO).

...to Paola
Your smile is my peace...

Acknowledgments

During my Ph.D. studies, I had the opportunity to work with lot of people, on different themes dealing with image processing, taking part in many activities, as European and regional projects, international conferences and schools, here in Italy and outside my country.

I take this opportunity to acknowledge my Supervisor Prof. Alessandro Piva, for the possibility to study and to work, in complete autonomy, on themes that I appreciate. I also thank him and the whole IAPP staff, Alessia De Rosa, Massimo Iuliani, Marco Fontani, Alessandro Nozzoli e Alessandro Lapini for the collaboration and the friendship in these years.

Moreover, I acknowledge Prof. Mauro Barni of the University of Siena for the support within the REWIND Project and for the opportunity that he and my Supervisor gave me to spend 6 months at the University of Campinas, in Brazil.

I take this opportunity also to thank the Prof. Anderson Rocha and his staff and students for the collaboration and their hospitality.

I also acknowledge Anna Pelagotti and the whole staff of the National Institute of Optics of Florence and of the Centro Regionale per Melanoma of Florence, with which I had the pleasure of working together within the Multispectral Imaging Diagnostics of Skin Tumors Project, in particular Chiara Delfino and Leonardo Pescitelli. By attending to this project, I increased my knowledge in Optics and in biomedical imaging.

I would also like to thank Francesca Uccheddu and Emanuela Massa for our cooperation in image processing applied to Cultural Heritage.

Finally, I thank my family and my fiends for the support in these years. Thank you.

Contents

Chapter 1

Introduction

A picture is worth a thousand words an old Chinese saying stated. The forcefulness of this claim has been kept itself over the centuries, and, if possible, it is reinforced itself nowadays. At the beginning was the painting the vehicle to describe the reality through pictures instead of "thousand words". In the 19th Century, this role was assumed by the photography and then by the cinema, which told us about wars, revolutions and daily news until today. At the beginning of the 21th Century, the "digital revolution" changed not only the way in which a picture is acquired, through digital devices as photocameras, cellphones and tablets (just to name a few), but also the way in which such contents are stored and given out. Personal Computer, Compact Disc, USB key are all examples of digital devices capable of storing pictures in a digital format; in addition to these, new "unmaterialized" systems able to save (and share) digital contents are strongly arising: social networks (as Facebook), websites (Flickr) and Cloud Systems (e.g. Google Drive) allow to store and to deliver images in an easy way, everywhere and anytime, by using digital devices connected to a network, as Internet.

Besides this, the availability of low cost software for image editing (as Adobe Photoshop, GIMP, IrfanView and many others) allowed to common people to easily modify images, to save them in several different formats and to generate new contents from several sources, for example by combining the contents coming from different images or employing Computer Graphics techniques. In a such complex environment, where images can be stored and shared on different platforms and subjected to any processing, the old idea that a picture is something of unchangeable, as an artwork or a picture

of the 19th Century, is overshoot. In our age, a picture in form of digital image, is something similar to a *living organism* with its own evolution over time. The same picture can be stored in many copies, often on different devices, generating different "organisms", which usually change their appearance. Such an evolution is allowed by using image editing tools, by means of several operations: color and geometrical transformations are the most common processing people employ to enhance some characteristics of an image or to make nicer a picture. Moreover, such editing tools allow to modify the content of an image in a very easy way, so everyone is able to do it. Canceling a detail, or making some composition of contents, for example faces of celebrities, are common operations.

In such a dynamic world, several issues arise: the first question is about the intellectual properties of each entity (or "organism") generated from the same picture. Secondly, since an image could be generated by image editing tools, is its own content representative of a fact happened in the real world? Such a question is not a purely intellectual exercise: in a newspaper, a breaking news website, or even within a court, there are images (or more in general multimedia content) that stand as proof of something, and are claimed to be a credible proof. Such questions about the copyright and the integrity of a digital image highlight the growing interest in reconstructing the evolution of digital contents.

Scientific community and industrial companies tried to answer to this questions by using digital watermarking techniques: by inserting an additional information (watermark) within an image, it is possible to track an image during its entire evolution. Although this solution appears promising, it is hard to put into practice. First of all, the watermark has to be included at the instant in which an image was acquired, but, nowadays, not every devices are equipped with an embedding watermark system, and also standardization is far from to be accepted. Secondly, tracing the evolution of an image would require two types of watermark: the first type would be robust, that is the watermark has to be recognizable even if the image is undergone any processing, to track the intellectual property of an image; the second one would be fragile, that is capable of revealing if a processing has been applied to an image, in order to understand what kind of mutation an image suffered.

Along this, in the past decade the attention of the scientific and industrial communities focused on new passive approach, able to recover the history

of the image without the need of inserting additional information when an image has been acquired. This hot research field is named Image Forensics, which is a multidisciplinary science aiming at acquiring important information on the history of digital images, including the acquisition chain, the coding process, and the editing operators. The extraction of such data can be exploited for different purposes, one of the most interesting is the verification of the trustworthiness of digital data. Image forensic techniques [1] work on the assumption that digital forgeries, although visually imperceptible, alter the underlying statistics of an image. These statistical properties can be interpreted as *digital fingerprints* characterizing the image life-cycle, during its acquisition and any successive processing. One of the tasks of Image Forensics is then to verify the presence or the absence of such digital fingerprints, similar to intrinsic watermarks, in order to uncover traces of tampering. For this reason, image forensic techniques seem to be an effective tool to reconstruct the evolution of the images, as it will be shown in this Thesis.

1.1 Overview and Contribution

Our Thesis is divided in two Parts: the first one deals with the case in which images evolve keeping its own semantic content. In Chapter 2 we boost a framework capable of reconstructing the phylogeny of a set of images subjected to small geometrical, color and compression transformations. In Chapter 3, we deal with the case of reconstruction of the evolution of a single image, without knowing who was the parent image. As case of study, we develop two approaches able to detect if a JPEG image has been saved in the same format (with an arbitrary quality factor) after a contrast enhancement has been applied.

In the second Part of this Thesis, we analyze the case in which an image changes its own semantic content, by the composition of multiple parent images. In Chapter 4, a framework to reconstruct the genealogy of a set of images is developed. In the two last chapters, we investigate the case of unavailable parent images, as done in the first Part. In Chapter 5 we develop a system to detect and localize, within a given image, the parts of the image coming from other images. As case of study, we use the artifacts introduced by Color Filter Array within color digital cameras. In the final Chapter, we integrate this tool within a general framework able to localize such regions

by means of a multi-clue analysis.

1.2 Publications List

The research activity related to this Thesis resulted in the following publications, in chronological order:

- International, peer reviewed journals:

    1. **P. Ferrara**, T. Bianchi, A. De Rosa, A. Piva, "*Image Forgery Localization via Fine-Grained Analysis of CFA Artifacts*," IEEE Transactions on Information Forensics and Security, vol. 7, no. 5, Oct. 2012, pp. 1566-1577.

- International, peer reviewed conferences:

    1. **P. Ferrara**, T. Bianchi, A. De Rosa, A. Piva, "*Reverse engineering of double compressed images in the presence of contrast enhancement*," Proceedings of IEEE 15th International Workshop on Multimedia Signal Processing, Pula, Sardinia, Italy, Sept. 2013.
    2. A. Oliveira; **P. Ferrara**; A. De Rosa; A. Piva; M. Barni; S. Goldstein; Z. Dias; A. Rocha, "*Multiple Parenting Identification in Image Phylogeny*," in IEEE International Conference on Image Processing, Oct. 2014, Paris, France.

The author of this Thesis also contributed to the publications listed below; they are not discussed in details in the Thesis as they deal with image processing applied to biomedical images and automatic texture mapping of 3D models.

1. A. Pelagotti, **P. Ferrara**, F. Uccheddu, "*Improving on fast and automatic texture mapping of 3D dense models*," Proceedings of 18th International Conference on Virtual Systems and Multimedia, Milan, Italy, Sept. 2012.

2. F. Uccheddu, A. Pelagotti, **P. Ferrara**, "*Automatic registration of multimodal views on large aerial images*," Proceedings of SPIE 8537, Image and Signal Processing for Remote Sensing XVIII, Edinburgh, United Kingdom, Sept. 2012.

3. **P. Ferrara**, F. Uccheddu, A. Pelagotti, "*Improvements on a MMI-based method for automatic texture mapping of 3D dense models*," Proceedings of SPIE 8650, Three-Dimensional Image Processing (3DIP) and Applications 2013, Burlingame, California, USA, Feb. 2013.

4. A. Pelagotti, **P. Ferrara**, L. Pescitelli, C. Delfino, G. Gerlini, A. Piva, L. Borgognoni, "*Multispectral imaging for early diagnosis of melanoma*," Proceedings of SPIE 8668, Medical Imaging, Lake Buena Vista, Florida, USA, Feb. 2013.

5. A. Pelagotti, **P. Ferrara**, L. Pescitelli, G. Gerlini, A. Piva, L. Borgognoni, "*Noninvasive inspection of skin lesions via multispectral imaging*," Proceedings of SPIE 8792, Optical Methods for Inspection, Characterization, and Imaging of Biomaterials, Munich, Germany, May 2013.

1.3 Activity within Research Projects

Most of the research activity presented in this Thesis has been carried out in the framework of the REWIND project (REVerse engineering of audio-VIsual content Data), funded by the European Commission under the FP7-FET programme and expired on June 2014. The goal of the project was to develop new theories and tools for investigating the digital history of multimedia contents. Also according to project reviewers opinion, REWIND reached and in some cases exceeded its objectives, so that it can be regarded as a successful story we are proud of being part of. Also, it was thanks to the REWIND project that the collaboration with other Universities flourished, in particular the Universidade Estadual de Campinas (UNICAMP), leading to some of the results presented in this Thesis. This participation brought a significant contribution: we learned the importance of establishing contacts, sharing knowledge with other partners, and we hopefully advanced in the ability to focus the efforts toward specific objectives.

Moreover, we participated to the Multispectral Imaging Diagnostics of Skin Tumors (MIDST) project, funded by Tuscan Regional Health Research Program 2009. In this project, a new device for early diagnosis of early melanoma has been developed using a multispectral imaging system, acquiring high spatial and spectral resolution images in the visible and near-infrared range. The images acquired reveal layering of structures in the

epidermal and dermal layer. Such images have been correlated with dermoscopic and histopathological data. Differences between healthy skin and melanoma lesions have been detected and investigated. Our contribution was the development of a software for the analysis and the use of multispectral images by dermatologists. The activity in such a project resulted in several publications and has boosted our knowledge in biomedical image processing.

Part I

Image Phylogeny: from parent to child

Abstract

Images have always played a key role in the transmission of information, mainly because of their immediacy. Nowadays, digital images can be taken, processed and distributed in several easy ways, generating more and more entities, or "organisms", of the same picture. In this Part, we deal with the case in which each picture (which is treated as a "specie") evolves keeping its semantic content. Firstly, we show as Image Phylogeny is able to trace back to the evolution of a picture, given a set of images, by means of a suitable dissimilarity measure between images and a reconstruction graph algorithm. Then, we study the case in which it is possible to reconstruct the evolutionary history of an image, without the availability of other organisms of the same specie. Due to the large amount of combinations of possible ways in which an image can evolve, we propose a case study, wherein two different approaches to trace back the evolution of a JPEG image subjected to a linear contrast enhancement and a further compression.

Chapter 2

Image Phylogeny

Given a set of semantically similar images obtained from a Near-Duplication detector, *Image Phylogeny* is the problem of reconstructing the structure that represents the history of generation of these images. Typical Image Phylogeny approaches break the problem into two steps: the estimation of the dissimilarity between each pair of images (it is not symmetrical, thus it is not a metric), and the reconstruction algorithm. In this Chapter, we propose new alternatives to the standard dissimilarity calculation formulation for image phylogeny. The new formulations explore a different family of color adjustment, local gradient, and mutual information.

2.1 Introduction

Image Phylogeny has been developed recently [2,3] in an attempt to find the relationship structure between a set of near-duplicates images [4]. We model these relationships as a tree where the root is the patient zero (the original image), where the edges represent "father-son" relationships, and where the leaves of the tree represent "terminal" images that have more modifications than their ancestors. In some cases, the near-duplicate set did not come from a single original document, they are images with the same semantic content but generated either from different sources or from the same source but at distinct time instances. In these cases, the set of near-duplicates can be represented by a forest correlating semantically similar images [5,6].

Dias et al. [2, 3] formally defined the problem of Image Phylogeny following two steps: the calculation of the dissimilarity between each pair of near-duplicate images and the reconstruction of the phylogeny tree.

Thus far, researchers mainly focused on proposing different phylogeny reconstruction approaches [2, 3, 5–8] often using a standard methodology for dissimilarity calculation as originally proposed by [3]. This dissimilarity calculation involves the transformation estimation applied for mapping the source image onto the target image's domain followed by the their comparison in a point-wise fashion. As the transformation estimation is not exact, the point-wise comparison method may be affected by artifacts generated in these processes.

Considering the dissimilarity calculation effects on the result of the final phylogeny reconstruction [3], here we introduce alternative methods to perform the dissimilarity calculation between images. We introduce a better family of color transformations, and rather than comparing pixels directly after mapping, we calculate the dissimilarity on the image gradients, rather than directly on the color domain, using the mutual information between them.

2.2 Dissimilarity calculation for Image Phylogeny

Dias et al. [2,3] formalized the image phylogeny problem, separating the problem in two basic steps: The dissimilarity calculation between images and the reconstruction of the phylogeny forest.

About the dissimilarity calculation, let $\mathcal{T}$ be a family of image transformations, and T be a transformation such that $T \in \mathcal{T}$. Given two images I_s (source) and I_t (target), the dissimilarity function d between them is defined as the lowest value of $d_{\mathcal{I}_s,\mathcal{I}_t}$, such that

$$d(\mathcal{I}_s, \mathcal{I}_t) = \min_{T_{\vec{\beta}}} |\mathcal{I}_t - T_{\vec{\beta}}(\mathcal{I}_s)|_{\text{ point-wise comparison }} \mathcal{L}, \tag{2.1}$$

for all possible values of the parameter β in $\mathcal{T}$. Equation 2.1 calculates the dissimilarity between the best transformation mapping $\mathcal{I}_s$ onto $\mathcal{I}_t$, according to the family of transformations $\mathcal{T}$ and $\mathcal{I}_t$. Then, the comparison between the images can be performed by any point-wise comparison method $\mathcal{L}$.

For the estimating of the transformation T used to transform I_s in I_t, the authores follow three basic steps:

1. **Image Registration:** first, we find interest points in each pair of images, using SURF [9], which will be used to estimate warping and cropping parameters robustly using RANSAC [10];
2. **Color matching:** the pixel color normalization parameters are calculated using the color transfer technique based in mean and standard deviation, proposed in [11];
3. **Compression matching:** the image I_s is compressed with the same JPEG compression parameters as the image I_t.

As a final step, both images are uncompressed and compared pixel-by-pixel according to the minimum squared error (MSE) metric.

This dissimilarity is calculated for each pair of images. After this, we have a dissimilarity matrix $M_{n\times n}$, where n is the number of near-duplicates and each region of the matrix represents the dissimilarity between one pair of images. Note that the matrix M is asymmetric, once that the dissimilarity $d(\mathcal{I}_i, \mathcal{I}_j) \neq d(\mathcal{I}_j, \mathcal{I}_i), \forall i, j = 1, 2, ..., n | i \neq j$. After calculating this dissimilarity matrix, we perform an algorithm for reconstruct the phylogeny forest. There are several approaches in the literature based on graphs algorithm [2,3,5,8], but these algorithms are not the main focus of this Chapter.

2.3 Dissimilarity Calculation

We now turn our attention to new approaches for improving the dissimilarity calculation.

2.3.1 Gradient Comparison

Image gradients describe the value and direction of pixel intensity variation. They can be used to extract different information about the image, such as texture and location of edges. Here se use the *Sobel* [12] gradient estimator [13].

As contrast enhancement and color transformations are ofter used when creating near duplicates, which directly affects the gradients of the image, this becomes an important information to add to the dissimilarity calculation. By comparing the gradients of transformed image $\mathcal{I}'_s = T_\beta(\mathcal{I}_s)$ and $\mathcal{I}_t$,

it is possible to compare both the intensity values (encoded in the gradient), as well as their variation throughout the image.

While the image comparison metric $\mathcal{L}$ stays the same, we first compute the gradients in the horizontal and vertical directions, by convolving the images to be compared with the 3×3 Sobel kernels S_h (horizontal direction) and S_v (vertical direction). The R, G and B channels of $\mathcal{I}'_s$ and $\mathcal{I}_t$ are treated separately resulting in a total of six gradients (two directions per color channel). The image comparison metric $\mathcal{L}$ is applied to each respective pair of gradient images of $\mathcal{I}'_s$ and $\mathcal{I}_t$, and the mean of the six values obtained in each position is taken as the final dissimilarity value.

2.3.2 Mutual Information Comparison

In Information Theory, mutual information (MI) is a measure of statistical dependency of two random variables, which represents the amount of information that one random variable contains about the other [14]. The mutual information between two random variables X and Y is given by:

$$MI(X,Y) = H(Y) - H(Y|X) = H(X) - H(X|Y), \tag{2.2}$$

where $H(X) = -E_x[\log(P(X))]$ is the entropy (i.e., the expected value of the information associated to a random variable) of X and $P(X)$ is the probability distribution of X. In the case of discrete random variables, MI is defined as:

$$MI(X,Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \left(\frac{p(x,y)}{p(x)(p(y)} \right), \tag{2.3}$$

where $p(x,y)$ is the joint Probability Distribution Function (PDF) [15] of X and Y, and both $p(x)$ and $p(y)$ are the marginal PDFs of X and Y, respectively.

MI has been widely employed in several image applications, such as gender identification [16], multi-modal data fusion [17], feature selection [18], and in image registration problems [19, 20] as a similarity measure (or cost function) to maximize when aligning two images (or volumes).

Applying MI to images means that the two random variables are the image $X = \mathcal{I}'_s$ and the image $Y = \mathcal{I}_t$, x and y are the value of two pixels belonging to $\mathcal{I}'_s$ and $\mathcal{I}_t$, respectively. Thus, $p(x,y)$ is the joint PDF of the images $\mathcal{I}'_s$ and $\mathcal{I}_t$, evaluated for the values (x,y), where $x, y \in [0 \dots 255]$.

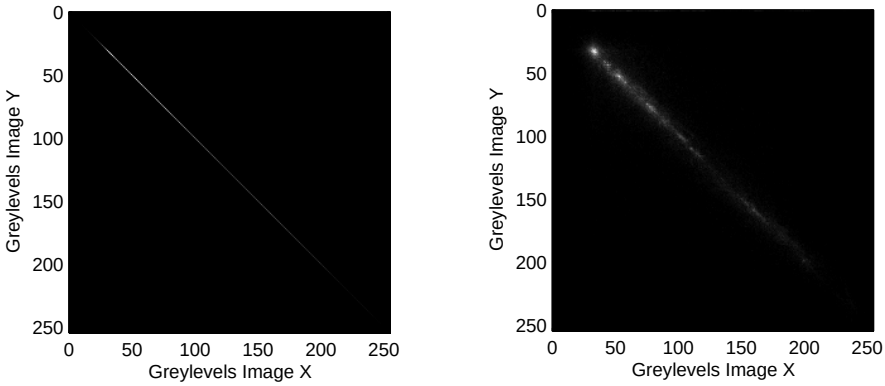


Figure 2.1: *Bi-dimensional representation of two joint histograms. White pixels mean zero values while the other pixels represent values greater than zero (the images were inverted for viewing purposes). Left, we show the joint histogram of two (gray-scale) images perfectly aligned. Right, we show the joint histogram of two slightly misaligned images.*

Clearly, the previous definitions involve the knowledge of the PDFs of pixels and, in particular, the joint PDF $p(x, y)$, from which it is easy to obtain $p(x)$ and $p(y)$ by marginalization. In general, such joint PDF is not *a priori* known, and needs to be estimated. Several methods [21] have been conceived to estimate the PDF of one or more random variables from a finite set of observations, such as the approximation of the joint PDF by the joint histogram

$$\hat{p}(x, y) = \frac{h(x, y)}{\sum_{x,y} h(x, y)}, \tag{2.4}$$

where $h(x, y)$ is the joint histogram of the images X and Y. MI has the following property: given two images $\mathcal{I}'_s$ and $\mathcal{I}_t$, $MI(\mathcal{I}'_s, \mathcal{I}_t)$ is bounded as $0 \leq MI(\mathcal{I}'_s, \mathcal{I}_t) \leq \min(H(\mathcal{I}'_s), H(\mathcal{I}_t))$. It can be demonstrated that MI is maximum when the two images are completely aligned (in terms of geometrical, color and compression transformation). Figure 2.1(a) shows a perfectly aligned case.

If we assume a good transformation $T_{\vec{\beta}}$ that maps an image $\mathcal{I}_s$ onto an image $\mathcal{I}_t$, the mutual information $MI(T_{\vec{\beta}}(\mathcal{I}_s), \mathcal{I}_t)$ is maximum. Moreover, since each transformation is not completely reversible, if we apply the inverse transformation $T^{-1}_{\vec{\beta}}$ to $\mathcal{I}_t$ to obtain $\mathcal{I}_s$, their joint histogram is similar to the

right plot of Figure 2.1.

2.3.3 Gradient Estimation and Mutual Information Combined

Here we consider the combination of the gradient information and the mutual information. First, we calculate the gradient of the images $\mathcal{I}'_s$ and $\mathcal{I}_t$ as descried on Section 2.3.1. After this, we compare each correspondent gradient of both images with mutual information, instead of using the image comparison metric $\mathcal{L}$. The final dissimilarity is the average of mutual information values for each gradient image.

With this approach, we aim at better capturing the information about variation in certain directions of the image (gradient information), as well as at seeking to avoid effects caused by slight misalignments during the mapping (mutual information estimation). This method also takes into consideration the amount of texture information preserved between two near duplicates for calculating the dissimilarity.

Unfortunately, the combined method slightly increases the computational cost of the dissimilarity calculation, once we need to estimate the mutual information six times after the gradient calculation. However, this method provides better reconstruction results as we shall discuss in Section 2.4.4.

2.3.4 Histogram Color Matching

As previously discussed, the second step of the transformation estimation T is mapping the color space of the source image $\mathcal{I}_s$ onto the target's image $\mathcal{I}_t$ color space, by normalizing each channel of $\mathcal{I}_s$ by the mean and standard deviation of $\mathcal{I}_t$'s corresponding channel [11]. This method, although simplistic, works reasonably well, specially when the color changes are minor. However, it leads to some problems when the transformations applied to the image when generating a child are stronger, specially in the case of contrast changes, which affects the distribution of pixel intensities throughout the image.

We propose to use the histogram matching technique [22] for color estimation between images.

To match the histograms of two images $\mathcal{I}_s$ and $\mathcal{I}_t$, we compute their histograms, H_s and H_t. Then we compute the *Cumulative Distribution Function* (CDF) [15]. For a gray-scale image F, with L gray levels, the gray level

i has the probability of

$$p_F(i) = \frac{n_i}{n}, \quad 0 \leq i < L \tag{2.5}$$

where n is the number of pixels in the image and n_i is the number of pixels of gray value i in the histogram of the image. The CDFs of an image F is

$$\mathcal{C}_F(i) = \sum_{k=0}^{i} p_F(k). \tag{2.6}$$

With $\mathcal{C}_s$ and $\mathcal{C}_t$, the CDFs for $\mathcal{I}_s$ and $\mathcal{I}_t$, we find a transformation $\mathcal{M}$ that maps $\mathcal{C}_s$ onto $\mathcal{C}_t$. For each gray level i of $\mathcal{I}_s$, finding the gray level j of $\mathcal{I}_t$ whose $\mathcal{C}_t(j)$ is the closest in $\mathcal{C}_t$ to $\mathcal{C}_s(i)$. Once the mapping is found, each pixel with gray level i in $\mathcal{I}_s$ has its value replaced by j. We treat each channel of these images independently, matching their histograms individually.

2.4 Experiments and Validation

In this section, we validate the proposed methods and compare them to the state-of-the-art MSE method for the dissimilarity calculation used in [2, 3, 5–8].

2.4.1 Phylogeny reconstruction

After calculating the dissimilarity matrix, we use an algorithm for reconstructing the phylogeny forest. Here, we use the Extended Automatic Optimum Branching (E-AOB) algorithm proposed by Costa et al. [6] currently the state-of-the-art for phylogeny reconstruction. This method is based on an optimum branching algorithm [23]. We use the best parameter reported by the authors ($\gamma = 2.0$).

2.4.2 Dataset

Here, we used the set of near-duplicate images from [6] – freely available. This set comprises images randomly selected from a set of 20 different scenes generated by 10 different acquisition cameras, 10 images per camera, 10 different tree topologies (i.e., the form of the trees in a forest) and 10 random variations of parameters for creating the near-duplicate images.

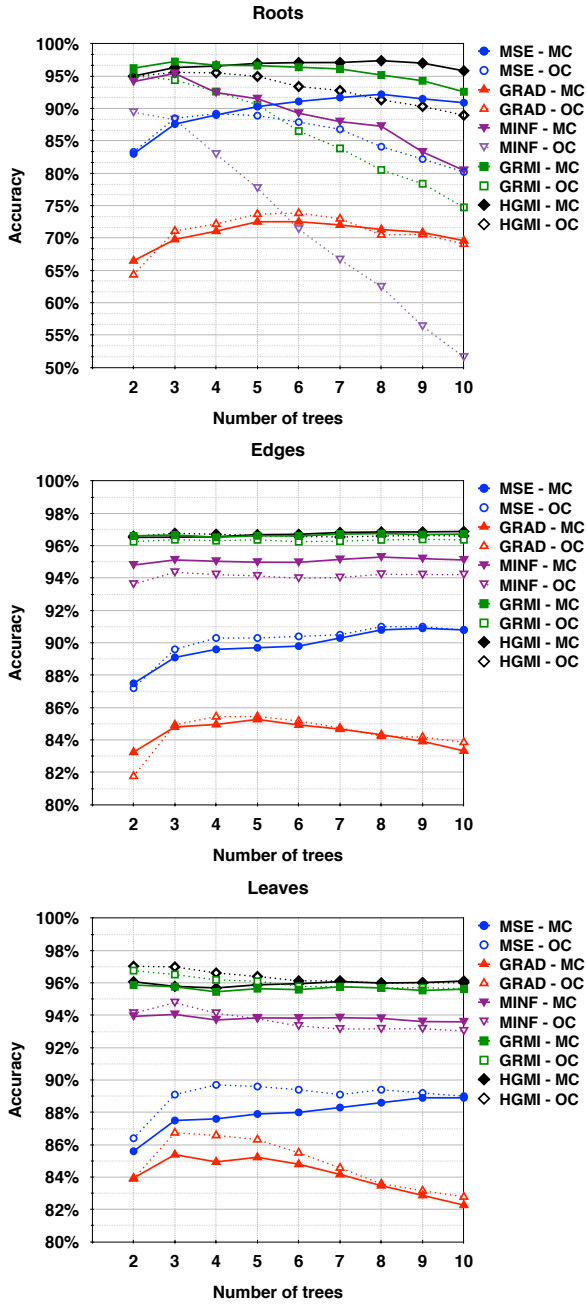


Figure 2.2: *Results of forest reconstruction in multiple cameras (MC) and one camera (OC) scenarios considering the metrics Roots, Edges and Ancestry. Similar results are obtained for the Leaves metric.*

We considered 2,000 forests of images generated by a single camera (Scenario *One Camera* – OC) and 2,000 forests generated by multiple cameras (Scenario *Multi Camera* – MC). The forests vary in the number of trees (size) $|F| = \{1..10\}$. The dataset has $2 \times 2,000 \times 10 = 40,000$ test cases.

The image transformations used to create the near-duplicates are the same used in [3, 6]: re-sampling, cropping, affine warping, brightness and contrast adjustment, and lossy compression using the standard JPEG algorithm.

2.4.3 Evaluation metrics

For a better assessment of the proposed methods, we consider scenarios in which the *ground truth* is available. We used the metrics introduced in [5] in the experiments: *Roots, Edges, Leaves* and *Ancestry* given by:

$$\mathrm{EM}(\mathrm{IPF}_R, \mathrm{IPF}_G) = \frac{S_R \cap S_G}{S_R \cup S_G} \tag{2.7}$$

where EM is the evaluation metric, IPF_R is the reconstructed forest with elements represented by S_R, and IPF_G is the forest ground truth with elements S_G. For instance, when considering the Edges metric, we calculate the intersection of the set of reconstructed edges with the set of edges in the ground truth normalized all edges present in both groups.

2.4.4 Results and discussion

In this section, we analyze the impacts of calculating the dissimilarities using image gradients instead of image intensities, the replacement of the point-wise comparison metric with a mutual information dissimilarity calculation, and the incorporation of color matching for better representing the mapping of a source image onto a target image before calculating the dissimilarity.

Figure 2.2 shows the results for the different approaches considered herein for calculating the dissimilarities. In all cases, the mapping of one source image onto a target image is already performed as discussed in Section 2.3. The phylogeny reconstruction part uses the E-AOB algorithm for all methods. The first dissimilarity calculation considered is the MSE, the state of the art, which compares two images point-wise using the pixel intensities. The proposed modifications are gradient estimation (GRAD), which still compares

the images point-wise but using image gradients instead of pixel intensities; mutual information (MINF), which replaces the point-wise comparison using pixel intensities with the mutual information calculation of pixel intensities; gradient estimation plus comparison with mutual information (GRMI), incorporating the calculus of dissimilarity using mutual information of image gradients; and, finally, and histogram color matching plus gradient estimation with mutual information (HGMI), extending the GRMI to incorporate a better color matching approach before comparison.

First of all, the dissimilarity calculation does not benefit directly from the replacement of point-wise pixel intensity comparison by a point-wise comparison of image gradients as the results show MSE outperforming GRAD for OC and MC scenarios. The gradient itself only captures directionality variations and small misaligments when comparing two gradient images affect the results more than slight misalignments in the pixel intensities.

If we change the point-wise comparison method to mutual information but still use the pixel intensities, we have MINF outperforming MSE for the MC case. With MINF, small misalignments are not as important as for the GRAD case. One interesting behavior, however, is the decrease in performance for the OC case (Root and Ancestry metrics). In the OC case, as all of the images come from the same camera, the color matching for such images should be more refined than just the mapping using the mean and standard deviation to differentiate an image and its descendant. A point-wise comparison, in this case, is more effective for small differences (MSE method).

The results improve when combining the gradient calculation with mutual information (GRMI). The first reason is that, by not comparing the intensities, the color information artifacts are not as strong. Second, the comparison is not done in a point-wise fashion but rather, in a probability distribution-like form better capturing the different variations in the gradient images as well as accounting for slight misalignments. Finally, solving the color matching problem when using MINF, we end up combining GR + MI + Color matching and creating the final method HGMI. As we can see, HGMI outperforms the MSE baseline for all cases, due to the fact that, with this approach, we can reduce the dissimilarity errors by better matching the color transformations involved in the process of near-duplicate generation and by comparing the images using gradients instead of pixel intensities and in a distribution-like form instead of a point-wise one.

According to a Wilcoxon signed-rank test [24], the best proposed approach, HGMI, is statistically better than the state-of-the-art MSE method for all cases and metrics, with 95% of confidence.

To compare a pair of typical images (each with about one megapixel), including the time to register both images, MSE takes about 0.6s, GRAD takes 0.8s, and MINF takes 0.7s. The best performing methods GRMI and HGMI take both about 1.5s. The experiments were performed in a machine with an Intel Xeon E5645 processor, 2.40GHz, 16GB of memory, and Ubuntu 12.04.5 LTS.

2.5 Conclusion

In this Chapter, we presented approaches for improving the dissimilarity calculation between images for the problem of image phylogeny forest reconstruction. Our approaches were based on gradient comparison and mutual information estimation. We also studied the incorporation of a more robust color matching approach for better estimating the involved changes during the generation of near duplicates.

This Chapter shows that comparing distributions is better than direct point-wise comparisons, it shows that gradient distributions are better color distributions, and it also shows that a more powerful family of color transformations enables better tree reconstruction at the end of the pipeline. In the supplemental material we provide direct comparison, using the Wilcoxon signed-rank test, between the GRMI and all combinations of these methods.

For future work, we intent to investigate other ways to calculate the dissimilarity between images. We can investigate the use of mutual information for estimating the step of image registration [20]. Finally, we will investigate the use the impacts of the new dissimilarity calculations to phylogeny estimation in different medias such as video and text.

Chapter 3

Blind parent reconstruction: a case of study.

Two forensic techniques for the reverse engineering of a chain composed by a double JPEG compression interleaved by a linear contrast enhancement are presented in this Chapter. The first approach is based on the well known peak-to-valley behavior of the histogram of double-quantized DCT coefficients, while the second approach is based on the distribution of the first digit of DCT coefficients. These methods have been extended to the study of the considered processing chain, for both the detection of the chain and the estimation of its parameters. More specifically, the proposed approaches provide an estimation of the quality factor of the previous JPEG compression and the amount of linear contrast enhancement.

3.1 Introduction

A variety of tools have been proposed so far for the analysis of fingerprints left by specific processing, leading to the detection of resampling [25, 26], the detection of contrast enhancement [27], the tracing of compression history [28–30], just to name a few. A common characteristic of most of the proposed works is to consider a single processing step at a time; on the contrary, in realistic scenarios a *chain* of such operations is employed to obtain the final processed image. Thus, to go one step further, the forensic analysis should consider the identification of operators in the presence of multiple processing steps. As an example in this sense, several methods have been proposed to

study the double JPEG compression that can be seen as a chain composed by two subsequent coding [28–30]. But if we consider heterogeneous chain, i.e. composed by two different processing operators, only a small effort has been made so far, for example in [31], where authors propose to analyze double JPEG compressed images when image resizing is applied between the two compressions, and provide a joint estimation of both the resize factor and the quality factor of the previous JPEG compression.

In this Chapter, we consider a chain composed by double JPEG compression interleaved by a linear contrast enhancement. A wide literature has been written about double compression (as mentioned previously) or contrast enhancement artifacts, but these fingerprints were treated separately. Usually, contrast enhancement detectors are based on the analysis of histograms of pixels as in [27, 32, 33], whose performance dramatically decreases when a lossy compression is subsequently applied.

Here, we assume the following processing chain: the luminance Y of a JPEG color image with quality QF_1 is linearly stretched and then re-saved in another JPEG color image with quality QF_2. We propose two approaches, borrowed by double JPEG compression detection and extended for the identification of the considered chain; furthermore, assuming QF_2 to be known, the methods provide the joint estimation of the chain operator parameters, i.e. the first quality QF_1 and the amount of contrast enhancement.

3.2 Proposed Approaches

In [28, 30, 34], the effects of double compression on DCT coefficients are well explained and exploited to detect double or single JPEG compression, to localize forged regions, or for steganalysis [35, 36]. Briefly, double compression involves a double quantization of DCT coefficients. Each quantization introduced a periodic peak-to-valley pattern across DCT coefficients histograms, due to the rounding to integers.

If we denote c^{kl} a generic unquantized coefficient, and q_1^{kl} and q_2^{kl} (where $k, l = 1, \dots 8$) the quantization matrix of the first and the second compression, respectively, the quantized coefficient c_1^{kl} is

$$c_1^{kl} = Q_{q_1^{kl}}(c^{kl}) = \left\lfloor \frac{c^{kl}}{q_1^{kl}} \right\rceil \tag{3.1}$$

and the corresponding dequantized d_1^{kl} is

$$d_1^{kl} = Q_{q_1^{kl}}^{-1}(Q_{q_1^{kl}}(c^{kl})) = \left\lceil \frac{c^{kl}}{q_1^{kl}} \right\rfloor q_1^{kl}. \tag{3.2}$$

Now we introduce the linear contrast enhancement as a linear mapping of pixel values, namely:

$$Y_{out} = \alpha Y_{in} + \beta. \tag{3.3}$$

When a *Discrete Cosine Transform* (DCT) is applied to a linear contrast enhanced grayscale image (as could be the luminance Y of a color image) each DCT coefficient is linearly mapped into another value by the same parameters α and β, due to the linearity of the transform, apart from an error term due to the rounding to 8-bit in pixel domain. In order to simplify the model, we can assume that the processing depends on α only, and thus we have $\beta = 0$, and we can neglect the effects of clipping to the range $[0, 255]$. By applying the enhancement considering the relation (3.3) with $\beta = 0$, and introducing an additive noise term ϵ taking into account the rounding to 8-bit in pixel domain, we have that the DCT coefficient after the processing will become:

$$d_1^{'kl} = \alpha \left\lceil \frac{c^{kl}}{q_1^{kl}} \right\rfloor q_1^{kl} + \epsilon, \tag{3.4}$$

and after the second quantization we will obtain the double quantized coefficient:

$$c_2^{kl} = \left\lceil \left(\alpha \left\lceil \frac{c^{kl}}{q_1^{kl}} \right\rfloor q_1^{kl} + \epsilon \right) \frac{1}{q_2^{kl}} \right\rfloor. \tag{3.5}$$

3.2.1 DCT Coefficients Histograms

The periodic pattern of the histogram of doubly compressed DCT coefficients can be modeled as in [34] by computing the number $n(c_2^{kl})$ of bins of the original histogram contributing to bin c_2^{kl} in the doubly compressed histogram, that in this case is given by

$$\begin{aligned} n(c_2^{kl}) = & q_1^{kl} \left\{ \left\lfloor \frac{1}{\alpha q_1^{kl}} \left(q_2^{kl} \left(c_2^{kl} + \frac{1}{2} \right) - \epsilon \right) \right\rfloor \right. \\ & \left. - \left\lceil \frac{1}{\alpha q_1^{kl}} \left(q_2^{kl} \left(c_2^{kl} - \frac{1}{2} \right) - \epsilon \right) \right\rceil + 1 \right\} \end{aligned} \tag{3.6}$$

It is possible to demonstrate that $n(c_2^{kl})$ is periodic with a period which can be computed as follows. Let us consider the following function

$$f_a(x) = \lfloor x + a \rfloor - \lceil x - a \rceil \tag{3.7}$$

where a is a real number. It can be easily demonstrated that the period of $f_a(x)$ is 1, for all real a. It is also easy to show that $f_a(x-b)$ has still period equal to 1, whereas the scaled version

$$f_a\left(\frac{x}{\gamma}\right) = \left\lfloor \frac{x}{\gamma} + a \right\rfloor - \left\lceil \frac{x}{\gamma} - a \right\rceil \tag{3.8}$$

has period equal to γ. By using the previous properties, we can write $n(c_2^{kl})$ using f_a as

$$n(c_2^{kl}) = q_1^{kl} \left\{ f_a \left(\frac{q_2^{kl}}{\alpha q_1^{kl}} c_2^{kl} - \frac{\epsilon}{\alpha q_1^{kl}} \right) \right\} \tag{3.9}$$

where $a = \frac{q_2^{kl}}{2\alpha q_1^{kl}}$ and the period is, as for the function (3.8),

$$\tau'^{kl} = \gamma = \frac{\alpha q_1^{kl}}{q_2^{kl}} \tag{3.10}$$

The result can be seen as a generalization of that found in [34], with the difference related to the presence of α and ϵ. In particular, we can observe that the periodicity of the function $n(c_2^{kl})$ depends on the value α, while it is not modified by ϵ. The period could not be an integer but a rational number.

We can now describe a method to detect the presence of such a chain leveraging on the previous analysis. To do this, we need to know the distribution of DCT coefficients histograms of an image in the presence and in the absence of double compression. Let us suppose that we are observing a double compressed image; as in [28, 37], a method to obtain a histogram of DCT coefficients without periodical pattern from a doubly compressed image is to compute the DCT coefficients by misaligning the grid of 8×8 blocks employed in JPEG standard. In such a way, we can observe two histograms for DCT coefficients at frequency kl: the first one, which we name $h(c_2^{kl})$, is obtained directly from the image, whereas the second one, which we name $h_s(c_2^{kl})$, is obtained as explained before and it represents the hypothesis of absence of a double compression (*smoothed* histogram). From these histograms, we estimate the *probability density functions* (pdf) of a

given DCT coefficient, $p(c_2^{kl})$ and $p_s(c_2^{kl})$, respectively, as in [37]. Ideally, $p(c_2^{kl}) = p_s(c_2^{kl})$ in a single compressed image, whereas $p(c_2^{kl}) \neq p_s(c_2^{kl})$ in double compressed images, because of the presence of a periodic pattern in $p(c_2^{kl})$ that does not appear in $p_s(c_2^{kl})$.

We propose to use two different measures of similarity between two probability distributions: the *Kullback-Liebler divergence* (D_{KL}) [38] and the *Kolmogorov-Smirnov distance* (D_{KS}) [29]. These measure are defined for each DCT coefficient histogram of Y. In order to obtain a scalar value, we assume to sum the Kullback-Leibler distances of each DCT histogram, as for the Kolmogorov-Smirnov divergences.

If the image is considered processed by the supposed chain, to estimate the first compression quality QF_1 and α is an interesting task from a research perspective. In [39], a Maximum Likelihood Estimation (MLE) approach has been proposed to detect JPEG compression in raster bitmap format images and to estimate the quantizer used. Although MLE approach may seem the trivial way to estimate the triplet (QF_1, QF_2, α), the computational cost of this approach grows considerably by increasing the number of parameters to be estimated.

Therefore, as in [40], [41] and others, we employ a *Discrete Fourier Transform* based analysis of DCT histograms. Before this, we pre-process $p(c_2^{kl})$ in order to reduce the effects of low-pass frequencies due to the shape of the histograms: the spectrum is then calculated on $p_n(c_2^{kl}) = p(c_2^{kl}) - p_s(c_2^{kl})$. After that, the period $\hat{\tau}'^{kl}$ is estimated by finding the peak with maximum amplitude through a smooth interpolation [31], in order to achieve a better estimate of the frequency $F = 1/\hat{\tau}^{kl}$.

An exhaustive search is performed over all possible α and QF_1, by discretizing them, to minimize the distance between the theoretical period τ'^{kl}, computed according to Equation (3.10), and the estimated period $\hat{\tau}'^{kl}$. This distance is the median value of residues defined as

$$\rho_{kl} = \left(\frac{\hat{\tau}'^{kl} - \tau'^{kl}}{\tau'^{kl}} \right)^2 \tag{3.11}$$

for a subset n_c of DCT coefficients c_2^{kl}. The choice of working on a subset of coefficients is due to fact that histograms with a small support don't show detectable peaks in their spectra, as shown in [40]. The median value is employed to bound the effects of ambiguities: it can be easily verified that it is well possible that different (α, QF_1) tuples result in equivalent periodic

artifacts in the histogram of a given coefficient. As we have a set n_c of DCT coefficients, these ambiguities can be present on a number of coefficients $n_a \leq n_c$. In all those cases in which $n_a \leq \lfloor n_c/2 \rfloor$, the distance doesn't take into account errors due to the ambiguities.

In our analysis we have to take into account that when the period τ'^{kl} is greater than 2, we observe in the spectrum the fundamental harmonic with frequency $F = 1/\tau'^{kl}$. Conversely, when $1 < \tau'^{kl} < 2$, we don't observe F, but the *aliased* frequency $F = 1 - 1/\tau'^{kl}$. Finally, when $\tau'^{kl} < 1$, we can not observe any fundamental period, because the histogram can be viewed as a sampled signal, where the sample period is 1. However, high order harmonics can still be observed if they are greater than 1, but the peaks associated with them could have undetectable amplitude. Because we do not know if the theoretical period is less or greater than 2, we test both $\tau'^{kl} = 1/F$ and $\tau'^{kl} = 1/(1-F)$ for each DCT coefficient separately, and the period giving lower residue is taken into account in (3.11).

3.2.2 Mode Based First Digit Features

In [42], it is observed that the distribution of the first digit of quantized DCT coefficients can be used to distinguish singly and doubly JPEG compressed images. Briefly, when an image is singly compressed, it is observed that the magnitudes of DCT coefficients approximately follow an exponential distribution. Hence, the distribution of the first digit of quantized DCT coefficients is well modeled by the *generalized Benford's law* [42]. Instead, in case of double compression, the distribution of the first digit is usually perturbed and it violates the generalized Benford's law.

In [43], the authors introduce a new feature based on the distribution of the first digit of DCT coefficients for each separate DCT frequency, or mode. The features are obtained by measuring the frequencies of the 9 possible nonzero values of the first digit for each of the first 20 DCT modes. The resulting $9 \times 20 = 180$ frequencies form a vector of features named Mode Based First Digit Features (MBFDF).

The approach based on Benford's law can be extended also to images modified by contrast enhancement. Even if contrast enhancement is expected to modify the distribution of the first digit, the resulting distribution will still violate the generalized Benford's law, so that MBFDF can still be used to distinguish singly and doubly compressed images. Moreover, different parameters of the contrast enhancement operator will produce different

patterns on the distribution of the first digit of DCT coefficients. Hence, MBFDF can also be used to discriminate different parameters of the processing chain.

Similarly to [43], in order to distinguish enhanced and recompressed images from singly compressed images, we propose to apply a two-class classification to MBFDF according to Fisher's linear discriminant analysis (LDA). The parameters of the processing chain, i.e., QF_1 and α, can be estimated by using a "one-against-one" multi-classification strategy, where each possible combination of values for QF_1, α is considered as a different class. Given N_C possible classes, we construct $N_C(N_C-1)/2$ two-class LDA classifiers, where the classifiers consider every possible combination of two classes. Each classifier "votes" for its winning class, and the class obtaining more votes corresponds to the estimated values QF_1, α.

The above approach works well in presence of a finite set of possible parameters, like in the case of QF_1. However, for continuous valued parameters, like α, it requires a quantization of the parameter space, with a proper choice of the quantization step, since a fine search of parameter values may be impractical due to the fact that the number of required classifiers grows quadratically with the number of parameter values.

3.3 Experimental results

In this section, we show the results about the detection of the considered processing a chain, i.e., we verify the presence/absence of double compression interleaved by contrast enhancement, and about the estimation of the parameters which characterize it. The proposed algorithms have been tested on a dataset composed by 300 TIFF images coming from 3 different cameras (*Nikon D90*, *Canon 5D*, *Panasonic DMC-G2*), cropped to 1024×1024 pixels, and representing landscapes, buildings and people, avoiding uniform content and with different degrees of texture. We fix $\alpha \in \{1.05, 1.15, 1.35, 1.55, 1.75\}$. For each α, we generate two datasets: the first dataset contains TIFF images which are first enhanced and then compressed (i.e. single compression scenario) with a quality factor QF_2, whereas the second dataset contains TIFF images compressed with a quality factor QF_1, whose luminance is enhanced, and then re-compressed with a quality factor QF_2 (i.e. double compression scenario). We compress images by applying the Matlab function `imwrite` at different quality factors chosen in $[50, 55, \ldots, 100]$, for each α. This policy is

then repeated for images with size 256×256 and 64×64.

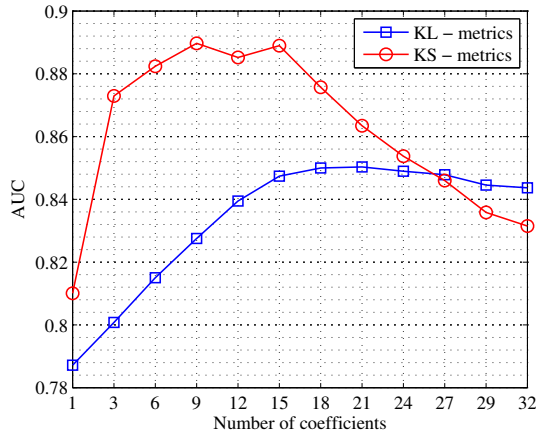


Figure 3.1: *AUCs of KS and KL metrics, evaluated for different number of coefficients, by starting from the DC coefficient and obtained by averaging over all possible α, QF_1 and QF_2.*

3.3.1 Detection

To compare histogram based features (as D_{KL} and D_{KS} ones) and MBFDF in detecting the presence of a double compression, we evaluate the performance of detectors by measuring the *true positive rate* and the *false positive rate*. The overall performance is evaluated by plotting their *receiver operating characteristic* (ROC) curves, obtained by thresholding the distributions of each feature in both hypotheses using a varying threshold value, and recording the corresponding value of true positive and false positive rate. Finally, the *area under the ROC curve* (AUC) is used as a scalar parameter: an AUC close to one indicates good detection performance, whereas an AUC close to 0.5 indicates that the detector has no better performance than choosing at random.

First of all, we evaluated the best performance between D_{KL} and D_{KS}, by varying the number of coefficients n_c. As shown in Figure 3.1, the best detection capability, in terms of AUC, is recorded by employing D_{KS} with the first $n_c = 9$ DCT coefficients; we then decided to fix this configuration for the successive detection analysis.

QF1/QF2	50	60	70	80	90	100
50	0.95	1.00	1.00	1.00	1.00	1.00
60	0.91	0.95	1.00	1.00	1.00	1.00
70	0.91	0.94	0.96	1.00	1.00	1.00
80	0.98	0.98	0.98	0.96	1.00	1.00
90	0.83	0.89	0.96	0.98	0.93	0.99
100	0.5	0.5	0.51	0.55	0.69	0.65

Table 3.1: *Detection performance: AUC values of KS metrics for a subset of pairs* (QF_1, QF_2) *with* $QF_1, QF_2 = \{50, 60, 70, 80, 90, 100\}$*, by fixing* $n_c = 9$ *and averaging over all possible values of* α*.*

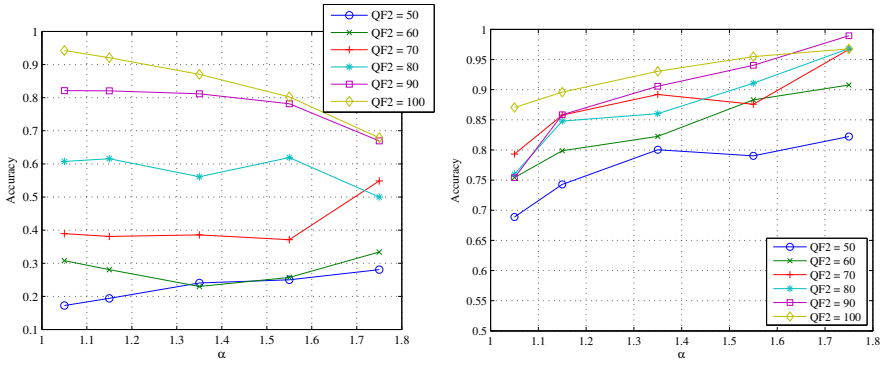
QF1/QF2	50	60	70	80	90	100
50	0.95	1.00	1.00	1.00	1.00	1.00
60	0.99	0.97	1.00	1.00	1.00	1.00
70	1.00	0.99	0.97	1.00	1.00	1.00
80	1.00	0.99	1.00	0.95	1.00	1.00
90	0.90	0.98	0.98	1.00	0.92	1.00
100	0.58	0.60	0.63	0.66	0.82	0.82

Table 3.2: *Detection performance: AUC values of MBFDF for a subset of pairs* (QF_1, QF_2) *with* $QF_1, QF_2 = \{50, 60, 70, 80, 90, 100\}$*, by mediating over all possible values of* α*.*

To compare histograms based versus MBFDF approach, AUC values have been evaluated for different couples (QF_1, QF_2), by mediating over all possible values of α. The results are reported in Table 3.1 and Table 3.2. For lack of space, only the subset $\{50, 60, 70, 80, 90, 100\}$ of all couples (QF_1, QF_2) is shown. When $QF_1 \leq QF_2$, both approaches have a very high capability of detecting double compression, but when $QF_1 > QF_2$, MBFDF method clearly outperforms histogram based ones.

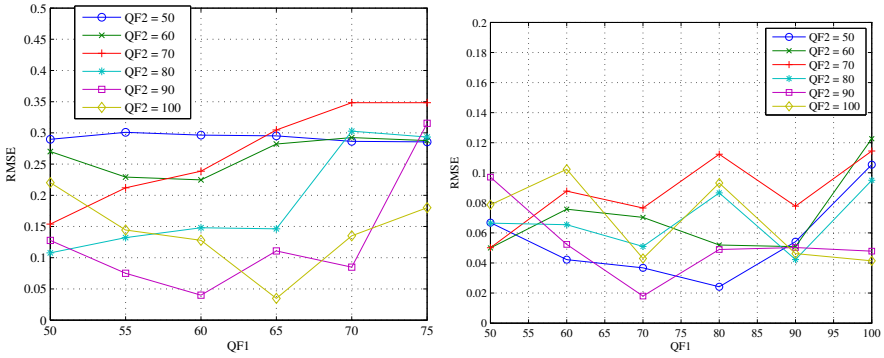
3.3.2 Estimation

We then evaluate the ability of the two approaches to estimate the parameter α and the first compression quality factor. In order to allow a fair comparison between the proposed approaches, we have decided to discretize $\alpha = [1, 2]$ with stepsize 0.05 and $QF_1, QF_2 \in \{50, 60, 70, 80, 90, 100\}$. Whereas the histogram based approach makes an exhaustive search over all



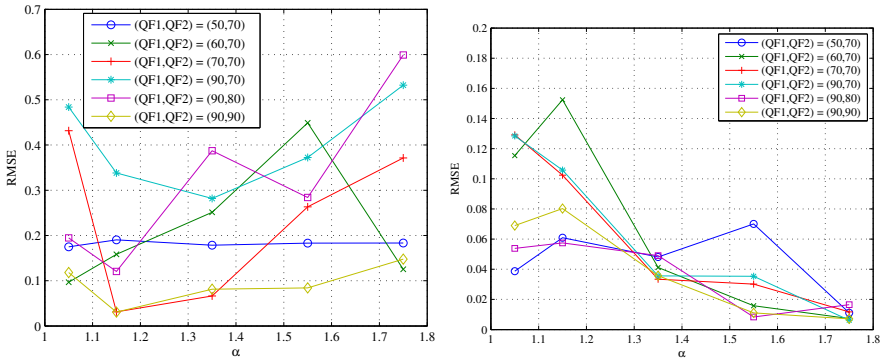
(a) Histogram based approach (b) MBFDF

Figure 3.2: *Accuracy of classification of QF_1, for different QF_2 and α values.*



(a) Histogram based approach (b) MBFDF

Figure 3.3: *Estimation of α: RMSE for different QF_1 and QF_2.*



(a) Histogram based approach (b) MBFDF

Figure 3.4: *Estimation of α values: RMSE for different fixed couples (QF_1, QF_2).*

couples (QF_1, α), as explained in 3.2.1, in the MBFDF based approach we trained a LDA classifier over all possible couples (QF_1, α), so that $N_C = 21 \times 6 = 126$, and the results are obtained by testing a subset of α, as in 3.3, i.e. $\alpha \in \{1.05, 1.15, 1.35, 1.55, 1.75\}$. From preliminary tests on histograms based approach, we fixed $n_c = 5$.

To evaluate estimation accuracy of QF_1, by fixing QF_2 and α, we define a *confusion matrix* for QF_1 as a matrix where each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class. By normalizing by the total number of instances, we obtain the percentage of decisions of each couple of classes. On the main diagonal, we have the percentage of correct decisions, for each value of QF_1. By averaging the percentage of correct decision over all values of QF_1 (i.e. values on the main diagonal), we obtain an average performance value of the classification of QF_1, for each couple (QF_2, α). We name this quantity *accuracy* of the estimate of QF_1.

To evaluate the estimation of α, we adopt the *root mean square error* (RMSE): let $\hat{\alpha}_{ij}$ with $i = 1, \ldots, N$ a set of estimated values of α_j, where $j = 1, \ldots, N_\alpha$ (i.e. $N_\alpha = 1$ when estimating a single value of α, otherwise $N_\alpha = 5$, equal to the number of tested α), we define the RMSE as:

$$RMSE = \sqrt{\frac{1}{N_\alpha \cdot N} \sum_{j=1}^{N_\alpha} \sum_{i=1}^{N} (\hat{\alpha}_{ij} - \alpha_j)^2} \tag{3.12}$$

	DCT Histograms			MBFDF		
Dimension	1024	256	64	1024	256	64
mean AUC	0.90	0.83	0.72	0.95	0.91	0.85
Accuracy of QF_1	0.52	0.51	0.45	0.86	0.77	0.66
RMSE of α	0.23	0.24	0.26	0.07	0.07	0.06

Table 3.3: *Performances of the proposed approaches for different image sizes (*1024×1024*,* 256×256 *and* 64×64*).*

The first comparison is about the *accuracy* of the classification of QF_1, by varying α and for different QF_2. The results shown in Figure 3.2, averaged over all QF_1 values, demonstrate that MBFDF based approach exhibits better performance than the histogram based one.

The second comparison is about the RMSE of the estimate of α, for each couple (QF_1, QF_2). The results presented in Figure 3.3, averaged over all α values, show again that MBFDF based approach has better performance than the histogram based one: the latter method shows performance almost comparable to the first one only when the second compression is greater than 90%, but it decreases for lower values of QF_2, whereas the performance of MBFDF remains good. To better understand this latter result, we evaluated the RMSE for different values of α, by fixing some couples of compression quality (QF_1, QF_2). It is possible to observe in Figure 3.4 that the histogram based approach gives results almost comparable to those obtained by MBFDF approach when $QF_2 \geq QF_1$, but its performance degrades quickly for $QF_2 \leq QF_1$; this behavior is well explained if we take into account the analysis done in 3.2.1, where we discussed the undetectability of the periodic pattern through spectrum analysis whenever $\tau^{'kl} < 1$, which corresponds to $QF_2 \leq QF_1$. As a last result, we show in Table 3.3 a comparison between the proposed approaches by varying the size of the image. Mean AUC values are obtained by averaging AUC values evaluated for each (QF_1, QF_2) in order to compare trained (i.e. MBFDF) and untrained (i.e. histogram based method) detectors, whereas *accuracy* of QF_1 and RMSE of α are calculated by mediating over all possible values of QF_1, QF_2 and α. As expected, the performance smoothly degrade by reducing the image size in both approaches, due to the lower number of available features for the detection and estimation procedures.

	DCT Histograms			MBFDF		
Dimension	1024	256	64	1024	256	64
Accuracy of QF_1	0.38	0.38	0.37	0.86	0.77	0.66
RMSE of α	0.30	0.31	0.29	0.07	0.07	0.06

Table 3.4: *Estimation of chain parameters: a comparison between the proposed approaches for different image sizes (*1024×1024, 256×256 *and* 64×64*) in case of untrained* α*.*

3.4 Conclusions

In this Chapter we have demonstrated how it is possible to detect the presence of a common image processing operation like contrast enhencement in the middle of a processing chain composed by two JPEG compressions. Two approaches previously developed to detect the presence of double compression have been properly modified to allow not only the detection, but also the estimation of the quality factor of the first JPEG compression and the parameter of the linear contrast enhancement. Each of the two methods has its own pros and cons: the approach based on the histogram of DCT coefficients has a low computational complexity, but exhibits good performance only when $QF_2 \geq QF_1$ and the second compression is mild; the method based on the distribution of the first digit of DCT coefficients has very good performance for every combination of quality factors, but its "one-against-one" multi-classification strategy may become impractical if a fine search of the processing parameter values is needed. These characteristics could suggest to use the histogram based approach when the image under analysis has a high compression quality, and resort to the other method when this property does not hold.

Part II

Image Mutations: from parents to child

Abstract

In this Part of the Thesis, we tackle with the problem of "mutations": some images are generated by combining contents coming from different images. In this way, new organisms come out with a new genetic makeup, which is different from the parent's one. So, we extend the Image Phylogeny to an Image Genealogy, also known as Multiple Parenting in Image Phylogeny, in which an image can have more than one parent. As we done in the first Part, we investigated also the scenario in which no information about parent images is available. In such a case, we developed an algorithm able to localize, within an image, regions whose content comes from other images. Since such an approach provides good performance only in well controlled scenarios, we extend this approach in order to build a framework based on a multi-clue analysis and data fusion techniques.

Chapter 4

Multiple Parenting Identification

Image phylogeny deals with tracing back parent-child relationships among a set of images sharing the same semantic content. The solution of this problem results in a visual structure showing the inheritance of semantic content among images giving rise to what is now called image phylogeny. In this Chapter, we extend upon the original image phylogeny formulation to deal with situations whereby an image may inherit semantic content not only from a single parent, as in the original phylogeny, but from multiple different parents, as commonly occurs during the frequent photomontage cases. We refer to this new scenario as multiple parenting phylogeny and we aim to represent the multiple parent relationships existent among a set of images. We propose a solution that starts from collecting near duplicate groups and reconstructing their phylogeny; then among the selected groups we identify the one(s) representing the composition images; finally, we detect the parenting relations between those compositions and their source images.

4.1 Introduction

Discovering multiple parenting relationships has many applications in practical scenarios, such as content tracking, forensics or copyright enforcement. As an example, we may consider pornographic compositions using personalities (such as celebrities or politicians), with the purpose of public shaming. By taking advantage of the large amount of images shared by users

(a) (b) (c)

Figure 4.1: *Example of a composite image (a), obtained by copying an object (the bear) from the alien image (b) onto an host image (c).*

and using multiple parenting phylogeny, it is possible not only to identify the image as a composition, but also to retrieve the source images used to create it. Knowing such sources serves as hard evidence that the pornographic image is a forgery, clearing the name of the victim. Finally, multiple parenting phylogeny can be easily extended to other types of media as well, such as texts, audio or videos, providing applications in those domains, such as plagiarism detection.

Multiple parenting phylogeny is a natural extension of the image phylogeny problem, allowing us to find the relationships not only between images with essentially the same content (near duplicates) but also those of seemingly unrelated content. This raises many challenges not present in image phylogeny, since we need to find relationships among images with no prior information about the amount of content they share. To do this, it is necessary to accurately reconstruct the phylogenies existent in a set of images, as well as precisely localize and compare the shared content between images. By overcoming those challenges, we can go even further in the analysis of the evolution of documents on the internet, and specially, how a new content is created by the combination of existing sources.

To find the multiple parenting relationships in a set of images, we introduce a method that works by grouping images into well-separated sets of near-duplicates, reconstructing their phylogenies, and pointing out which groups are compositions, finding the sources used to create them.

4.2 Method

The main objective of multiple parenting phylogeny is to discover the inheritance of content between compositions and their sources. In the most common scenario, we have three types of images: hosts and aliens (both the *source* images), and compositions, each one related to a near-duplicate set. The composition is the result of inserting a portion of an alien image into a host image, as shown in Figures 4.1. We can divide this particular scenario in three major problems: (1) determining the images depicting the same semantic content; (2) analyzing each group of near duplicates and pointing out whether the images therein are sources (images that could be used in a composition), compositions or unrelated to the rest; and finally, (3) inspecting each classified group and identifying the phylogeny relationship of the images therein and the ones actually used to create the compositions.

To solve the aforementioned problems, in this section, we introduce a 3-step method to automatically (1) find and (2) group near-duplicate images; and (3) classify nodes as host images (used as backgrounds in a composition), aliens (image pieces spliced with other images) or compositions (result of a combination of host and alien images). The following sections show details of each step of the method, whose full pipeline is depicted in Figure 4.2.

4.2.1 Finding near-duplicate groups

The problem of phylogeny forests arises when, in a set of images with the same semantic content, not all of them are associated by the same acquisition process (come from one original source). This happens when multiple pictures are taken from the same scene, with the same camera and different parameters or from the same scene with different cameras. In this case, the set of images will have multiple phylogeny trees, and a forest algorithm is responsible for identifying them. Dias et al. proposed [44] an approach based on the modification of their oriented Kruskal algorithm. This modification works by adding edges to a tree only if the weight of that edge is not higher than an adaptive threshold calculated on the edge weights already added to the solution. This threshold is dependent on the higher dissimilarity between images that have related content but are from different sources. Therefore, it is intuitive to see how a forest algorithm would also work in a scenario of different trees of unrelated semantic content.

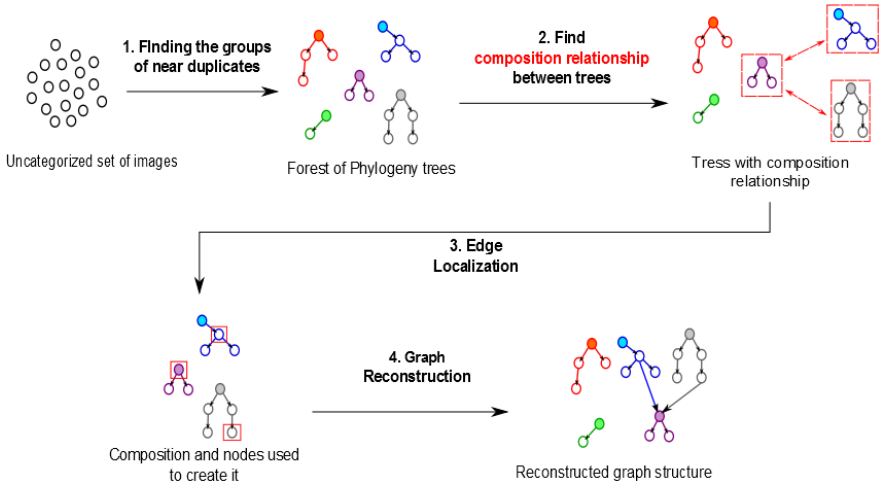


Figure 4.2: *Pipeline of our multiple parenting approach. Step 1 separates the images of the set in groups of similar semantic content, using a phylogeny forest algorithm. Step 2 looks for shared content between the scenes of each tree, classifying the trees in compositions and sources. Step 3 searches for the nodes used to generate the composition. Finally, the graph with the multiple parenting relationships is reconstructed.*

4.2.2 Group classification

After finding groups of near duplicates we still have no information about the relations between compositions and parent images. Also, the dissimilarity measure is unsuitable to discover those relations because it is strongly dependent on the type of composition (e.g., the size of the tampering region). A content-dependent descriptor, possibly invariant to geometrical, color and compression transformation, is needed to detect shared content among groups of images. To detect composition trees, we have adopted a SIFT-based approach [45].

For simplicity, we assume that a composition image is obtained by the composition of only two images, by copying a patch or portion cut from an image (*alien*) to another one (*host*). Moreover it is reasonable to assume that the patch is small with respect to the background, belonging to the host image.

Ideally, since each root of the tree obtained at the previous step summarizes the content of each tree, we perform a pairwise comparison between all possible combinations of the roots. The comparison is based on the extraction of keypoints and their SIFT descriptors, the matching of the keypoints as proposed by Lowe [46], and their clustering. In practice, however, this strategy proves to be error-prone and not sufficient: the presence of outliers coming from the matching strategy requires a robust clustering. Therefore, we adopt a J-Linkage clustering algorithm [47] to cluster keypoints in function of the estimated geometrical transformation applied to parent images. The method consists of generating a fixed number of geometrical transformation hypothesis by a random sampling of a neighborhood of matched keypoints. After that, for each pair of matched keypoints, a preference set vector (PS) is defined indicating which transformations the pair prefers. The PSs are used in a hierarchical agglomerative clustering to estimate the transformations. This algorithm starts by assigning each PS to a cluster; then, for each step of the algorithm, the two clusters with smallest distance are merged. The PS of a cluster is computed as the intersection of the preference sets of matched pairs, and the distance between two clusters is computed as the *Jaccard* distance between the respective preference sets.

J-Linkage presents some advantages: it is robust to the presence of outliers, it can be easily applied in case of more than two parent images and it does not need a priori information about the percentage of outliers, as in the case of RANSAC. This last property is suitable in our scenario because the number of outliers changes in function of the matched keypoints (i.e., on the content). For instance, when comparing images whose content is completely unrelated, all matched keypoints are outliers; conversely, when comparing a composition image with its parent image, the vast majority of the matched keypoints are inliers, rather than outliers. The main J-Linkage's limitation is the generation of small clusters of keypoints with degenerative models, due to the outliers. To reduce their impact, a threshold on the minimum number ($N_c = 5$) of keypoints satisfying the estimated transformation is applied.

Sometimes the aforementioned strategy fails because the composition is obtained from near duplicate of the roots (instead of the roots themselves), which have undergone a set of color, compression and geometrical transformations, altering the SIFT descriptors and making some matching undetectable. Therefore, we also extend the useful information from the roots to the trees, by randomly sampling an image (node) from each tree. The

test based on SIFTs is repeated a fixed number of times, and all those trees which have at least one image with more than one relation with other images is classified as composition tree.

Finally, to classify the parent trees as either alien or host, we employ the dissimilarities $d(I_A, I_B)$, evaluated at the previous step for the pairs composition-host and composition-alien roots. Since patches are small with respect to the background, the dissimilarity between host and composition root is lower than the dissimilarity between alien and composition ones.

4.2.3 Parents identification

In the previous step, relations between different groups of images are established, but they miss information about the exact sources that have generated the composition. Due to the different nature of the relations between host and composition groups and between alien and composition ones, we employ both dissimilarity measure and SIFT matching based approaches to trace the images which have exactly generated a composition. In the case of the host parent, we employ the dissimilarity $d(I_A, I_B)$ rather than a SIFT-based approach (which introduces a considerable computational effort), by observing that the host parent of the composition is the node of the host tree that has the lowest $d(I_A, I_B)$ with the composition root. This constraint is acceptable if we assume that the content coming from the alien parent is relatively small with respect to the background.

In case of the alien parent, we need to localize the shared content inside the composition root and evaluate the dissimilarity $d(I_A, I_B)$ only on that portion of content, to avoid noise due to the background belonging to the host. We use the same SIFT-based approach as in the previous step, by comparing all alien images with the related composition root. After identifying the cluster of matches between one of the aliens and the composition, we use the mean of the distances between the matches in the cluster as the dissimilarity between the shared content of both images. We select as the alien parent the one with the smallest dissimilarity among the tested nodes.

4.3 Experimental setup

This section presents the validation protocol for all experiments.

4.3.1 Dataset and Test Cases

The dataset[1] used comprises 100 host and 150 alien base images, as well as 5000 compositions. The host images are outdoor and indoor background scenes, such as rooms, streets or fields, obtained from the *Inria Holidays* [48] dataset. The aliens are images of varied objects, such as people, cars or animals, in common backgrounds. Those images were collected from *Berkley Segmentation* [49] and *Graz-02* [50] datasets, with segmentation masks from *Interactive Segmentation Tool* [51] and *Inria Annotations* [52], respectively.

The dataset also has a number of phylogeny tree files, associated with the base images, which describe a tree topology and the parameters of a set of image processing operations. Using the base image as root, the images are transformed following the topology and the operations described in order to generate the whole phylogeny tree. The operations and their parameter ranges were the same used in Dias et al. work [3]. All host and alien base images have 25 phylogeny trees (5 different topologies, with 5 parameter variations each) of 25 nodes. Because compositions are unique, they only have a single phylogeny tree of 25 nodes.

The test cases are phylogeny forests of 75 nodes, consisting of a host, an alien, and a composition tree. To generate one, we first randomly select a pair of host and alien base images, as well as two of their phylogeny tree files, and build the respective trees. Two random host and alien nodes are then picked from each tree to create the composition, by automatically cutting the object from the alien parent (using its segmentation mask) and pasting it randomly in the host parent. Composition types differ by pasting method, being either *direct pasting*, where the object is cut and pasted into the host with no changes whatsoever, or *poisson blending*, where the pasted object is blended into the host using Pérez et al.'s [53] method of gradient adjustment. Finally, the composition phylogeny tree is built, completing the generation of the test case. In this work, 300 direct pasting and 300 poisson blending test cases were used.

4.3.2 Metrics

To evaluate the accuracy of the groups and the reconstructed phylogeny forest, we use the metrics *roots*, *edges*, *leaves* and *ancestry* defined by Dias

[1]The dataset and test cases used in this work are available at http://dx.doi.org/10.6084/m9.figshare.1050094

et al. [2, 3] and a new *subset* metric, all measured considering we have the *groundtruth* forest. The *roots* metric checks whether the roots in the reconstructed and in the groundtruth forests are the same, while the *leaves* metric does the same for the leaves. The *edges* metric measures the percentage of right parenting relationships found in the reconstructed forest. Finally, the *ancestry* metric evaluates if each node in the reconstructed forest has the same set of ancestors as in the groundtruth. The *subset* measure, developed for this work, measures if images with the same semantic content end up in the same trees in the reconstructed forest, i.e., it measures if the image phylogeny forest algorithm correctly separates the images in meaningful groups. First, we define the set:

$$\begin{aligned}\sigma(F_R, F_{GT}) =& \{(I_A, I_B) | \pi(I_A, F_R) = I_B \quad \wedge \\ & \tau(I_A, F_{GT}) = \tau(I_B, F_{GT}), \\ & \forall I_A \in F_R \setminus \rho(F_R)\}\end{aligned} \tag{4.1}$$

where F_x, with $x \in \{R, GT\}$, is a reconstructed (R) or a ground truth (GT) forest, (I_A, I_B) is a generic couple of images and $\pi(I, F)$ is a function returning the parent of an image I in the forest F. Finally, $\tau(I, F)$ returns the tree to which the image I belongs in F, and $\rho(F)$ gives the roots of F. The *subset* metric is defined as:

$$subset = \frac{|\sigma(F_R, F_GT)|}{|F_R \setminus \rho(F_R)|} \tag{4.2}$$

The subset metric is important because it gives information about the *separation* of the host, alien and composition subset.

To evaluate the results of our multiple parenting approach we introduce the metrics *composition root* (CR), *host parent* (HP) and *alien parent* (AP), which test if such nodes were correctly found in each test case. Additionally, we employ the metrics *composition node* (CN), *host node* (HN) and *alien node* (AN), used to check if the composition root and host and alien parents are, respectively, composition, host and alien images. This second set of metrics is used to evaluate the classification of the trees.

4.4 Results and discussion

This section shows the experiments and results for multiple parenting identification. First, we present results for finding the groups and the phylogeny relations within each group. Then we show results for multiple parenting identification, rates with which the proposed method correctly classifies the trees in host/alien/composition, and its accuracy at identifying the nodes that generated compositions.

4.4.1 Forest Algorithm Results

Since there is no solution in the literature yet for the multiple parenting phylogeny problem, we consider two different forest algorithms in the experiments. The first one is a modification of the Oriented Kruskal [2, 3] to extract from a dissimilarity matrix exactly three trees, under the assumption that we know the number of trees in the forest, which we call K3T. The other is the automatic oriented Kruskal (AOK) as presented by Dias et al. [44], which tries to automatically identify the number of trees in the forest. As previously discussed, the AOK algorithm relies on a threshold parameter for adding new edges to the phylogeny forest. Using a smaller and completely separated set of 100 test cases, it was found that the best value for this parameter was 3.0. K3T is used just as an upper bound as in practice everything needs to be automatically calculated and we do not know the number of trees in the forest. It was also observed that, in most cases, the number of trees found by AOK was equal to or very close to three indicating that, even though AOK is automatic, it still has good results in finding the correct number of trees in the forest, making it a safe choice as the image grouping algorithm. Table 4.1 shows results for the reconstructed phylogeny forests, divided by direct pasting and poisson blending types of image composition.

Both algorithms show similar and good results, with K3T slightly better in the *roots* and *ancestry* metrics as expected. It is also important to note that K3T and AOK present nearly perfect results for the subset metrics which means that both algorithms are effective for separating the host, alien and composition trees. This is specially important for tree identification, as a bad separation of trees could lead to a wrong classification further on in the method.

Table 4.1: *Forest algorithm results for finding near-duplicate groups.*

		Metrics				
Type	**Algorithm**	**root**	**edges**	**leaves**	**ancestry**	**subset**
Direct	AOK	81.6%	74.3%	81.4%	65.5%	99.9%
	K3T	83.9%	74.4%	81.4%	66.6%	99.9%
Poisson	AOK	78.7%	74.5%	81.3%	63.2%	99.7%
	K3T	82.2%	74.6%	81.4%	66.1%	99.9%

4.4.2 Multiple Parenting Results

Table 4.2 shows the results for tree classification and multiple parenting identification. As detailed in Section 4.2.2, we classify the trees found by the forest algorithm applied in the first step by choosing random nodes of each tree and comparing their content to find shared objects, repeating this process a fixed number of times. The algorithm was tested with the number of repetitions: $\{1, 3, 5, 10, 15, 20, 25\}$. As there were no obvious gains of accuracy with more repetitions, it was decided to fix the number in five, as the computational cost tends to rise as more repetitions are used.

Table 4.2: *Multiple parenting results.*

		Metrics					
Type	**Algorithm**	**CR**	**CN**	**HP**	**HN**	**AP**	**AN**
Direct	**AOK**	73.0%	91.7%	76.0%	93.0%	33.7%	98.3%
	K3T	74.7%	92.7%	78.0%	94.3%	34.3%	99.0%
Poisson	**AOK**	66.3%	85.3%	73.0%	88.3%	11.3%	98.7%
	K3T	66.3%	87.0%	75.7%	88.3%	11.3%	99.3%

The algorithms present similar performance for the two types of compositions. Considering that in about 30% of the test cases AOK does not find three trees, those results are important to show that even when the number of trees found is incorrect, the classification of the trees, as shown by the *CN*, *HN* and *AN* metrics, still presents good accuracy. This is due to the robust process of classification that counts the number of content relationships between different trees in the forest, which keeps valid even if a tree is broken into sub-trees. When the composition tree is split into two trees, the low dissimilarity between the two might lead to wrongly classifying one of them as host tree. However, by comparing AOK with K3T results, those cases have small impact on the overall accuracy of the method.

We have good results in finding the original composition and its host parent, as shown by the CR and HP results. The CR value, in special, is dependent on the roots found by the forest algorithm, as we always choose the root of the tree identified as composition as the original composition. Finally, even though the proposed method shows very good results in identifying the alien tree, as in about 99% of the cases the alien parent identified is one of the alien nodes, we still are not very good at finding the correct alien node used in the composition process. As discussed before, we currently use the SIFT distance of the shared content existent between composition and alien as the comparison metric. This measure is not perfect at identifying the transformations the shared content went through, which might lead to a wrong classification.

4.5 Conclusion

In this Chapter, we presented a novel method for the identification of multiple parenting relationships in sets of images. It combines a phylogeny forest approach for group separation with object detection techniques for identification of shared content between images. Using this pipeline, the final result is a graph structure showing both the relationships between images with the same semantic and images with partially shared content.

The proposed method shows promising results in finding the different semantic groups (with an effectiveness exceeding 99%) and discovering the relationships between those groups (at least 85% of the cases), labeling them as compositions, hosts and aliens. Our future efforts will focus on finding the correct alien parent with a higher accuracy, by improving the estimation of the shared content region as well as our metrics to compare them, and expanding the proposed method to work with other types of compositions.

Chapter 5

Blind mutation detection: a case of study

In this Chapter, a forensic tool able to discriminate between original and forged regions in an image captured by a digital camera is presented. The main assumption is that the image was acquired using a Color Filter Array, as the majority of the modern digital camera does, and that tampering removes the artifacts due to the demosaicing algorithm. The proposed method is based on a new feature measuring the presence of demosaicing artifacts at a local level, and on a new statistical model, based on Gaussian Mixture Model, allowing to derive the tampering probability of each 2×2 image block without requiring to know a priori the position of the forged region. Experimental results on different cameras equipped with different demosaicing algorithms demonstrate both the validity of the theoretical model and the effectiveness of our scheme.

5.1 Introduction

In the recent literature, the forgery localization problem has been tackled with in different ways. A first class of forgery localization algorithms adopts a supervised approach, i.e., they rely on the hypothesis that a user has previously identified the location of possibly manipulated areas. Such a category includes all the tools analyzing inconsistencies at the scene level, like lighting, shadows [54], colors, geometry perspective [55], and those based on the computation of the difference of properly chosen statistics between

the possibly tampered area and the rest of the image [29].

Being fairly independent on the low-level characteristics of images, the above techniques (with some noticeable exception like [29]) are extremely robust to compression, filtering, and other image processing operations, thus being applicable even when the quality of the image is low. However, it is worth highlighting that being human assisted and based on rather stringent hypotheses, such techniques work only on restricted scenarios, and cannot be used and tested on massive amounts of data.

A first class of unsupervised forgery localization algorithms looks for the presence of tampered objects by decomposing the image under analysis into subparts. In region-wise approaches, the image is first segmented into homogeneous regions and then each region is analyzed separately [56]; in block-wise approaches, the image is split into sliding square windows, and each block is processed independently. Inconsistencies in the presence or the absence of specific footprints related to acquisition, coding, or editing within one or more sub-parts of the image indirectly reveal that some processing has been applied on a particular region of the image [57,58]. Concerning the limits of these methods, in the region-wise approach very often the segmentation does not produce reliable results without a priori information about the possible tampered area. In the block-wise approach, usually a sufficiently large portion of the image (e.g. a $B \times B$ block, with $B \geq 100$) is needed for a reliable statistical analysis of the footprint, so that only a coarse grained localization of tampering is possible.

A last class of unsupervised tamper localization algorithms is represented by forensic schemes designed to localize in an automatic way the tampered regions with a fine-grained scale of $B \times B$ image blocks (where usually $B = 8$), assuming to have no information on the position of possibly manipulated pixels. The output of these methods is a likelihood map indicating for each pixel (or small block) its probability of being tampered.

To the best of our knowledge, only few algorithms exploiting the presence of double JPEG compression [59–61] or the artifacts due to CFA interpolation [62] belong to this category. The main limit of these approaches is the strong dependence of the results on local and global properties of the image (content, dimension, compression etc) and by the noiseness of the output map, so that it is always necessary to apply a postprocessing (often assisted) phase to obtain reliable results.

In this Chapter, we focus our attention on the fine grained forgery local-

ization problem, assuming to have no information on the position of possibly manipulated pixels. Among the numerous fingerprints considered in image forensic literature [63, 64], we consider the traces left by the *interpolation* process. Image interpolation is the process of estimating values at new pixel locations by using known values at neighbouring locations. During the image life cycle, there are two main phases in which interpolation is applied:

- Acquisition processing, to obtain the 3 color channels (red, green, and blue). The light is filtered by the *Color Filter Array* (CFA) before reaching the sensor (CCD or CMOS), so that for each pixel only one particular color is gathered. Thus, starting from a single layer containing a mosaic of red, green, and blue pixels, the missing pixel values for the three color layers are obtained by applying the interpolation process, also referred to as *demosaicing*.

- Geometric transformations, to obtain a transformed image. When scaling (shrinking and zooming), rotation, translation, shearing, are applied to an image, pixels within the to-be-transformed image are relocated to a new lattice, and new intensity values have to be assigned to such positions by means of interpolation of the known values, also referred to as *resampling* operation.

The artifacts left in the image by the interpolation process can be analyzed to reveal image forgery. Ideally, an image coming from a digital camera, in the absence of any successive processing, will show demosaicing artifacts on every group of pixels corresponding to a CFA element. On the contrary, demosaicing inconsistencies between different parts of the image, as well as resampling artifacts in all or part of the analyzed image, will put image integrity in doubt.

Our effort is focused on the study of demosaicing artifacts at a local level: by means of a local analysis of such traces we aim at localizing image forgeries whenever the presence of CFA interpolation is not present. Obviously our approach is based on the hypothesis that unmodified images coming from a digital camera are characterized by the presence of CFA demosaicing artifacts. Starting from such an assumption, we propose a new feature that measures the presence/absence of these artifacts even at the smallest 2×2 block level, thus providing as final output a forgery map indicating with fine localization the probability of the image to be manipulated.

5.2 Related Work

Previous works considering CFA demosaicing as the to be analyzed fingerprint can be divided in two main classes, i) algorithms aiming at estimating the parameters of the color interpolation algorithm, and ii) algorithms aiming at evaluating the presence/absence of demosaicing traces. Whereas the second class focuses on forgery detection (inconsistencies in the CFA interpolation reveal the presence of forged regions), algorithms within the first class are mostly intended to classify different source cameras, though sometimes they can also be used to detect tampering.

As to the first class, Swaminathan et al. in [65] propose a method for camera identification by the estimation of the CFA pattern and interpolation kernel; while in [66] the same authors exploit the inconsistencies among the estimated demosaicing parameters as proof of tampering. Cao and Kot in [67] aim at estimating the demosaicing formulas, employing a partial second-order image derivative correlation model, in order to classify different demosaicing algorithms. In [68], Bayram et al. detect and classify traces of demosaicing by jointly analyzing features coming from two previous works (see [69] and [70] below), in order to identify the source camera model. In [71], Fan et al. propose a neural network framework for recognizing the demosaicing algorithms in raw CFA images, and use it for digital photo authentication.

Regarding the detection of demosaicing traces, Popescu and Farid propose an approach for detecting the interpolation artifacts left on digital images by resampling [25] and demosaicing [69] processes. In their approach, the Expectation-Maximization algorithm is applied to estimate the interpolation kernel parameters, and a probability map is achieved that for each pixel provides its probability to be correlated to neighbouring pixels. The presence of interpolated pixels results in the periodicity of the map that is clearly visible in the Fourier domain. Such an analysis can be applied to a given image region, however a minimum size is needed for assuring the accuracy of the results: authors tested their algorithms on 256×256 and 512×512 sized areas.

Gallagher in [70] observed that the variance of the second derivative of an interpolated signal is periodic: he thus looked for the periodicity in the second derivative of the overall image by analyzing its Fourier transform. Successively, for detecting traces of demosaicing, Gallagher and Chen pro-

posed in [72] to apply Fourier analysis to the image after high pass filtering, for capturing the presence of periodicity in the variance of interpolated/acquired coefficients. The procedure has been tested only up to 64×64 image blocks, whereas a variation yielding a pixel-by-pixel tampering map is based on a 256-point discrete Fourier transform computed on a sliding window, thus lacking resolution.

In [73] by Dirik and Memon, the sensor noise power of the analyzed image is taken into account: its change across the image (i.e. its difference value for interpolated and acquired pixels) is considered for demonstrating the presence of demosaicked pixels. In the above paper, a block based CFA detection was also proposed, however the features proposed therein have to be computed on 96×96 blocks, thus permitting only a coarse grained localization of tampering.

Demosaicing can also be detected using methods which analyze generic resampling artifacts. In this area, Kirchner in [74, 75] consider an approach similar to [25], by observing that the actual prediction weights of the resampling filter are not necessary for revealing periodic artifacts, thus simplifying the analysis, however experimental results consider only 512×512 images. Mahdian and Saic in [76] consider the derivatives of the interpolated image and apply the method to suspected windows of size at least 64×64, while in [77] they adopt the spectral correlation function, but only considering 512×512 sized images. Finally, in [78] Vazquez-Padin et al. demonstrate that the interpolated image is an almost cyclostationary process, with a period depending on the resampling factor. However, the authors use image blocks of size 128×128 pixels for the analysis, which only permits a coarse forgery localization.

5.3 CFA Modeling

During the CFA interpolation process, the estimation of the values in the new lattice based on the known values can be locally approximated as a filtering process through an interpolation kernel periodically applied to the original image to achieve the resulting image. Thus, the identification of artifacts due to CFA demosaicing can be seen as a particular case of the detection of interpolation artifacts, that has been deeply studied in these last years, as exposed in Section 5.2.

In [74], Kirchner demonstrated that for a resampled stationary and non-

constant signal $s(x)$, with $x \in \mathbb{Z}$, the variance of the residue of a linear predictor $\mathrm{Var}[e(x)]$ is periodic with a period equal to the original sampling rate. Hence, if we consider the signal resampled according to an integer interpolation factor r, we have $\mathrm{Var}[e(x)] = \mathrm{Var}[e(x+r)]$, since the original sampling period corresponds to r samples of the resampled signal.

For the case of CFA demosaicing, if we consider a single dimension, the general result presented in [74] turns into $\mathrm{Var}[e(x)] = \mathrm{Var}[e(x+2)]$, that is the variance of the prediction error assumes only two possible values, one for the odd positions and another one for the even positions. In more detail, considering for example the interpolation of the green color channel $G(x)$ in a particular row of the image, the acquired signal $s_A(x)$ is

$$s_A(x) = \begin{cases} G(x) & x \text{ even} \\ 0 & x \text{ odd} \end{cases} \tag{5.1}$$

If we consider a simplified demosaicing model, the resulting signal $s_R(x)$, composed by the acquired component $s_A(x)$ and by the interpolated component, takes values:

$$s_R(x) = \begin{cases} s_A(x) = G(x) & x \text{ even} \\ \sum_u h_u s_A(x+u) & x \text{ odd} \end{cases} \tag{5.2}$$

where h_u represents the interpolation kernel. In the above model, we assume that each color channel is independently interpolated using a linear filter and that original sensor samples are not modified by the interpolation process[1]. In practice, since only odd values of u contribute to the above summation, we will restrict our attention to the case $h_u = 0$ for u odd. The prediction error is then defined as $e(x) = s_R(x) - s_P(x)$, with:

$$s_P(x) = \sum_u k_u s_R(x+u) \tag{5.3}$$

the predicted signal, and k_u the prediction kernel. Hence:

$$e(x) = \begin{cases} G(x) - \sum_u k_u s_R(x+u) & x \text{ even} \\ \sum_u h_u s_A(x+u) - \sum_u k_u s_R(x+u) & x \text{ odd} \end{cases} \tag{5.4}$$

[1]The first assumption is often not verified in practice, however the second one usually holds since most practical demosaicing algorithms do not change the value of acquired pixels.

By assuming to use the same kernel for the interpolation and the prediction (i.e. $h_u = k_u$), the prediction error in odd positions is identically zero, while in the even positions takes values different from zero. Hence, in such an ideal case, var$[e(x)]$ is expected to be zero in the positions corresponding to the demosaicked signal, and different from zero in the positions corresponding to the acquired signal.

In general, the exact interpolation coefficients may not be known, however we can assume that $k_u = 0$ for u odd. Moreover, we can also assume $\sum_u k_u = \sum_u h_u = 1$, which usually holds for common interpolation kernels. In this case, equation (5.4) above can be rewritten as

$$e(x) = \begin{cases} G(x) - \sum_u k_u \sum_v h_v G(x+u+v) & x \text{ even} \\ \sum_u (h_u - k_u) G(x+u) & x \text{ odd} \end{cases} \tag{5.5}$$

By assuming the acquired signal samples to be independent and identically distributed (i.i.d.) with mean μ_G and variance σ_G^2, the mean of the prediction error can be evaluated as

$$E[e(x)] = \begin{cases} \mu_G - \mu_G \sum_u k_u \sum_v h_v = 0 & x \text{ even} \\ \mu_G \left(\sum_u h_u - \sum_u k_u\right) = 0 & x \text{ odd} \end{cases} \tag{5.6}$$

whereas the variance of the prediction error is

$$\begin{aligned} \mathrm{Var}[e(x)] =& \mathrm{Var}\Big[(1 - \sum_u k_u h_{-u}) G(x) \\ &+ \sum_{t \neq 0} \Big(\sum_u k_u h_{t-u} \Big) G(x+t) \Big] \\ =& \sigma_G^2 \Big[(1 - \sum_u k_u h_{-u})^2 + \sum_{t \neq 0} (\sum_u k_u h_{t-u})^2 \Big] \end{aligned} \tag{5.7}$$

for x even and

$$\mathrm{Var}[e(x)] = \mathrm{Var}\Big[\sum_u (h_u - k_u) G(x+u) \Big] = \sigma_G^2 \sum_u (h_u - k_u)^2 \tag{5.8}$$

for x odd. According to the above model, the prediction error has zero mean and variance proportional to the variance of the acquired signal. However, when the prediction kernel is close to the interpolation kernel, the variance of prediction error will be much higher at the positions of the acquired pixels than at the positions of interpolated pixels.

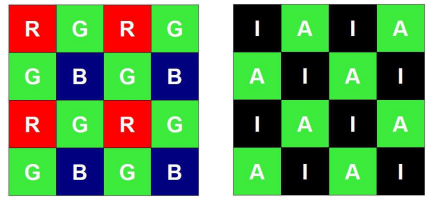


Figure 5.1: *(a) the Bayer's filter mosaic; (b) the quincunx lattice $\mathcal{A}$ for the acquired green channels and the complementary quincunx lattice $\mathcal{I}$ for the interpolated green channels.*

Leaving the ideal conditions, the acquired signal will be only locally i.i.d. and its variance only locally stationary: thus σ_G^2 has to be computed on small parts of the signal and consequently var$[e(x)]$ will assume different values depending on the specific signal content. Also, additive noise may be present on pixel values due to rounding and truncation effects. Nevertheless, we can still expect the variance of $e(x)$ to be higher at the positions of acquired pixels.

5.4 Proposed algorithm

In order to extend the previous analysis to the bidimensional case, without loss of generality we will consider as specific CFA the most frequently used Bayer's filter mosaic, a 2×2 array having red and green filters for one row and green and blue filters for the other (see Fig. 5.1(a)). Furthermore, we will consider only the green channel; since the green channel is upsampled by a factor 2, for a generic square block we have the same number of samples (and the same estimation reliability) for both classes of pixels (either acquired or interpolated).

By focusing on the green channel, the even/odd positions (i.e. acquired/interpolated samples) of the one-dimensional case turn into the quincunx lattice $\mathcal{A}$ for the acquired green values and the complementary quincunx lattice $\mathcal{I}$ for the interpolated green values (see Fig. 5.1(b)). Similar to the one-dimensional case, we assume that in the presence of CFA interpolation the variance of the prediction error on lattice $\mathcal{A}$ is higher than the variance of the prediction error on lattice $\mathcal{I}$, and in both cases it is content dependent. On the contrary, when no demosaicing has been applied, the variance

of the prediction error assumes similar values on the two lattices. Hence, in order to detect the presence/absence of demosaicing artifacts, it is possible to evaluate the imbalance between the variance of the prediction error in the two different lattices.

5.4.1 Proposed feature

Let us suppose that $s(x, y)$, with $(x, y) \in \mathbb{Z}^2$, is an observed image. The prediction error can be obtained as:

$$e(x, y) = s(x, y) - \sum_{u,v \neq 0} k_{u,v} s(x + u, y + v) \tag{5.9}$$

where $k_{u,v}$ is a bidimensional prediction filter. In the ideal case, $k_{u,v} = h_{u,v}$ $\forall (u, v)$ where $h_{u,v}$ is the interpolation kernel of the demosaicing algorithm. In general, we can assume that $k_{u,v} \neq h_{u,v}$, since the in-camera demosaicing algorithm is usually unknown.

Because of the local stationarity of the residue, the variance of the prediction error $e(x, y)$ is *locally* estimated pixel-by-pixel for each position (demosaicked or acquired) only from a neighborhood of interpolated ($\mathcal{I}$) or acquired ($\mathcal{A}$) pixels respectively. In this work, we assume to know the spatial pattern of the CFA (for example the Bayer CFA). This hypothesis is not a serious constraint, because it is reasonable to suppose either to know the CFA pattern or to estimate it by adopting a proper estimation algorithm [65].

By assuming that the local stationarity of prediction error is valid in a $(2K + 1) \times (2K + 1)$ window, it is possible to define the local weighted variance of the prediction error as:

$$\sigma_e^2(x, y) = \frac{1}{c}\Big[\Big(\sum_{i,j=-K}^{K} \alpha_{ij}\, e^2(x + i, y + j)\Big) - (\mu_e)^2\Big] \tag{5.10}$$

where α_{ij} are suitable weights, $\mu_e = \sum_{i,j=-K}^{K} \alpha_{ij}\ e(x + i, y + j)$ is a local weighted mean of the prediction error and $c = 1 - \sum_{i,j=-K}^{K} \alpha_{ij}^2$ is a scale factor that makes the estimator unbiased, i.e., $E[\sigma_e^2(x, y)] = \text{var}[e(x, y)]$, for

each pixel class. The weights α_{ij} are obtained as $\alpha_{ij} = \alpha'_{ij} / \sum_{i,j} \alpha'_{ij}$ where

$$\alpha'_{ij} = \begin{cases} W(i,j) & \text{if } e(x+i, x+j) \text{ belongs to} \\ & \text{the same class of } e(x,y) \\ 0 & \text{otherwise} \end{cases}$$

and $W(i,j)$ is a $(2K+1)\times(2K+1)$ Gaussian window with standard deviation $K/2$.

Given a $N \times N$ image, we analyze it by considering $B \times B$ non-overlapping blocks, where B is related to the period of Bayer's filter mosaic: the smallest period (and block dimension) is $(2,2)$, but also multiples can be adopted. The generic block in position (k,l) is denoted as $\mathcal{B}_{k,l}$ with $k,l = 0,\ldots,\frac{N}{B}-1$. Each block is composed by disjoint sets of acquired and interpolated pixels, indicated as $\mathcal{B}_{A_{k,l}}$ and $\mathcal{B}_{I_{k,l}}$, respectively. We then define the feature $\mathbf{L}$:

$$\mathbf{L}(k,l) = \log\left[\frac{GM_A(k,l)}{GM_I(k,l)}\right] \tag{5.11}$$

where $GM_A(k,l)$ is the *geometric mean* of the variance of prediction errors at acquired pixel positions, defined as:

$$GM_A(k,l) = \left[\prod_{i,j \in \mathcal{B}_{A_{k,l}}} \sigma_e^2(i,j)\right]^{\frac{1}{|\mathcal{B}_{A_{k,l}}|}} \tag{5.12}$$

whereas $GM_I(k,l)$ is similarly defined for the interpolated pixels.

The proposed feature $\mathbf{L}$ allows us to evaluate the imbalance between the local variance of prediction errors when an image is demosaicked: indeed, in this case the local variance of the prediction error of acquired pixels is higher than that of interpolated pixels and thus the expected value of $\mathbf{L}(k,l)$ is a nonzero positive amount. On the other hand, if an image is not demosaicked, this difference between the variance of prediction errors of acquired an interpolated pixels disappears, since the content can be assumed to present locally the same statistical properties, and the expected value of $\mathbf{L}(k,l)$ is zero. Our inference will be based on these two key observations.

Let us now suppose that a demosaicked image has been tampered by introducing a new content, and that in order to make this forgery more realistic, some processing (blurring, shearing, rotation, compression, etc.) has

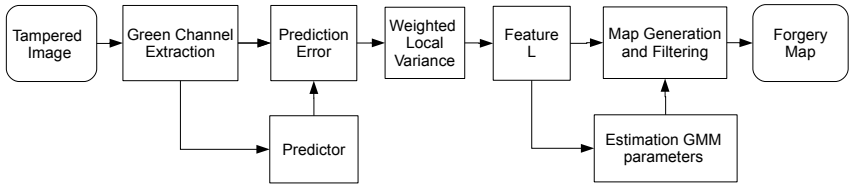


Figure 5.2: *The work flow of our algorithm.*

been likely applied to the added content, thus destroying the demosaicing traces on the forged region. The proposed feature $\mathbf{L}(k,l)$ will assume inconsistent values within the tampered image: in some regions (the untampered ones) it will be significatively greater than zero, while in other regions (the tampered ones) the feature will be close to zero. We can thus employ these inconsistencies to finely localize forgeries.

In some respects, the proposed feature is conceptually similar to the approach in [72], where the variance is approximated using the average of absolute values. However, an important difference is that the technique of [72] requires a Fourier analysis, thus limiting the resolution of the method when aiming at the fine-grained localization of CFA artifacts. Moreover, the proposed feature can be described using a very convenient statistical model, described in the following, which allows us to associate to each block a probability of being manipulated.

5.4.2 Feature modeling

By using a *Bayesian approach*, for each block $\mathcal{B}_{k,l}$ it is possible to derive the probability that CFA artifacts are present/absent conditioned on the observed values of $\mathbf{L}(k,l)$.

Let M_1 and M_2 be the hypotheses of presence and absence of CFA artifacts, respectively. In order to have a simple and tractable model, we assume that $\mathbf{L}(k,l)$ is Gaussian distributed under both hypotheses and for any possible size B of the blocks $\mathcal{B}_{k,l}$. For a fixed B, we can characterize our feature using the following *conditional probability density functions*:

$$Pr\{\mathbf{L}(k,l)|M_1\} = \mathcal{N}(\mu_1, \sigma_1^2) \tag{5.13}$$

with $\mu_1 > 0$, and

$$Pr\{\mathbf{L}(k,l)|M_2\} = \mathcal{N}(0, \sigma_2^2). \tag{5.14}$$

The above densities hold $\forall k, l = 0, \ldots, \frac{N}{B} - 1$, i.e., we assume that the parameters of the two conditional pdfs do not change over the considered image, such that they can be globally estimated.

If a demosaicked image contains some tampered regions in which CFA artifacts have been destroyed (as it may occur in a common splicing operation), both hypotheses M_1 and M_2 are present, therefore $\mathbf{L}(k,l)$ can be modeled as a mixture of Gaussian distributions. The first component, with $\mu_1 > 0$, is due to regions in which CFA artifacts are present, whereas the second component, with $\mu_2 = 0$, is due to tampered regions in which CFA artifacts have been removed[2]. In order to estimate simultaneously the parameters of the proposed Gaussian Mixture Model (GMM), we employ the *Expectation-Maximization (EM) algorithm* [79]. This is a standard iterative algorithm that estimates the mean and the variance of the component distributions by maximizing the expected value of a *complete log-likelihood function* with respect to the distribution parameters. In our case, the EM algorithm is used to estimate only μ_1, σ_1, and σ_2, since we assume $\mu_2 = 0$.

5.4.3 Map generation

The final aim we point at is to achieve a map indicating for each $B \times B$ block $\mathcal{B}_{k,l}$ its probability to be original/tampered, based on its probability to contain or not CFA artifacts. Starting from equations (5.13) and (5.14) and assuming a-priori probabilities $Pr\{M_1\} = Pr\{M_2\} = 1/2$, we obtain the *posterior probability* of being an original block. By exploiting Bayes' Theorem and relying on the observed feature $\mathbf{L}(k,l)$ for each $\mathcal{B}_{k,l}$ block, we achieve:

$$Pr\{M_1|\mathbf{L}(k,l)\} = \frac{Pr\{\mathbf{L}(k,l)|M_1\}}{Pr\{\mathbf{L}(k,l)|M_1\} + Pr\{\mathbf{L}(k,l)|M_2\}} \tag{5.15}$$

[2]The above model may not be accurate in the case of copy-move forgeries exhibiting a nonaligned CFA pattern, since these areas will result in negative values of $\mathbf{L}(k,l)$. However, this is only a small subset of the possible forgeries and it does not appears reasonable to complicate the model to cope with this particular case.

which can be expressed as:

$$Pr\{M_1|\mathbf{L}(k,l)\} = \frac{1}{1+\mathcal{L}(\mathbf{L}(k,l))} \tag{5.16}$$

where $\mathcal{L}$ is the *likelihood ratio* of $\mathbf{L}(k,l)$ defined as:

$$\mathcal{L}(\mathbf{L}(k,l)) = \frac{Pr\{\mathbf{L}(k,l)|M_2\}}{Pr\{\mathbf{L}(k,l)|M_1\}}. \tag{5.17}$$

Let us note that equations (5.16) and (5.17) have the same statistical information. Applying equation (5.17) to each block of an image, we obtain a *likelihood map* (LM), where each pixel of the map is the likelihood ratio associated to a $B \times B$ block.

These maps are usually noisy because they associate a probability (or a likelihood ratio) value to a single realization of $\mathbf{L}(k,l)$, which is very noisy itself. In order to denoise these maps, we can cumulate feature values on larger blocks whose size is $C \times C$, where $C = n \cdot B$ with $n \in \mathbb{Z}^+$. Assuming blocks to be conditionally independent given either M_1 or M_2, the accumulated likelihood ratio is obtained as:

$$\mathcal{L}_{cum}(\mathbf{L}(k',l')) = \frac{\prod_{k,l} Pr\{\mathbf{L}(k,l)|M_2\}}{\prod_{k,l} Pr\{\mathbf{L}(k,l)|M_1\}}. \tag{5.18}$$

In order to further improve the localization performance, we note that in a realistic forged image the manipulated areas are usually connected regions, due to the image semantic content. These connected regions can be highlighted by applying to the map a simple low-pass spatial filter, like a mean filter or a median filter. For better numerical stability, such filters are applied to the logarithm of the likelihood map.

5.4.4 Overall system

In Fig. 5.2 we show the overall system that, given a suspected image, produces the corresponding forgery map: each pixel in the forgery map indicates for each $C \times C$ image block its probability to contain CFA artifacts, so that low values in the output map correspond to likely forged areas.

As a first step, the green channel is extracted from the image, and the prediction error is computed. Because in-camera processing algorithms are usually unknown, a fixed predictor is used: the choice of the adopted pre-

dictor will be discussed and validated in Section 5.5. The weighted local variance is then estimated and the feature $\mathbf{L}(k,l)$ is obtained for each $B \times B$ block. The GMM parameters are globally estimated exploiting the EM algorithm and used for the generation of the forgery map. When $C = B$ the forgery map is generated using the likelihood ratios in (5.17), whereas for $C > B$ we use the cumulated likelihood map in (5.18). Optionally, the intermediate log-likelihood map can be filtered using either a mean filter or a median filter.

5.5 Experimental Results

The results presented in this section have been obtained on a dataset consisting of 400 original color images, in TIFF uncompressed format, coming from 4 different cameras (100 images for each camera): Canon EOS 450D, Nikon D50, Nikon D90, Nikon D7000. All cameras are equipped with a Bayer CFA, thus respecting our requirement that authentic images come from a camera leaving demosaicing traces, but the in-camera demosaicing algorithms of such devices are unknown. Each image was cropped to 512×512 pixels, maintaining the original Bayer pattern, which is assumed to be known[3]. We will refer to such a dataset as the *original dataset*.

5.5.1 Model Validation

The first step was to verify the assumption of Gaussian distribution on $\mathbf{L}(k,l)$, both in the presence and in the absence of CFA artifacts. To this end, starting from 100 images selected from the *original dataset*, we have created two datasets satisfying the M_1 (presence of CFA) and M_2 (absence of CFA) hypotheses. To create the dataset corresponding to M_1, the original images have been sampled according to the Bayer CFA pattern and then re-interpolated using four possible demosaicing algorithms, namely bilinear, bicubic, gradient-based and median (see [69] for more details on such interpolation algorithms). This allowed us to know the interpolation kernel on the whole image, and then to exactly predict the interpolated values with the four different predictors (we refer to these cases as 'ideal'). To create the dataset corresponding to M_2, each color channel of the original images

[3]The correct CFA configuration has been verified by inspecting the technical specifications of the raw image format.

has been upsampled by a factor two, blurred with a 7×7 median filter, and downsampled by a factor two, thus removing all CFA artifacts. Features are then computed using again the four predictors as before.

Moving towards realistic conditions, we also computed the value of $\mathbf{L}(k,l)$ under the M_1 hypothesis on the *original dataset* of 400 TIFF uncompressed images interpolated using their unknown in-camera demosaicing algorithms, and applying bilinear, bicubic, gradient-based and median predictors.

We verified the approximate Gaussian distribution of the features for all the classes described so far, i.e.: absence of CFA, presence of CFA with known interpolation kernel, and the four sets of cameras with unknown CFA demosaicing algorithms; for each of these six classes, the features have been computed with the four different interpolation algorithms (bilinear, bicubic, gradient-based, median) setting $B = 8$. The approximately Gaussian behavior of the features has been verified by fitting them with a generalized Gaussian distribution (GGD), given by

$$p(\mathbf{L}) = \frac{1}{Z} e^{-(|\mathbf{L}-\mu|/\eta)^{\nu}} \tag{5.19}$$

where μ is a location parameter (mean), η is a scale parameter, ν is a shape parameter, and Z is a normalization factor so that $p(\mathbf{L})$ integrates to one. The Gaussian distribution is a particular case of the GGD for $\nu = 2$. Since our conjecture is that the Gaussian assumption holds for a single image, but not necessarily over the whole dataset, the shape parameter has been independently estimated for each image using the Mallat's method [80]. In Table 5.1 we report the median value of the estimated shape parameters for the six classes and the four interpolation algorithms. The values indicate a reasonable fit of the proposed model. Interestingly, the model appears more fitting in the presence of CFA artifacts, and when the predictor is matched to the actual interpolation algorithm.

Furthermore, we plot the mean value of the features in order to verify how features in M_1 hypothesis can be discriminated by features in M_2 hypothesis, both in ideal and in realistic cases. In Fig. 5.3, we show the results for the ideal case in absence of CFA (first row) and presence of known CFA (second row). In Fig. 5.4, we show the 16 histograms of the mean values of $\mathbf{L}(k,l)$: along each row we have histograms referring to the same camera, from top to bottom, Canon EOS 450D, Nikon D50, Nikon D90, Nikon D7000. For both the Figures along each column we have histograms referring to the same

Table 5.1: *Median value of the GGD shape parameters estimated from the distribution of the feature* $\mathbf{L}(k,l)$ *for each image, considering different predictors on different datasets.*

	bilinear	bicubic	gradient-based	median
No CFA	1.589	1.558	1.672	1.812
Ideal	2.168	2.134	2.049	2.016
Canon EOS	2.001	1.908	1.897	1.962
Nikon D50	1.736	1.797	1.834	1.814
Nikon D7000	2.206	2.066	1.729	1.899
Nikon D90	1.998	1.924	1.667	1.927

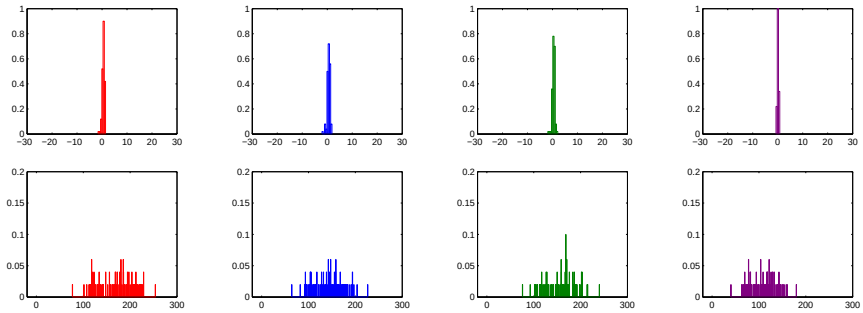


Figure 5.3: *Distribution of the average value of* $\mathbf{L}(k,l)$ *on an image, feature evaluated on* 8×8 *blocks, in the absence of CFA artifacts (top row) and when the predictor is the same as the demosaicing algorithm (bottom row), using different predictors: from left to right, bilinear (red), bicubic (blue), gradient-based (green), median (violet).*

predictor, from left to right, bilinear (red), bicubic (blue), gradient-based (green), median (violet).

Globally, the above results confirm that the proposed features has zero mean under the M_2 hypothesis and mean greater than zero under the M_1 hypothesis. The histograms also highlight that the four predictors have different behaviors. The median predictor does not seem well suited to detect CFA artifacts, since it produces values of $\mathbf{L}(k,l)$ closer to zero than the other predictors, irrespective of the camera.

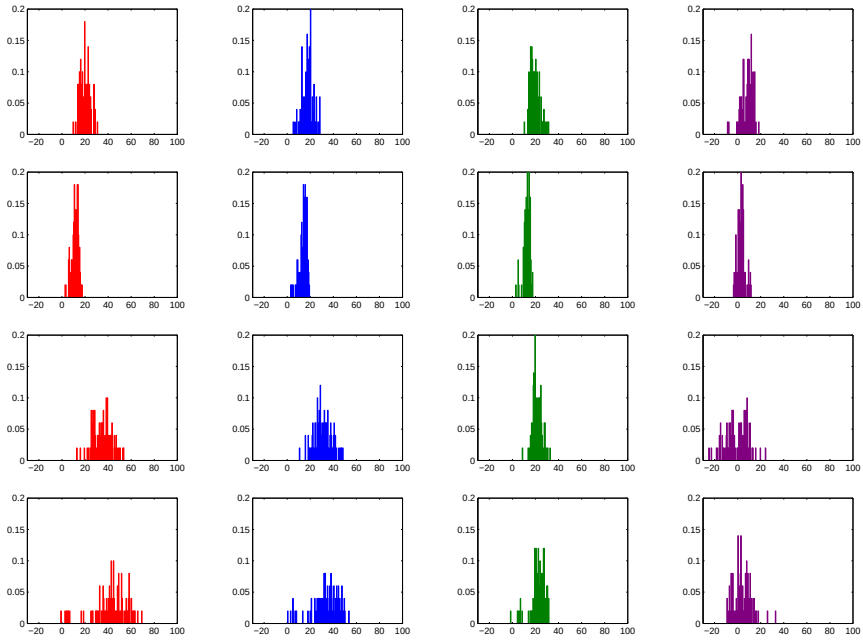


Figure 5.4: *Distribution of the average value of* $\mathbf{L}(k,l)$ *on an image, feature evaluated on* 8×8 *blocks, with unknown in-camera demosaicing algorithms and using different predictors: along each row we have histograms referring to the same camera, from top to bottom, Canon EOS 450D, Nikon D50, Nikon D7000, Nikon D90; along each column we have histograms referring to the same predictor, from left to right, bilinear (red), bicubic (blue), gradient-based (green), median (violet).*

5.5.2 Detection Performance Validation

In this section, the detection capability of the proposed forgery localization algorithm is investigated. Firstly, the behavior with respect to different predictors is analyzed. Then, in order to characterize the algorithm performance in different conditions, a particular predictor is chosen – the bilinear – and the results are evaluated considering different scenarios, different forgery sizes, and different choices of algorithm parameters.

Experimental Methodology

The considered scenarios correspond to nine different datasets derived from the *original dataset*: a first group of four datasets include uncompressed images obtained by applying bilinear, bicubic, gradient-based, and median demosaicing (as described in the previous section), representing the ideal case; a second group of five datasets include uncompressed images obtained using the demosaicing algorithm of the respective four cameras and JPEG compressed images obtained from the previous images using four different quality factors: 100%, 95%, 90% and 85%. The idea underlying this choice is to verify the performance on sets of images that completely satisfy the requirements of the proposed model as well as on more realistic images.

For each dataset, forgery has been simulated by applying to the central region of the image the procedure for removing CFA artifacts described in the previous section. As to the size of the forgery, we considered tampered regions of 128×128, 64×64, and 32×32 pixels. In the case of JPEG datasets, CFA removal has been simulated before JPEG compression.

The analysis has been carried out under different *resolutions* and *filtering* of the *likelihood map*. Concerning the *resolution*, in order to permit a fine-grained localization of the tampered regions, we chose to compute the proposed metric $\mathbf{L}$ starting from 2×2 blocks ($B = 2$), the smallest possible size to detect CFA artifacts. Different resolutions, equivalent to 4×4 blocks and 8×8 blocks, can be obtained in two ways: the first one is to define our features on larger blocks (e.g. $B = 4$ or $B = 8$). The second way is to compute the proposed metric on 2×2 or 4×4 blocks, and then to cumulate the posterior probabilities according to (5.18) on $C \times C$ blocks ($C = 8$). Concerning the *filtering* of the *likelihood map*, three cases were considered: no filtering at all, 5×5 weighted average filtering using a Gaussian window, and 5×5 bidimensional median filtering. In all cases, filtering is applied on

log likelihood maps to avoid numerical problems.

As to the EM algorithm, we initialized μ_1 and σ_1^2 to the mean and variance of the observed features, $\sigma_2^2 = \sigma_1^2/10$, and $\alpha = 0.5$. Convergence was assumed if the increase of the likelihood function with respect to the previous iteration was less than 10^{-3} or after 500 iterations.

The performance of the proposed algorithm has been measured by the *true positive rate* (R_{TP}), measuring the fraction of tampered blocks correctly detected as forgery, and the *false positive rate* (R_{FP}), measuring the fraction of unchanged blocks erroneously detected as forgery. If we assume N_{R1} the amount of blocks in the untampered region R_1, N_{R2} the amount of blocks in the forged region R_2, N_{mR1} the amount of blocks detected as tampered in region R_1 and N_{mR2} the amount of blocks detected as tampered in region R_2, we have:

$$R_{TP} = \frac{N_{mR2}}{N_{R2}}; \tag{5.20}$$

$$R_{FP} = \frac{N_{mR1}}{N_{R1}}. \tag{5.21}$$

The overall performance of the detector is evaluated by plotting its *receiver operating characteristic* (ROC) curve, obtained by thresholding the output maps (i.e. the *cumulated and filtered likelihood maps*) using a varying threshold value and recording the corresponding values of R_{TP} and R_{FP}. Finally, the *area under the ROC curve* (AUC) is used as a scalar parameter to describe detector capabilities: an AUC close to one indicates good detection performance, whereas an AUC close to 0.5 indicates that the detector has no better performance than choosing at random.

Results

In Fig. 5.5(a), we show the detection performance on the four ideal datasets, where for each datasets we use a predictor matched to the demosaicing algorithm, whereas in Fig. 5.5(b), we show the detection performance on the dataset using in-camera demosaicing when different predictors are applied. For each test a 128×128 tampered region has been considered. Detection results are averaged over the four different cameras. As to the resolution of the likelihood map, we have $B = C = 8$. The results show that when the predictor matches the demosaicing algorithm the performance is nearly optimal, irrespective of the used predictor, whereas in the presence of

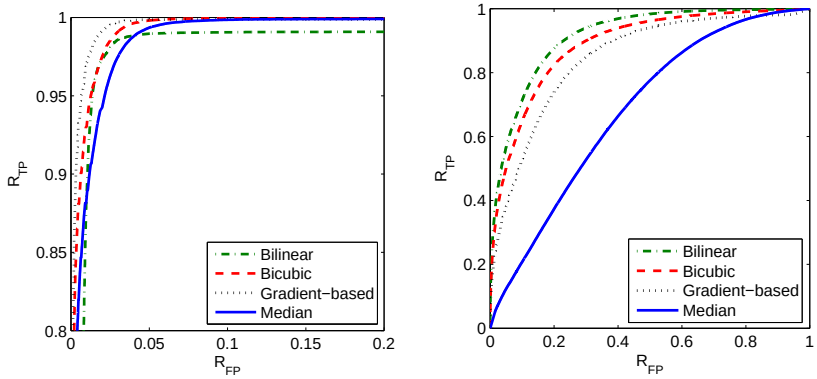


Figure 5.5: *ROC curves considering images from the* original dataset *with* 128×128 *tampered regions. Features are computed on* 8×8 *blocks: (a) ideal case: the* 400 *original images have been sampled according to the Bayer CFA pattern and then re-interpolated using the four chosen interpolation algorithms; results from all the* 400 *images are aggregated for each of the four predictors and the behaviour is shown separately; for the sake of readability, we show a zoom of the ROC curves for* $R_{TP} > 0.8$ *and* $R_{FP} < 0.2$*; AUC values are: bicubic* 0.9975*, bilinear* 0.9845*, gradient-based* 0.9975*, median* 0.9954*; (b) real case: the* 400 *original images coming from the* 4 *cameras with unknown demosaicing algorithms; results from all the* 400 *images are aggregated for each of the four predictors and the behaviour is shown separately.*

a realistic and unknown demosaicing algorithm the best average performance is obtained using the bilinear predictor. It is worth noting that in the latter case the performance of the median predictor is far worse than that of the other predictors, which is in accordance with the histograms in Fig. 5.4.

The following results show the detection performance, averaged over the four cameras, when using the bilinear predictor and different choices of algorithm parameters. In Fig. 5.6 we report the AUC values obtained using different *likelihood map* resolutions without filtering the likelihood map, under six different scenarios and considering different sizes of the tampered area. In all cases, the best performance is obtained when the exact interpolation kernel is known (in this case bilinear). Note also that the ability to localize forged regions sensibly decreases when the JPEG compression qual-

ity is below 95%. This is due to the low-pass behavior of JPEG compression, which drastically attenuates high frequency signals, such as the prediction error. With a quality factor 85%, our algorithm is unable to discriminate between the presence and the absence of CFA artifacts.

By comparing the different curves, we observe that defining our features on larger blocks makes our model more robust. These better performances are obtained at the expense of map resolution. However, in realistic conditions forgery sizes less than 8 pixels are unusual. It is also worth noting that computing the features on 2×2 or 4×4 blocks and cumulating the probabilities on 8×8 block yields slightly worse results than directly computing the features on 8×8 blocks. Lastly, the performance of the proposed detector appears similar for different forgery sizes, even though smaller tampered areas are more difficult to detect due to the reduced number of tampered blocks which decreases the reliability of the GMM estimation.

In Fig. 5.7, we compare the performance of the proposed detector using the most favorable combination of parameters, namely 8×8 resolution without cumulation, with the performance of the algorithms proposed by Dirik and Memon in [73] (DM) and by Gallagher and Chen in [72], namely the blockwise version (GC-B) and the version based on local statistics (GC-L). For a fair comparison, the DM and GC-B algorithms have been applied on 8×8 blocks, whereas the features of GC-L algorithm have been computed using 7×7 local averaging and 16-point discrete Fourier transform. The proposed feature clearly outperforms the previous approaches, demonstrating far better localization capabilities. It is also evident that the performance of all CFA-based methods degrades similarly in the presence of JPEG compression when such methods are used to localize CFA artifacts at a fine-grained resolution.

We also investigated the use of filtering on the *likelihood map*. In Figure 5.8, the AUC values are shown in the absence or presence of either mean or median filtering, using 8×8-features. The size of the tampered region is 128×128 pixels. We can see that filtering improves performances, except in the ideal case, where the effects of the loss of resolution on the edges of the tampered region is predominant, and that median filtering gives better results than mean filtering.

5.5.3 Examples

In this section, some examples of forgery localization are shown on realistically tampered images. In all the cases, the corresponding *forgery maps* have been obtained by computing features on 8×8 blocks ($C = B = 8$), using the bilinear predictor and applying median filtering on the *log likelihood map*.

In Fig. 5.9 a copy-move forgery on an image acquired with a Nikon D90 is shown. Both the original image, in Fig. 5.9(a), and the tampered copy, in Fig. 5.9(b), are saved in TIFF uncompressed format. The flower in the upper-left corner has been pasted disaligning the CFA pattern, whereas the flower in the upper right corner has been pasted maintaining the same CFA pattern. In Figs. 5.9(c)-(f) we show the forgery maps obtained with the proposed algorithm and the DM, GC-B, and GC-L algorithms, respectively. Even if the case of copy-move forgery does not perfectly fit the proposed model, since in the case of misaligned CFA artifacts the expected value of $\mathbf{L}$ is less than zero, the proposed algorithm correctly localizes the flower in the upper-left corner, whereas it is not able to localize the flower in the upper-right corner. This is not surprising, since the proposed method gives higher likelihood values for positive values of the feature and reveals local inconsistencies of the CFA artifacts even when $\mathbf{L} < 0$. As to the other algorithms, only the GC-B is able to localize the upper-left flower. Moreover, some false alarms are present in the case of saturated white regions, in which CFA artifacts are not detectable.

Very often, to make the forgery more convincing some image processing operations, like smoothing, filtering, stretching, rotating, etc., are applied. These operations, removing CFA artifacts from the tampered regions, make easier the forgery localization. In Fig. 5.10, we show an example where a tampering is done by splicing a geometrically transformed image onto an image taken by a Nikon D90 camera. In Figs. 5.10(c)-(n) we show forgery maps obtained with different algorithms, from top to bottom, the proposed algorithm, DM, GC-B, and GC-L algorithms, assuming that the tampered image was saved in JPEG format with quality, from left to right, 100%, 95%, and 90%. As can be seen, the forged region can be correctly detected in high quality images, but false alarms increase abruptly when the quality of JPEG compression decreases, because lossy compression tends to delete CFA artifacts. On this example, DM algorithm appears less effective than the other algorithms.

The inspection of the forgery maps in Figs. 5.9-5.10 suggests that the proposed method is less effective in the presence of either almost flat areas or sharp edges. In the first case, the prediction error is almost zero irrespective of the presence of CFA artifacts, so that this appears as an intrinsic limit of the method. In the second case, this can be ascribed to the signal adaptive and possibly non-linear behavior of realistic in-camera demosaicing algorithms. At least in theory, such effects could be eliminated by using some prior knowledge regarding in-camera CFA interpolation, which should yield results very close to the ideal behavior shown in Fig. 5.5. An alternative approach could be that of reverse engineering the CFA interpolation algorithm, for example using methods such as in [65] to take into account a signal adaptive behavior. However, in the presence of heavily manipulated images this approach is likely to produce a biased estimate and must be handled with care.

5.6 Conclusions

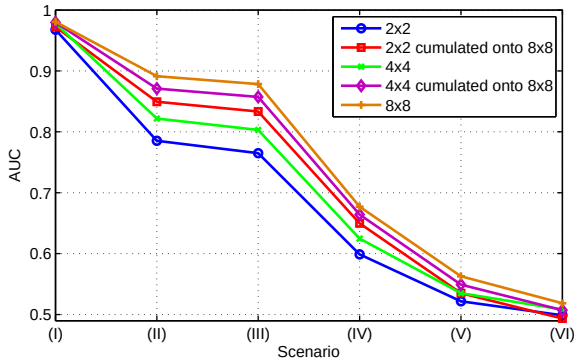
In this Chapter, a forensic algorithm to localize forged regions in a digital image without any a-priori knowledge about the location of the possibly tampered areas has been presented. Considering the CFA demosaicing artifacts as a digital fingerprint, we proposed a new feature measuring the presence of demosaicing artifacts even at the smallest 2×2 block level; by interpreting the local absence of CFA artifacts as an evidence of tampering, the proposed scheme provides as output a forgery map indicating the probability of each block to be trustworthy.

The validity of the proposed system has been demonstrated by computing the ROC curve of a forgery detector based on thresholding the probability map, considering different scenarios and different algorithm parameters, and comparing the performance with the approaches in [73] and [72]. The results show that by a proper parameter configuration CFA artifacts are usually reliably localized even at 8×8 block resolution. Results are also confirmed by tests carried out on realistic forgeries.

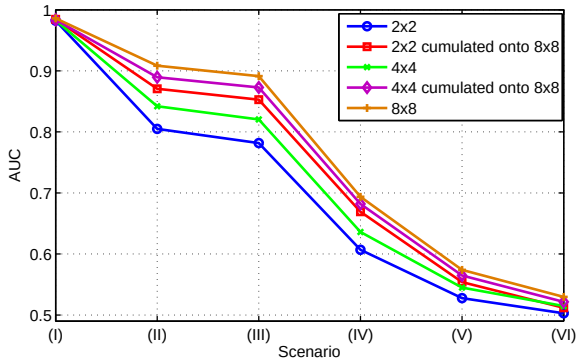
The fine-grained localization of tampered regions using CFA artifacts is the main contribution of this part, since in previous approaches either the area to be investigated has to be manually selected, or automatic block processing obtains poor detection performance when forced to reveal CFA artifacts at a fine-grained scale. The results show that the proposed algorithm

can be a valid tool for detecting and localizing forgeries in images acquired by a digital camera. However, it should be remarked that the detection performance is strongly affected by JPEG compression, limiting the applicability to scenarios in which the image under test is either uncompressed or compressed with high quality factors. Moreover, the present method may not be directly applicable to cameras using a super CCD [81].

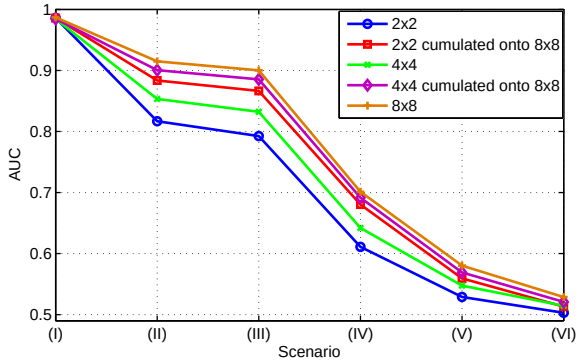
Test on realistically tampered images demonstrate that, due to the presence of uniform or very sharp regions, automatic detection may give a remarkable false positive rate. Therefore, in order to limit the incidence of false positives human interpretation of the *forgery maps* is still required. Future work will be then devoted to the study of segmentation algorithms that, by taking into account the local content characteristics, allow to produce a final map with reduced false positives.



(a)



(b)



(c)

Figure 5.6: *Effects of the* Likelihood Map *resolution on the AUC values. We consider TIFF images with bilinear interpolation (I) and TIFF images with in-camera demosaicing (II). These latter images are then compressed in JPEG format with quality at 100% (III), 95% (IV), 90% (V) and 85% (VI). Different forgery sizes are investigated: (a)* 32×32 *pixels; (b)* 64×64 *pixels; (c)* 128×128 *pixels.*

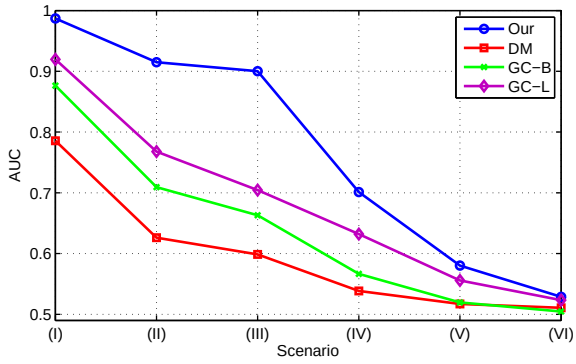


Figure 5.7: *Comparison between the proposed algorithm and the algorithms by Dirik and Memon (DM) [73] and by Gallagher and Chen (GC-B and GC-L) [72]. We consider TIFF images with bilinear interpolation (I) and TIFF images with in-camera demosaicing (II). These latter images are then compressed in JPEG format with quality at 100% (III), 95% (IV), 90% (V) and 85% (VI). The features are computed on* 8×8 *blocks. Tampered region is* 128×128 *pixels.*

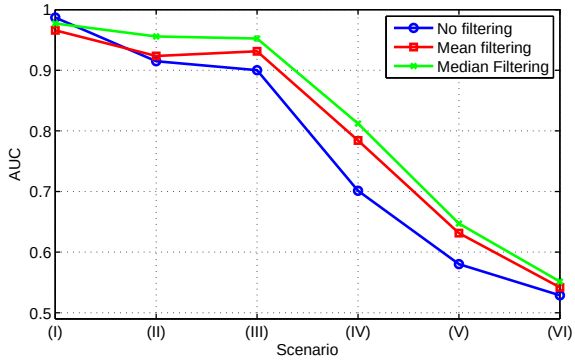


Figure 5.8: *Effects of* Likelihood Map *filtering on the AUC values. We consider TIFF images with bilinear interpolation (I) and TIFF images with in-camera demosaicing (II). These latter images are then compressed in JPEG format with quality at 100% (III), 95% (IV), 90% (V) and 85% (VI). The features are computed on* 8×8 *blocks. Tampered region is* 128×128 *pixels.*

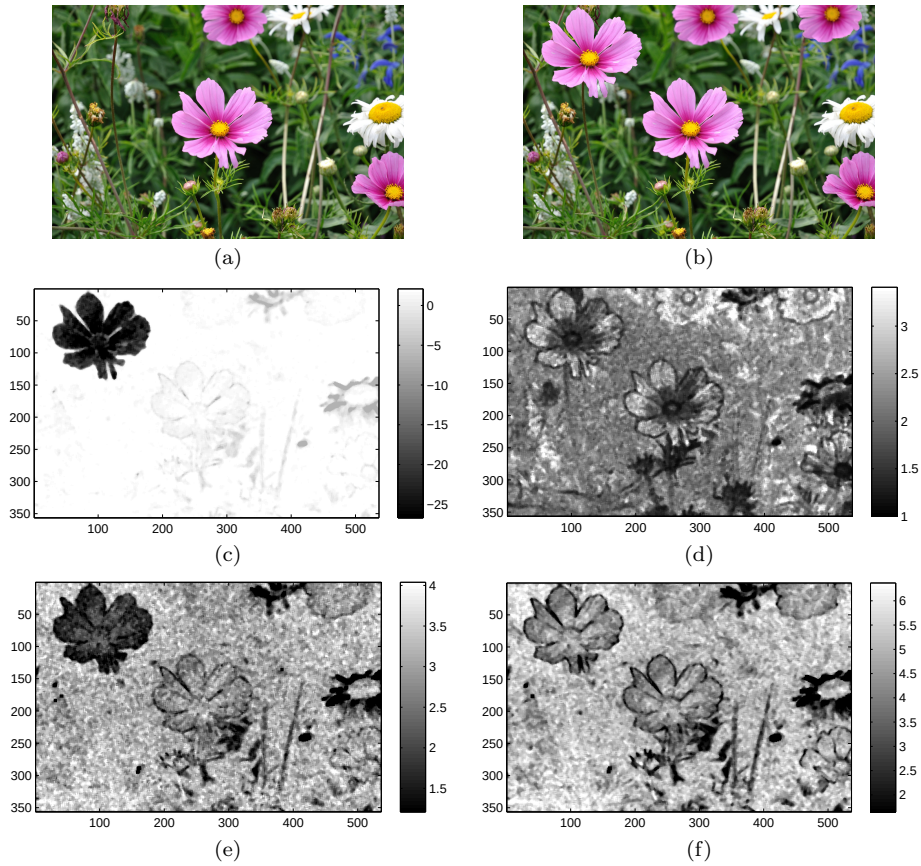


Figure 5.9: *Example of a copy-move forgery in an image with CFA artifacts. The resulting image is saved in TIFF format: (a) original image acquired by the Nikon D90 camera; (b) tampered image; forgery maps obtained with the proposed (c), DM (d), GC-B (e), and GC-L (f) algorithms. Bright areas indicate high probability of presence of CFA artifacts (unchanged blocks), whereas dark areas indicate low probability of presence of CFA artifacts (tampered blocks).*

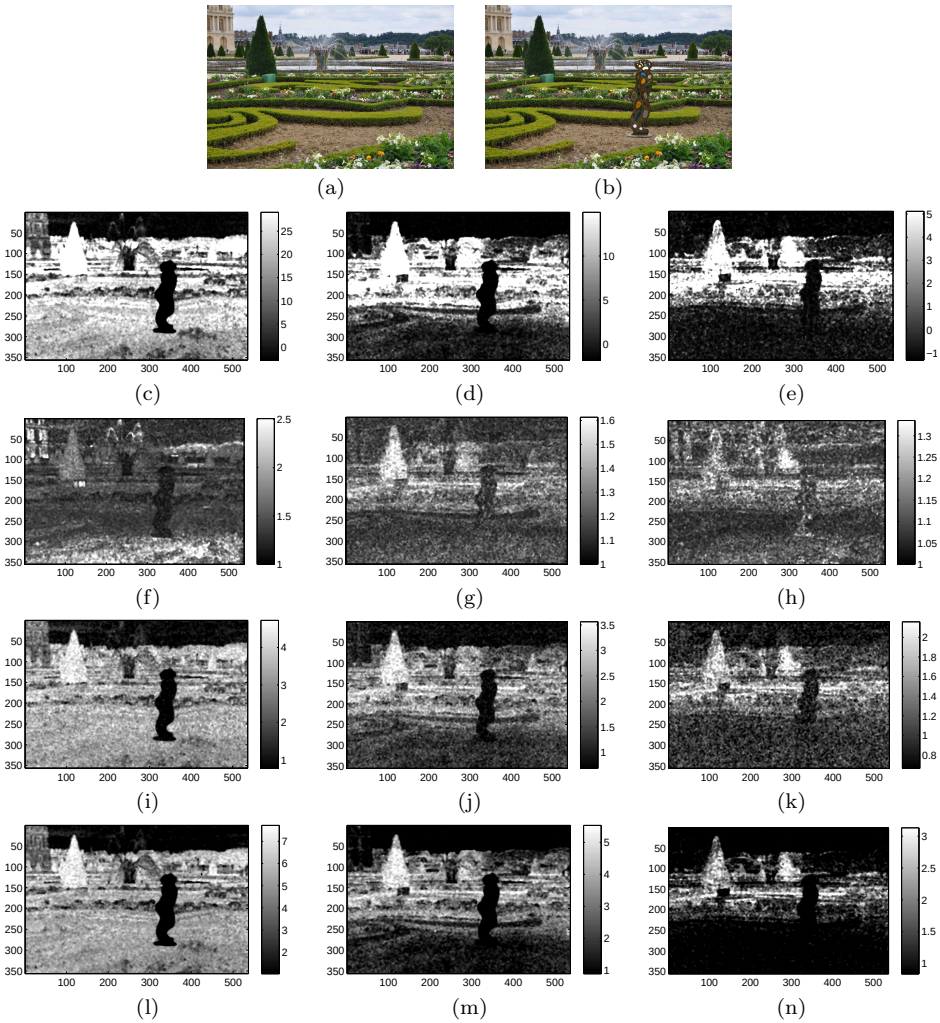


Figure 5.10: *Example of a forgery in which a processed content (statue) is pasted on an image with CFA artifacts: (a) original image; (b) tampered image; (c)-(n)* forgery maps *obtained after saving in JPEG format with quality, from left to right,* 100%, 95% *and* 90%*: (c)-(e) proposed algorithm; (f)-(h) DM algorithm; (i)-(k) GC-B algorithm; (l)-(n) GC-L algorithm.*

Chapter 6

Blind mutations detection by using a multi-clue analysis

Image authenticity verification has usually to be carried out without any knowledge about the processing undergone by the image or the region that suffered some forgery. In this setting, it is fundamental to rely on a multi-clue analysis, that cleverly merges the information stemming from several complementary tools. In this Chapter we introduce a fully automatic framework for fusing the maps output by a set of unsupervised forgery localization algorithms. The framework takes into account the forgery maps, their reliability and the compatibility among the different traces considered by the different tools. The achieved localization map is then refined by exploiting image content, thus improving the overall performance of the proposed system with respect to state of the art approaches.

6.1 Introduction

An important limit of the approach introduced in Chapter 5 is that it is based on the observation of a single forensic trace. In practical scenarios, the simultaneous analysis of different footprints could improve tampering detection and localization. As to traces detected on the whole image, a number of techniques have been proposed to fuse the information at the feature level, i.e., by devising a complex classifier that accounts for multiple footprints [82–85]. Other approaches work at the score level, meaning that the scalar output of the tools is considered during fusion [86,87]. The overall

performance of the above methods can be futher improved by taking into account background information during fusion [88].

As to forgery localization, simple pixel-level fusion of different forensic tool outputs has been investigated in [89]. The main limitation of this work is that no information about tools reliability and compatibility has been used. This is the step forward we take with our framework, in which we propose a multi-clue approach for the unsupervised localization of forgeries in digital images. The proposed method is based on Dempster-Shafer Theory of Evidence (DST) [90]: under this flexible framework, we are able not only to fuse information coming from different unsupervised forensic tools, but also to exploit several kinds of background information to increase the reliability of the results. More precisely, our approach is able to exploit: i) *tool-based* information, since the fusion algorithm knows the reliability of each tool under different working conditions and exploits information about local and global properties of the analyzed content to better interpret the output of tools. This fact is usually beneficial for forgery detection [88], and is likely to be even more important for forgery localization, where the output is a fine-resolution probability map; ii) *trace-based* information, meaning that the fusion algorithm knows the compatibility relationships between traces and can manage the case where two incompatible traces are simultaneously present; iii) *semantic-based* information, which means exploiting the content of the analyzed image to improve the forgery localization map.

6.2 Elements of Dempster-Shafer Theory of Evidence

Dempster-Shafer Theory [90,91] is a mathematical tool providing a way to model uncertainty and to combine information coming from multiple sources. Let us denote with $\Theta = \{\theta_1, \theta_2, \ldots, \theta_n\}$ the exhaustive set of mutually exclusive possible conclusions to be drawn. The frame of discernment of Θ is its power set 2^{Θ}, that is the set of all possible subsets of Θ (whose cardinality is $2^{|\Theta|}$). A Basic Belief Assignment is a function assigning a *mass* to elements of the frame of discernment associated to Θ.

Definition Let Θ be a frame. A function $m^{\Theta} : 2^{\Theta} \rightarrow [0,1]$ is called a

Basic Belief Assignment (BBA) over the frame Θ if:

$$m^{\Theta}(\emptyset) = 0; \qquad \sum_{A \in 2^{\Theta}} m^{\Theta}(A) = 1 \tag{6.1}$$

where the summation is taken over all possible subsets A of Θ.

Intuitively, the mass assigned to a set is the amount of certainty supporting exactly that set, and not any of its subsets; for example it may be that $m^{\Theta}(\{\theta_1 \cup \theta_2\}) < m^{\Theta}(\{\theta_1\})$. The function accumulating the certainty about a set and all its subsets is called *belief* function:

Definition Given a BBA m^{Θ} over Θ, the Belief function $Bel : 2^{\Theta} \rightarrow [0, 1]$ is defined as follows:

$$Bel^{\Theta}(A) = \sum_{B \subseteq A} m^{\Theta}(B). \tag{6.2}$$

$Bel^{\Theta}(A)$ summarizes all our reasons to believe in A based on the available knowledge. Going back to the previous example, we surely have: $Bel^{\Theta}(\{\theta_1 \cup \theta_2\}) \geq Bel^{\Theta}(\{\theta_1\})$. The reader can find more details and properties in [90].

DST is widely known as a tool for combining the evidence coming from multiple independent sources of information. Indeed, given two BBAs m_1^{Θ} and m_2^{Θ}, we can obtain a fused BBA by using Dempster's Combination Rule:

Definition Let Bel_1 and Bel_2 be belief functions over the same frame Θ with BBAs m_1 and m_2. For all non-empty $X \subseteq \Theta$ the function m_{12} defined as:

$$m_{12}(X) = \frac{1}{1-K} \cdot \sum_{\substack{A,B \subseteq \Theta: \\ A \cap B = X}} m_1(A) m_2(B) \tag{6.3}$$

where $K = \sum_{A,B:A \cap B = \emptyset} m_1(A) m_2(B)$, is a BBA function defined over Θ and is called the *orthogonal sum* of Bel_1 and Bel_2, denoted by $Bel_1 \oplus Bel_2$.

The concept of "independence" in DST is not rigorously defined, it generically means that information must be provided by unrelated sources. When new evidence defined on a different domain becomes available, it is necessary to redefine available and new BBAs over a common frame of discernment before applying of the combination rule, through belief extension, as defined in [90].

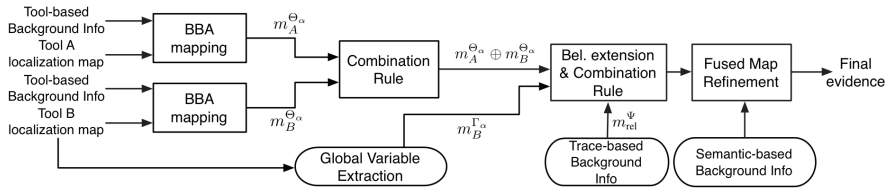


Figure 6.1: *Block scheme of the proposed framework for forgery localization, where two tools A and B searching for a forensic trace α are considered. For the sake of clarity, global variables for Tool A are omitted in the drawing.*

6.3 DST-Based Multi-Clue Analysis for Forgery Localization

The framework we propose aims at exploiting the output of an arbitrary set of unsupervised tamper localization algorithms and several kinds of background information so as to produce a single comprehensive and more reliable map.

Our system is reminiscent of the data fusion scheme described in [86]. In this scheme, the user manually selects a sufficiently large region and runs a set of tools assigning to the region a scalar value measuring the presence of a certain forensic trace in it. Then, the goal is to merge these outputs, by also taking into account some local properties of the region that may influence the reliability of the forensic tools. The way this is performed is briefly sketched below:

1. output from each tool is interpreted and written as a BBA about presence/absence of a trace in the selected region;

2. BBAs obtained from different tools are combined using Dempster's combination rule [90], after applying belief extension for combining the information about different traces;

3. compatibility relationships between different traces (modeled through a BBA) are introduced using Dempster's rule;

4. final decision: the total belief that the region has been forged is computed based on the merged information.

The most intuitive approach to extend the above analysis to forgery localization would be to simply apply the whole procedure separately to each single element of the map (also called "analysis block", from now on). However, this choice is potentially misleading because of the nature of forgery localization tools. Indeed, as stated in Section 6.1, the accuracy of forgery localization tools is strongly affected by the local properties of the image: for example, very smooth or saturated regions are critical for many tools (see, for example, [60, 62]), so that values assumed by the map in those regions are less reliable. As a consequence, attention must be paid in properly interpreting the output of the tool locally. To this aim, for a forensic trace α, we define the set $\Theta_\alpha = \{t\alpha, n\alpha\}$, where $t\alpha$ is the proposition "trace α is present in the analysis block" and $n\alpha$ is the proposition "trace α is not present in the analysis block". We model this *local* information provided by the tool τ with the following BBA over the frame Θ_α:

$$m_\tau^{\Theta_\alpha}(X) = \begin{cases} \mathrm{L}_\tau(i) & \text{for } X = \quad \{(t\alpha)\} \\ \mathrm{N}_\tau(i) & \text{for } X = \quad \{(n\alpha)\} \\ \mathrm{D}_\tau(i) & \text{for } X = \{(t\alpha) \cup (n\alpha)\} \end{cases} . \tag{6.4}$$

In the above equation $\mathrm{L}_\tau(i)$, $\mathrm{N}_\tau(i)$ and $\mathrm{D}_\tau(i)$ are scalar values obtained by interpreting the output of the tool in the i-th analysis block. It is here that *tool-based* background information enters the picture: besides considering the value of the localization map in the position of block i, a set of local properties of the image is evaluated (e.g., mean value or variance of pixels in the analysis block i) and used to determine the mentioned values for equation (6.4). To perform this mapping from tool outputs and background information to BBAs, we rely on the method recently proposed in [88]: such method exploits a set of training images to learn how local properties affect the output of the tool. Thus, given image and forgery localization map, using this approach we obtain values for (6.4) for each block of pixels. This stage of the framework is represented in the left-most side of Figure 6.1 ("BBA mapping" blocks).

6.3.1 Global variables

There is another fundamental difference between forgery detection and forgery localization tools. Independently from the analysis domain (e.g., pixel or DCT domain), unsupervised forgery localization tools typically as-

sume that the signal under analysis is the mixture of two components: one component deriving from parts of the image that were manipulated, and one deriving from unaltered parts [34, 60, 92]. A statistical model is defined for each component, and the parameters of the models are estimated from available data. Finally, each (block of) pixels is assigned a probability of belonging to each model, thus producing a forgery localization map, like the one in the center of Figure 6.2. However, when for some reason the two components are not correctly separated, the produced localization map is practically useless, although it assigns a sensible value to each region (right hand of Figure 6.2). A simple yet effective way to understand whether the tool managed or not to separate the two components is to analyze the produced localization map as a whole: when the components are not separated, the whole map takes values in a narrow range, meaning that all pixels belong to the same component, while the opposite happens when two components are separated (compare the two maps of Figure 6.2 for an explicative example).

The above discussion suggests that we cannot simply interpret elements of the localization map as "stand alone small blocks", we must also model the global information that is obtained from the localization map as a whole. In order to do that, we propose to introduce for each considered forensic trace also a *global* variable. Taking again the general forensic trace α as reference, we define the frame $\Gamma_\alpha = \{T\alpha, N\alpha\}$ where $T\alpha$ is the proposition "the two components related to α were separated" while $N\alpha$ has the opposite meaning. After running a localization tool searching for α, a BBA over Γ_α must be defined. We are not forced to give a binary interpretation: indeed the border between the two cases is not always sharp. Hence, for a generic tool τ, we propose to model this information through the following BBA:

$$m_\tau^{\Gamma_\alpha}(X) = \begin{cases} (1 - \mathrm{W}_\tau)\mathrm{G}_\tau & \text{for } X = \{(T\alpha)\} \\ (1 - \mathrm{W}_\tau)(1 - \mathrm{G}_\tau) & \text{for } X = \{(N\alpha)\} \\ \mathrm{W}_\tau & \text{for } X = \{(T\alpha) \cup (N\alpha)\} \end{cases} . \tag{6.5}$$

If the tool τ is based on model separation, then $\mathrm{G}_\tau \in [0, 1]$ quantifies the confidence about the two components of the mixture being successfully separated, and $\mathrm{W}_\tau = 0$. Instead, if τ is not based on model separation, we assign all the mass to the doubt by setting $\mathrm{W}_\tau = 1$, thus yielding the neutral element for Dempster's combination rule [91] and disabling the contribution of this BBA.

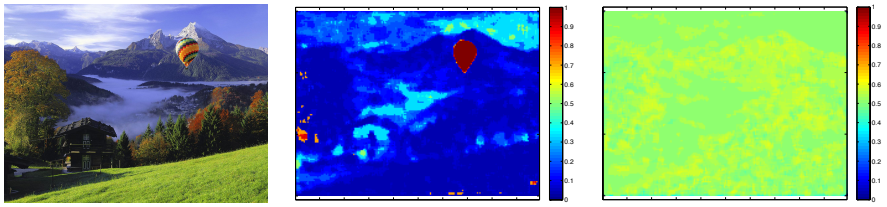


Figure 6.2: *A forged image (the baloon is pasted) and the forgery localization map obtained with the tool in [60] on the tampered file (center plot) and on a re-compressed version of the tampered file (right-most plot). As we can see, in the latter case the map is not discriminative as it takes values near to 0.5 everywhere; on the contrary, the same value in the center plot clearly characterizes not-tampered regions.*

This phase of the framework is drawn in the lower part of Figure 6.1.

Notice that, for the moment, the above BBA is not linked in any way to that in eq. (6.4) (they are also defined on different frames, Γ_α and Θ_α respectively). This means that we are not still logically linking local and global information about the presence of the trace.

6.3.2 Inclusion of trace-based background information

Decision fusion is particularly interesting when tools searching for different traces are merged together. In fact, by knowing the theoretical properties of each forensic trace, in many cases the analyst can explicitly tell whether a combination of traces is plausible or not: this is what we call *trace-based* background information. As it was shown in [86], DST allows to write rather easily such information in terms of BBAs, allowing to combine it with the information provided by single forensic tools.

Also in this case, as we turn to forgery localization some noticeable differences appear. In the framework proposed in [86] each forensic trace is modelled with one variable, so that only relationships between different traces are to be considered. In the scenario addressed in this work, instead, each trace is better represented with two variables (one referring to the local presence of the trace and one to the suitability of the global model). Hence, we also have a relationship between these two variables establishing the link between local and global information about the trace, and allowing to change the interpretation of the local output of the tool based on the global information.

Table 6.1: *Example of traces relationships.*

$\boldsymbol{\Theta}_\alpha$	$\boldsymbol{\Gamma}_\alpha$	$\boldsymbol{\Theta}_\beta$	$\boldsymbol{\Gamma}_\beta$	**Plausible**	**Interpr.**
$t\alpha$	$T\alpha$	$n\beta$	$T\beta$	Y	Tamp.
$n\alpha$	$T\alpha$	$n\beta$	$T\beta$	Y	Auth.
$t\alpha$	$T\alpha$	$t\beta$	$T\beta$	N	-

It is worth noting that the global information about the presence of one trace can also affect the interpretation of different forensic traces. Therefore, we propose to write together these compatibility relationships. A good way to do that in practice is to write a table listing on rows the combinations of variables: each row is then labelled by the analyst as either plausible or not plausible. For plausible rows, the analyst also specifies the interpretation associated to that row in terms of authenticity of the block. Of course, this has to be done only once for a set of forensic traces. An example for two traces α and β is given in Table 6.1: the first row states that, for any analysis block of an image where the global models of both trace α and β were successfully separated, it is plausible to find only the trace α and not the other; moreover, the interpretation associated to this combination is "the block is tampered". The second row of the table tells that local absence of both traces is plausible and is to be interpreted as the block being authentic (based on the available information). The last row, instead, states that the two traces cannot be present simultaneously in the same block. The table is truncated for the sake of brevity; the complete version has 16 rows, even though it makes sense to write explicitly only plausible combinations.

Compatibility tables can be easily written in terms of a BBA as follows: for a given set $\mathcal{T}$ of considered traces, let us define as $\Psi = \prod_{j\in\mathcal{T}} \Theta_j \times \Gamma_j$ the common frame of discernment, where $\prod$ and $\times$ denote the Cartesian product. Let us also denote by $\Psi_{\mathrm{PL}} \subseteq \Psi$ the union set of all combinations that are considered plausible. Then, the following BBA declares that combinations that are not plausible have to be considered as conflicting information:

$$m^{\Psi}_{\mathrm{rel}}(X) = \begin{cases} 1 & \text{for } X \in \Psi_{\mathrm{PL}} \\ 0 & \text{for } X \notin \Psi_{\mathrm{PL}} \end{cases}; \tag{6.6}$$

this phase is denoted in Figure 6.1 by the block whose output is m^{Ψ}_{rel}.

6.3.3 Obtaining the fused localization map

By applying Dempster's combination rule to the BBA resulting from traces relationship and those available from single tools, we obtain a single BBA summarizing the available information. Then, it makes sense to compute the belief of the set composed by all plausible combinations whose interpretation is "tampered", using equation (6.2). Notice that this computation is to be done only once for a given set of forensic traces; the resulting formula remains the same for every image, so it can be stored and evaluated when needed. By evaluating the formula for each analysis block of an image, a map taking values in [0,1] is produced, which tells the total belief for each block of being tampered.

6.3.4 Map refinement by guided filtering

As the vast majority of forgery localization tools process each analysis block independently of the others [34, 60, 62], the resulting localization map are typically affected by noise. In some cases, authors proposed to filter the map to reduce noise (e.g., in [60] median filtering is advised), but this solution could be not sufficient when several maps have to be fused. Moreover, the use of filtering based on fixed window (i.e. as median or mean ones) rises the problem of how to set the window size: a large window produces more reliable results, but reduces the effective resolution of the localization map; conversely, a small window has a better capability to localize forgery (especially in the case of small tampering), but with limited noise reduction capability. To this aim, we propose to exploit what we call *semantic-based* background information, meaning that we let the content of the analysed image to drive the map processing. Recently, authors of [93] proposed to use guided filtering [94] to accomplish this task. Guided filter computes the filtered output by considering the content of the guidance image. In this application, the input is the localization map and the guidance image is the image under inspection. The main advantage is that the guided filter transfers the structures of the guidance image (i.e. tampered image) to the filtered output (i.e. filtered map). Moreover, as shown in [94], this filter can be efficiently computed in $O(N)$ time, and this makes it more efficient than other *edge-preserving* filters, as bilateral filter, whose extended version can be found in [95].

6.4 Experimental results

In this section we discuss the experiments that we carried out to prove the validity of the proposed approach.

6.4.1 Case Study

The tools we employ are based on *aligned* double JPEG compression (AJPEG) footprints [60], *non-aligned* double JPEG (NAJPEG) footprints [92] and Color Filter Array (CFA) inconsistencies [62]. We summarize briefly their underlying scenarios.

In [60], it is analyzed a scenario in which an original JPEG image, after some localized forgery, is saved again in JPEG format. Such a forgery disrupts JPEG compression footprints. Examples of this kind of manipulation are a cut and paste from either an uncompressed image or a resized image, or the insertion of computer generated content. In this case, DCT coefficients of unmodified areas undergo a double JPEG compression thus exhibiting double quantization (DQ) artifacts, while, very likely, DCT coefficients of forged areas do not show such artifacts. If the image was not cropped between the first and the second compression, the grid of the DCT coefficients of the first compression is *aligned* to the second one.

In [92] a different scenario is proposed for image splicing. Here, it is assumed that a region from a JPEG image is pasted onto a host image that does not exhibit the same JPEG compression statistics, and that the resulting image is re-compressed in JPEG format. In this case, the forged region exhibits double compression artifacts, whereas the not manipulated region does not. By assuming a random placement of the spliced region, there is a probability of 63/64 that the grid of the DCT coefficients of the first compression is *not aligned* to the second one (NAJPEG artifacts).

In [62], authors propose a forgery localization method based on the traces left by CFA interpolation. The scenario is a one in which a local forgery destroys the correlation introduced by in-camera *demosaicing*. Thus, the forged region does not show CFA artifacts, whereas the remaining part of the image presents them.

6.4.2 Methodology

To simplify our case study, we set the dimension of each block to 8×8 pixels, which represents the minimum resolution on which double JPEG compression based algorithms work. In order to define the mapping from the localization maps to BBAs (Eq. (6.4)), we adopt the method proposed in [88], choosing the following set of properties to locally characterize the reliability of each tool τ:

1. q_2: the value of the last compression factor, if any;
2. μ: mean value intensity of the block of pixels;
3. σ: standard deviation of the intensity of the block of pixels;
4. q_1: the value of the first compression factor, if any.

It is worth noting that q_1 is not directly observable, but it is estimated by AJPEG and NAJPEG tools, and it is employed only for CFA, since as shown in [62], traces of CFA artifacts could be removed by strong past compression. The generic analysis block is thus described by the vector $v = (o_\tau, q_2, \mu, \sigma, q_1)$, where o_τ denotes the value of the block in the map produced by tool τ (in our case, $\tau \in \mathcal{T} = \{\text{AJPEG}; \text{NAJPEG}; \text{CFA}\}$). By applying the approach proposed in [88], each vector is associated to scalar values L_τ, N_τ and D_τ (see Eq. 6.4); as to the parameters required in [88], we used $\alpha = 0.85$ and $\hat{\eta} = 12$ for each tool, whereas $\hat{\gamma} = 0.5$ for CFA tool, $\hat{\gamma} = 512$ for AJPEG tool and $\hat{\gamma} = 2048$ for NAJPEG tool. These values were gathered through 5-fold cross validation and grid search.

Finally, as motivated in section 6.3.2, we define an empirical method to assign values to global variables, telling to what extent the tool successfully separated the two components for its own trace. Since all the considered tools are based on model separation, according to equation (6.5) we set $W_\tau = 0 \; \forall \tau \in \mathcal{T}$, and we define a linear piecewise function:

$$G_\tau(\rho) = \begin{cases} 1 & \text{for } \rho \geq a \\ \rho / a & \text{for } \rho < a \end{cases}, \tag{6.7}$$

where the input ρ is the percentage of blocks belonging to the less populated model, as explained in Section 6.3.1. By definition, G_τ takes values in $[0, 1]$ and it also depends on the parameter a, which represents the minimum percentage of blocks allowing a model to be detected. The value of a was

derived from experimental evidence, set to $a = 1/8$. The rationale is that two components can be separated if at least $1/8$ of the blocks shows the footprints searched for.

6.4.3 Results

Here we show the improvements in localizing forgeries in an unsupervised scenario. To quantify it, we generate three different sets of images to train and test the proposed framework. Firstly, we define a *training* set to train the BBA mapping module, incorporating *tool-based* background information. The second step is to design a proper dataset (we refer as *testing*) to compare the performance of each tool employed individually with respect to those of the framework. It is worth noting that we assume a *blind* case, i.e. each tool is applied without any a priori information about the type of tampering applied to the image. Finally, we build a dataset of realistic spliced images in order to show the real capabilities of localizing a forged region. The details are listed below.

Training: Starting from 100 uncompressed TIFF images cropped to a 1024×1024 resolution, three different tampering (AJPEG, NAJPEG and CFA destruction) have been applied separately, in such a way that the traces detected by each algorithm have been inserted (or deleted) from the left half of each image. For the AJPEG and NAJPEG traces, the quality factors of the first and second compression are in $\{50, 60, 70, 80, 90, 100\}$, whereas for the CFA footprint, the quality factors employed are in $\{50, 60, 70, 80, 90, 100, \text{Inf}\}$, where Inf represents the case of TIFF uncompressed images. By combining all possible compression factors, we obtain a set composed by 3600 images for AJPEG, 3600 for NAJPEG and 700 for CFA case.

Testing: Starting from 50 uncompressed TIFF images, with a different content from the training set, we apply the same tampering as before to the central block of 512×512 of the images. For AJPEG and NAJPEG traces, the quality factors of the first compression are in $\{60, 70\}$, whereas the quality factors of the second are in $\{80, 90\}$. For the CFA based tampering, a median filtering is applied to remove traces of CFA artifacts. Overall, 750 test images have been created: 200 with AJPEG tampering, 200 with a NAJPEG tampering, 150 with CFA tampering and 200 containing AJPEG and NAJPEG traces at the same time.

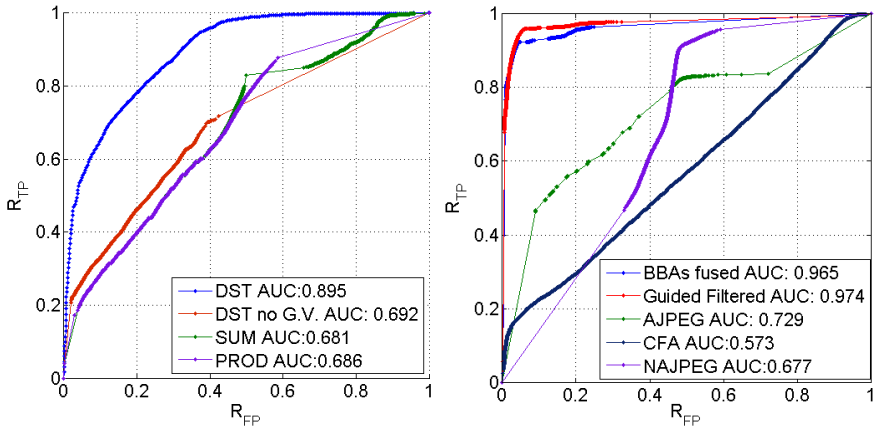
Realistic: 19 realistic forgeries have been created through a *cut and past* strategy, by inserting a content (i.e. an object) coming from an image

onto another one, and keeping track of the forged region position. The set is composed of 4 TIFF images, whereby an object (without CFA artifacts) is pasted onto another (with CFA artifacts), 6 images with AJPEG footprints , 5 images with NAJPEG footprints and 4 images whereby objects with NAJPEG traces have been inserted in images with AJPEG traces. All forgeries were made in such a way that each footprint is easily detected, since the aim of this dataset is to evaluate the capability of localizing a realistic forgery.

To prove the validity of the framework, we use the *true positive rate* (R_{TP}), measuring the fraction of tampered blocks correctly detected as forgery, and the *false positive rate* (R_{FP}), measuring the fraction of unchanged blocks erroneously detected as forgery. The overall performance of the compared methods are evaluated by plotting its *receiver operating characteristic* (ROC) curve, obtained by thresholding the output maps with a varying threshold value and recording the corresponding values of R_{TP} and R_{FP}. The *area under the curve* (AUC) is finally employed to summarize the discrimination capability of detectors.

The first test is carried out on the *testing* dataset, with the aim to compare our framework to each tool, applied independently and in a blind way. The performance, evaluated in terms of AUC, show that the DST-based framework (AUC = 0.895) outperforms the single detectors AJPEG (AUC = 0.854), NAJPEG (AUC = 0.607) and CFA (AUC = 0.588). It is worth noting that no post-filtering has been applied to the output of fusion step.

As second step, we make a comparison between the performance of our framework and those of the methods proposed in [89], based on the sum and the product of the output map provided by each tool. Moreover, the performance of the framework are evaluated by employing or not global variables, as defined in Section 6.3.1. The results are shown in Fig. 6.3 (a) by means of ROC curves, evaluated on the *testing* dataset. As we can see, the proposed framework has the best capability of localizing forgeries, and the introduction of global variables dramatically impacts the performance. This is explained by the fact that the introduction of global variables provides further information about the reliability of the value given by a tool. Finally, we present the localization capability of the framework in the case of realistic tampering. In Figure 6.3 (b), we show the performance of the method without post-filtering and in case of guided filtering at the end of the fusion framework. Moreover, a comparison with each tool performance is proposed. As expected, the refinement by using guided filtering increases the



(a) (b)

Figure 6.3: *In Fig. (a), we show a comparison of our framework (blue curve) with the methods proposed in [89], based on the sum (green) and the product (purple) of the output map. Moreover, we show the decrease in the case of absence of global variables (red). The performance are evaluated on the* testing *dataset. In Fig. (b), we show a comparison of the localization capability without post-filtering (blu curve), with the use of guided filtering (red) and the application of each single tool AJPEG (green), NAJPEG (purple) and CFA (black), applied to the* realistic *dataset.*

accuracy in localizing realistic forgeries. Even in this case, the DST-based framework has better capabilities with respect to each single tool, applied independently and in a blind way.

6.5 Conclusions and Future Work

In this Chapter a framework for unsupervised multi-clue forgery localization has been proposed, which merges information provided by a set of forensic tools with background information freely available to the analyst. Such a framework exploits the peculiar properties of those localization tools that are based on mixture models, by introducing global variables that are taken into account by the system. Although the way we assigned values to such variables is still rather empirical, their impact on the overall performance is dramatic. The formalization of global variable assignments and the extension to the case of copy-move detectors, whose output map can not distinguish between original and pasted regions, is left for future work.

Chapter 7

Conclusions

In this thesis we walked through different approaches, borrowed from Image Forensics, to attempt to reconstruct the evolution of digital images. We started from the case in which an image changes over time, keeping its own semantic content. We defined a new dissimilarity measure to reliably reconstruct the phylogeny of an image. Moreover, we tackled the same problem when other images with the same content are not available, by solving a well defined case study. The step forward was to extend our study to the case in which images change partially its own content. Our efforts were in extending the Image Phylogeny approach to the case of multiple parenting. Moreover, we dealt with the scenario in which the hypothesis on which image Phylogeny works are not satisfied: we developed a new algorithm able to provide what regions of the image suffered some processing, by using statistical correlations introduced by Colour Filter Array, and finally we integrated such a tool in a general multi-clue based framework.

Besides summarizing our contributions, this final chapter outlines some important open issues that, we believe, should be pursued in the near future, and provides a few remarks on the reconstruction of image evolution.

7.1 Summary

Image Forensics was proved to be suitable to study how images evolve over time, and it has received a lot of attention in recent past years in the academic and industrial community. Today we have tens of different tools, together with many elegant mathematical formulations of topics like multi-

ple quantization or resampling. However there is a concern about practical applicability of image forensic tools, so that only few of them are ready to be used in real world cases today, especially in the blind reconstruction of image evolutionary history.

7.2 Open issue

Before drawing the final remarks, we would like to focus the attention on two topics that have received few consideration up to today, namely contextual analysis and sentiment analysis. Contextual analysis refers to the task of detecting whether an image is used out of the correct context, so to mislead the user. It is easy to understand that deliberately placing a picture in the wrong position can totally subvert its meaning, or the meaning of surrounding content, even without changing one pixel. We may say that altering the semantic meaning of a picture can be done either by manipulating the picture or by manipulating the context wherein the picture is placed. Also this represents a sort of evolution of the image. Of course, this kind of investigation sets big challenges, also due to the difficulty of interpreting the semantic meaning of multimedia objects and environments. We may consider the existent studies on image phylogeny as a first step in this research direction: given a set of near-duplicate images, phylogeny methods aim at recovering the dependency graph telling which picture originated which. A step forward could be the analysis of metadata associated to image storage formats as JPEG. Information as camera model, camera parameters, timestamp or GPS could be employed in a forensics analysis, especially when we want to go back to the sources that have generated a composite image. Sentiment analysis of images has the aim of going back to the motivation (or causes) of the evolution of images, and how such a processing modifies the feeling of beholder. Such an information could provide, if available, a complete vision of how and why an image is evolved.

7.3 Final remarks

We spend some words on the main limitations shared by image forensic methods for image evolution study. As long as the literature is concerned, the first enemy of image forensic techniques is counter-forensics, as it aims at erasing the (already fragile) traces left during image processing. In practice,

however, the real enemy is the way digital contents are commonly archived and shared. When an image is uploaded on, say, Flickr or Facebook, it is resized and recompressed by default. Unfortunately, such operations are a very effective, though involuntary, counter-forensic mean. In general, we can say that the main problem for the Image Forensics is that it has to work on contents whose integrity is seldom preserved: such limitation is extremely felt in the case of blind reconstruction scenarios. The only way for forensic techniques to face with this problem is to devise more robust methods, searching for traces that survive these kind of processing. One noticeable example is given by geometrical and physical features (shadows, lighting conditions, perspective consistency), however such techniques require the manual aid of a clever and patient analyst, and they are barely applicable on large amount of data. Another effective counter-measure is the synergic use of many different tools, hoping that at least some traces of manipulation survive the whole chain linking the forger to the analyst, as we tried to demonstrate in the final chapter of our Thesis.

Although being aware of all its limits, we believe that image forensics applied to image evolution study can bring an important contribution in copyright, security and justice, so that it is easy to foresee an increasing interest in this topic in the near future.

References

[1] A. Piva, "An overview on image forensics," *ISRN Signal Processing*, vol. 2013, 2013.

[2] Z. Dias, A. Rocha, and S. Goldenstein, "First steps toward image phylogeny," in *IEEE International Workshop on Information Forensics and Security*, 2010, pp. 1–6.

[3] Z. Dias, A. Rocha, and S. Goldenstein, "Image phylogeny by minimal spanning trees," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 774–788, 2012.

[4] A. Joly, O. Buisson, and C. Frélicot, "Content-based copy retrieval using distortion-based probabilistic similarity search," *IEEE Transactions on Multimedia*, vol. 9, no. 2, pp. 293–306, 2007.

[5] Z. Dias, A. Rocha, and S. Goldenstein, "Large-scale image phylogeny: Tracing image ancestral relationships," *IEEE Multimedia*, vol. 20, no. 3, pp. 58–70, 2013.

[6] F. O. Costa, M. Oikawa, Z. Dias, S. Goldenstein, and A. Rocha, "Image phylogeny forest reconstruction," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 10, pp. 1533–1546, 2014.

[7] Z. Dias, A. Rocha, and S. Goldenstein, "Video phylogeny: Recovering near-duplicate video relationships," in *IEEE Workshop on Information Forensics and Security*, 2011, pp. 1–6.

[8] Z. Dias, S. Goldenstein, and A. Rocha, "Exploring heuristic and optimum branching algorithms for image phylogeny," *Elsevier Journal of Visual Coimmunication and Image Representation*, vol. 24, pp. 1124–1134, October 2013.

[9] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Elsevier Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

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

[11] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley, "Color transfer between images," *IEEE Computer Graphics Applications*, vol. 21, pp. 34–41, 2001.

[12] I. Sobel and G. Feldman, "A 3x3 isotropic gradient operator for image processing," *a talk at the Stanford Artificial Project in*, pp. 271–272, 1968.

[13] Rafael Gonzalez and Richard Woods, *Digital Image Processing*, Prentice-Hall, 3 edition, 2007.

[14] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, 1948.

[15] J. G. MacKinnon, *Numerical distribution functions for unit root and cointegration tests*, Institute for Economic Research, Queen's University, 1995.

[16] J.E. Tapia and C.A. Perez, "Gender classification based on fusion of different spatial scale features selected by mutual information from histogram of lbp, intensity, and shape," vol. 8, no. 3, pp. 488–499, March 2013.

[17] R. Bramon, I. Boada, A. Bardera, J. Rodriguez, M. Feixas, and M. Sbert., "Multimodal data fusion based on mutual information," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 9, pp. 1574–1587, September 2012.

[18] R. Battiti, "Using mutual information for selecting features in supervised neural net learning," *IEEE Transactions on Neural Networks*, vol. 5, no. 4, pp. 537–550, July 1994.

[19] P. Viola and W. M. Wells, "Alignment by maximization of mutual information," *International Journal of Computer Vision*, vol. 24, pp. 137–154, 1997.

[20] F. Maes, A. Collignon, D. Vandermueln, G. Marchal, and P. Suetens, "Multimodality image registration by maximization of mutual information," *IEEE Transaction on Medical Imaging*, vol. 16, pp. 187–198, 1997.

[21] K. A. Brownlee, *Statistical theory and methodology in science and engineering*, Wiley series in probability and mathematical statistics: Applied probability and statistics. Wiley, 1965.

[22] R. C. Gonzalez and R. E. Woods, *Digital Image Processing (3rd Edition)*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.

[23] J. Edmonds, "Optimum branchings," *Journal of Research of National Institute of Standards and Technology*, vol. 71B, pp. 48–50, 1967.

[24] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics bulletin*, pp. 80–83, 1945.

[25] A. C. Popescu and H. Farid, "Exposing digital forgeries by detecting traces of resampling," *IEEE Transactions on Signal Processing*, vol. 53, no. 2, 2005.

[26] M. Kirchner, "Fast and reliable resampling detection by spectral analysis of fixed linear prediction residue," *ACM Multimedia and Security Workshop (ACM MM&Sec)*, pp. 11–20, 2008.

[27] M. C. Stamm and K. J. R. Liu, "Forensic detection of image manipulation using statistical intrinsic fingerprints," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 3, pp. 492–506, 2010.

[28] J. Lukas and J. Fridrich, "Estimation of primary quantization matrix in double compressed JPEG images," in *Proc. of DFRWS*, 2003.

[29] H. Farid, "Exposing digital forgeries from JPEG ghosts," *IEEE Transaction on Information Forensics and Security*, vol. 4, no. 1, pp. 154–160, 2009.

[30] A.C. Popescu, *Statistical Tools for Digital Image Forensics*, Ph.D. thesis, Department of Computer Science, Dartmouth College, Hannover, 2005.

[31] T. Bianchi and A. Piva, "Reverse engineering of double JPEG compression in the presence of image resizing," in *IEEE International Workshop on Information Forensics and Security*, 2012, pp. 127–132.

[32] M. Stamm and K. J. R. Liu, "Forensic estimation and reconstruction of a contrast enhancement mapping," in *International Conference on Acoustics, Speech, and Signal Processing*, 2010, pp. 1698–1701.

[33] M. Stamm and K.J.R. Liu, "Blind forensics of contrast enhancement in digital images," in *IEEE International Conference on Image Processing*, 2008, pp. 3112–3115.

[34] Z. Lin, J. He, X. Tang, and C.-K. Tang, "Fast, automatic and fine-grained tampered JPEG image detection via DCT coefficient analysis," *Pattern Recognition*, vol. 42, no. 11, pp. 2492–2501, 2009.

[35] J. Fridrich, M. Goljan, and D. Hogea, "Steganalysis of JPEG images: Breaking the F5 algorithm," in *International Workshop on Information Hiding*. 2002, pp. 310–323, Springer-Verlag.

[36] J. Fridrich, "Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes," in *Information Hiding*, 2005, pp. 67–81.

[37] T. Bianchi and A. Piva, "Image forgery localization via block-grained analysis of JPEG artifacts," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 1003–1017, June 2012.

[38] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Statistics*, vol. 2, pp. 79–86, 1951.

[39] Z. Fan and R.L. de Queiroz, "Identification of bitmap compression history: Jpeg detection and quantizer estimation," *IEEE Transactions on Image Processing*, vol. 12, no. 2, pp. 230–235, 2003.

[40] T. Gloe, "Demystifying histograms of multi-quantised dct coefficients," in *IEEE International Conference on Multimedia and Expo*, 2011, pp. 1–6.

[41] Shuiming Ye, Q. Sun, and E.-C. Chang, "Detecting digital image forgeries by measuring inconsistencies of blocking artifact," in *IEEE International Conference on Multimedia and Expo*, 2007, pp. 12–15.

[42] D. Fu, Y. Q. Shi, and W. Su, "A generalized Benford's law for JPEG coefficients and its applications in image forensics," in *SPIE Conference on Security, Steganography, and Watermarking of Multimedia Contents*, 2007, vol. 6505.

[43] B. Li, Y. Q. Shi, and J. Huang, "Detecting doubly compressed JPEG images by using Mode Based First Digit Features," in *Multimedia Signal Processing*, 2008, pp. 730–735.

[44] Z. Dias, A. Rocha, and S. Goldenstein, "Toward image phylogeny forests: Automatically recovering semantically similar image relationships," *Elsevier Forensic Science International*, vol. 231, no. 1–3, pp. 178–189, 2013.

[45] I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo, L. Del Tongo, and G. Serra, "Copy-move forgery detection and localization by means of robust clustering with J-Linkage," *Signal Processing: Image Communication*, vol. 28, no. 6, pp. 659–669, 2013.

[46] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Springer International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[47] R. Toldo and A. Fusiello, "Robust multiple structures estimation with J-Linkage," in *Springer European Conference on Computer Vision*, 2008, pp. 537–547.

[48] H. Jégou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *Springer European Conference on Computer Vision*, 2008, pp. 304–317.

[49] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *IEEE International Conference on Computer Vision*, 2001, vol. 2, pp. 416–423.

[50] A. Opelt, A. Pinz, M. Fussenegger, and P. Auer, "Generic object recognition with boosting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 3, pp. 416–431, 2006.

[51] K. McGuinness and N. E. O'Connor, "A comparative evaluation of interactive segmentation algorithms," *Elsevier Pattern Recognition*, vol. 43, no. 2, pp. 434–444, 2010.

[52] M. Marszatek and C. Schmid, "Accurate object localization with shape masks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[53] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," in *ACM Special Interest Group on GRAPHics and Interactive Techniques*, 2003, pp. 313–318.

[54] Q. Liu, X. Cao, C. Deng, and X. Guo, "Identifying image composites through shadow matte consistency," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 1111–1122, 2011.

[55] H. Yao, S. Wang, Y. Zhao, and X. Zhang, "Detecting image forgery using perspective constraints," *IEEE Signal Processing Letters*, vol. 19, no. 3, pp. 123–126, 2012.

[56] M. Barni, A. Costanzo, and L. Sabatini, "Identification of cut & paste tampering by means of double-JPEG detection and image segmentation," in *IEEE International Symposium on Circuits and Systems*, 2010, pp. 1687–1690.

[57] M. Chen, J. Fridrich, M. Goljan, and J. Lukas, "Determining image origin and integrity using sensor noise," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 1, pp. 74–90, 2008.

[58] A. Swaminathan, M. Wu, and K. J. R. Liu, "Digital image forensics via intrinsic fingerprints," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 1, pp. 101–117, 2008.

[59] Z. C. Lin, J. F. He, X. Tang, and C. K. Tang, "Fast, automatic and fine-grained tampered JPEG image detection via DCT coefficient analysis," *Pattern Recognition*, vol. 42, no. 11, pp. 2492–2501, 2009.

[60] T. Bianchi, A. De Rosa, and A. Piva, "Improved DCT coefficient analysis for forgery localization in JPEG images," in *IEEE International Conference on Acoustic, Speech and Signal Processing*, 2011, pp. 2444–2447.

[61] T. Bianchi and A. Piva, "Image forgery localization via block-grained analysis of jpeg artifacts," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 1003–1017, 2012.

[62] P. Ferrara, T. Bianchi, A. De Rosa, and A. Piva, "Image forgery localization via fine-grained analysis of cfa artifacts," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 5, pp. 1566–1577, 2012.

[63] H. Farid, "Image forgery detection – a survey," *IEEE Signal Processing Magazine*, vol. 2, no. 26, pp. 16–25, 2009.

[64] J. A. Redi, W. Taktak, and J.-L. Dugelay, "Digital image forensics: a booklet for beginners," *Multimedia Tools and Applications*, vol. 51, no. 1, pp. 133–162, 2011.

[65] A. Swaminathan, M. Wu, and K.J. R. Liu, "Nonintrusive component forensics of visual sensors using output images," *IEEE Transactions on Information Foresics Security*, vol. 2, no. 1, pp. 91–106, 2007.

[66] A. Swaminathan, M. Wu, and K. J. R. Liu, "Digital image forensics via intrinsic fingerprints," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 1, pp. 101–117, 2008.

[67] H. Cao and A. C. Kot, "Accurate detection of demosaicing regularity for digital image forensics," *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 4, pp. 899–910, 2009.

[68] S. Bayram, H. T. Sencar, and N. Memon, "Classification of digital camera-models based on demosaicing artifacts," *Digital Investigation*, vol. 5, pp. 46–59, 2008.

[69] A. C. Popescu and H. Farid, "Exposing digital forgeries in color filter array interpolated images," *IEEE Transactions on Signal Processing*, vol. 53, no. 10, pp. 3948–3959, 2005.

[70] Andrew C. Gallagher, "Detection of linear and cubic interpolation in JPEG compressed images," *Computer and Robot Vision, Canadian Conference*, vol. 0, pp. 65–72, 2005.

[71] Na Fan, Cheng Jin, and Yizhen Huang, "A pixel-based digital photo authentication framework via demosaicking inter-pixel correlation," in *11th ACM Multimedia and Security Workshop*, 2009, pp. 125–129.

[72] A. C. Gallagher and T. Chen, "Image authentication by detecting traces of demosaicing," in *IEEE Computer Vision and Pattern Recognition Workshops*, 2008, pp. 1–8.

[73] A. E. Dirik and N. Memon, "Image tamper detection based on demosaicing artifacts," in *16th IEEE International Conference on Image Processing*, 2009, pp. 1497–1500.

[74] M. Kirchner, "Fast and reliable resampling detection by spectral analysis of fixed linear prediction residue," in *10th ACM Multimedia and Security Workshop*, 2008, pp. 11–20.

[75] M. Kirchner and T. Gloe, "On resampling detection in re-compressed images," in *First IEEE International Workshop on Information Forensics and Security*, December, pp. 21–25.

[76] B. Mahdian and S. Saic, "Blind authentication using periodic properties of interpolation," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 3, pp. 529–538, 2008.

[77] B. Mahdian and S. Saic, "A cyclostationarity analysis applied to image forensics," in *IEEE Workshop on Applications of Computer Vision*, 2009, pp. 389–399.

[78] D. Vazquez-Padin, C. Mosquera, and F. Perez-Gonzalez, "Two-dimensional statistical test for the presence of almost cyclostationarity on images," *17th IEEE International Conference on Image Processing*, pp. 1745–1748, 2010.

[79] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society: Series B 39*, pp. 1–38, 1977.

[80] S.G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674 –693, July 1989.

[81] T. Yamada, K. Ikeda, Yong-Gwan Kim, H. Wakoh, T. Toma, T. Sakamoto, K. Ogawa, E. Okamoto, K. Masukane, K. Oda, and M. Inuiya, "A progressive scan ccd image sensor for dsc applications," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 12, pp. 2044–2054, Dec 2000.

[82] Y.-F. Hsu and S.-F. Chang, "Statistical fusion of multiple cues for image tampering detection," in *Conference on Signals, Systems and Computers*, 2008, pp. 1386–1390.

[83] P. Zhang and X. Kong, "Detecting image tampering using feature fusion," *IEEE International Conference on Availability, Reliability and Security*, pp. 335–340, 2009.

[84] G. Chetty, J. Goodwin, and M. Singh, "Digital image tamper detection based on multimodal fusion of residue features," in *Advanced Concepts for Intelligent Vision Systems*, vol. 6475, pp. 79–87. 2010.

[85] G. Chetty and M. Singh, "Nonintrusive image tamper detection based on fuzzy fusion," *International Journal of Computer Science and Network Security*, vol. 10, no. 9, pp. 86–90, 2010.

[86] M. Fontani, T. Bianchi, A. De Rosa, A. Piva, and M. Barni, "A framework for decision fusion in image forensics based on Dempster-Shafer Theory of Evidence," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 4, pp. 593–607, 2013.

[87] M. Barni and A. Costanzo, "A fuzzy approach to deal with uncertainty in image forensics," *Signal Processing: Image Communication*, vol. 27, no. 9, pp. 998–1010, 2012.

[88] M. Fontani, E. Argones-Rua, C. Troncoso, and M. Barni, "The watchful forensic analyst: Multi-clue information fusion with background knowledge," in *IEEE International Workshop on Information Forensics and Security*, 2013, pp. 120–125.

[89] D. Cozzolino, F. Gargiulo, C. Sansone, and L. Verdoliva, "Multiple classifier systems for image forgery detection," in *Image Analysis and Processing*, vol. 8157, pp. 259–268. Springer, 2013.

[90] G. Shafer, *A Mathematical Theory of Evidence*, Princeton University Press, 1976.

[91] A. P. Dempster, "Upper and lower probabilities induced by a multivalued mapping," *Annals of Mathematical Statistics*, vol. 38, pp. 325–339, 1967.

[92] T. Bianchi and A. Piva, "Detection of non-aligned double JPEG compression with estimation of primary compression parameters," in *IEEE International Conference on Image Processing*, 2011, pp. 1929–1932.

[93] G. Chierchia, D. Cozzolino, G. Poggi, C. Sansone, and L. Verdoliva, "Guided filtering for PRNU-based localization of small-size image forgeries," in *IEEE International Conference on Acoustic, Speech and Signal Processing*, 2014.

[94] K. He, J. Sun, and X. Tang, "Guided image filtering," in *Computer Vision–ECCV*, pp. 1–14. Springer, 2010.

[95] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, and K. Toyama, "Digital photography with flash and no-flash image pairs," *ACM Transaction on Graphics*, vol. 23, no. 3, pp. 664–672, 2004.