Università degli Studi di Firenze
Corso di Dottorato in Ingegneria Informatica,
Multimedialità e Telecomunicazioni
Media Integration and Communication Center (MICC)
ING-INF/03

Image Forensics: source identification and tampering detection

Candidate
Irene Amerini

Supervisors
Prof. Vito Cappellini

Dott. Ing. Alessandro Piva

Dott. Ing. Roberto Caldelli

PhD Coordinator
Prof. Giacomo Bucci

ciclo XXIII, 2008-2010

sotto l'azzurro fitto
del cielo qualche uccello di mare se ne va;
né sosta mai: perché tutte le immagini portano scritto:
piú in lá!.
beneath the dense blue
sky, seabirds flash by, never
pausing, driven by images below:
"Farther, farther!"
(E. Montale, Maestrale, Ossi di seppia.)

Contents

Chapter 1

An Introduction to Digital Forensics

The use of scientific methods to gain probative facts in criminal investigations are referred to forensic sciences. This term has its etymologic roots in the Latin word "forum", which means "main square", a public place where various activities took place in Ancient Rome like politic discussions, meetings and also public hearings.

It's possible to subdivide all forensic sciences by their domain of evidence (see Figure 1.1); classical (analog) forensics deals with physical evidence, whereas digital forensics explore digital evidence.

In particular digital forensics science emerged in the last decade in response to the escalation of crimes committed by the use of electronic devices as an instrument used to commit a crime or as a repository of evidences related to a crime. A first definition for digital forensics science is given in 2001 during the first Digital Forensic Workshop and, for the sake of completeness, it is quoted in the following :
"The use of scientifically derived and proven methods toward the preservation, collection, validation, identification, analysis, interpretation, documentation and presentation of ***digital evidence*** *derived from digital sources for the purpose of facilitating or furthering the reconstruction of events found to be criminal, or helping to anticipate unauthorized actions shown to be disruptive to planned operations [3]."*
The term ***digital evidence*** means *"any probative information stored or transmitted in digital form that a party to a court case may use at trial [4]"*.

The use of digital evidence in the courts has increased in the past few

decades allowing for example the use of e-mails, digital photographs, word processing documents, instant message histories, internet browser histories, databases, the contents of computer memory, Global Positioning System tracks, and digital video or audio files. As with any evidence, the proponent of digital evidence must lay the proper foundation of such digital evidence and the courts sometimes require the authentication of the evidence called "for a more comprehensive foundation".

Anyways, the more comprehensive foundation remains good practice and there are not any guidelines to follow nowadays. The American Law Reports lists only a number of ways to establish the comprehensive foundation and in the United Kingdom the admissibility of computer generated and electronic evidence is governed by the Best Practice Guide drafted by the Association of Chief Police Officers.

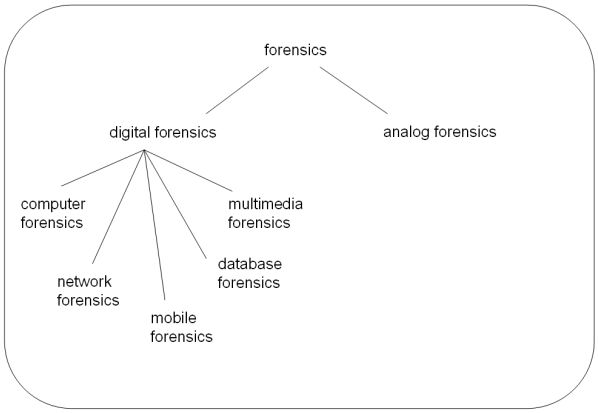


Figure 1.1: *Digital forensics and Analog forensics.*

Digital forensics science is divided into several sub-branches: *computer forensics, network forensics, database forensics, mobile device forensics and recently multimedia forensics* [5].

In *computer forensics*, forensic investigators want to extract probative facts from the computers involved. Forensic investigators typically follow a standard set of procedures: after physically isolating the computer in question to make sure it cannot be accidentally contaminated, investigators make a digital copy of the hard drive and all investigation is done on the digital copy. Investigators use a variety of techniques and proprietary forensic applications to examine the hard drive copy, searching hidden folders and unallocated disk space for copies of deleted, encrypted, or damaged files.

Network forensics analyses network events in order to discover security attacks and a forensic examination of a database may focus on identifying transactions within a database system that indicate evidence of a fraud (deleting some record in a database for example).

Mobile device forensics instead refers to digital evidence recovering from a mobile device or any digital device that has both internal memory and communication ability.

Finally *multimedia forensics* deals with digital representations of parts of reality, such as images, videos or audio files captured from a digital camera, a camcorder and so on. The main goal of multimedia forensics is to demonstrate that such digital evidence can be use in a trial because is reliable and authentic or otherwise demonstrate the contrary. How this goal is realized is described in detail in Chapter 2 where a deepened look into multimedia forensics techniques is given. In the following a quick overview of the problem is outlined focusing the attention on *Image Forensics*.

Photography lost its innocence many years ago and shortly after the first commercially available analog camera was introduced, photographs were being manipulated and altered. For example the portrait of the U.S. President Abraham Lincoln (around 1860) was a fake, having been created by splicing together the head of Lincoln with the body of the politician John Calhoun (Figure 1.2). Another example is reported in Figure 1.3 where in order to create a more heroic portrait of Benito Mussolini, the horse handler has been removed from the original photograph.

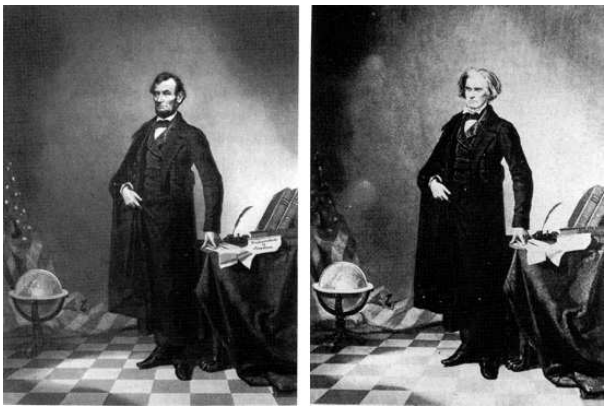


Figure 1.2: *The 1860 portrait of President Abraham Lincoln on the left and the politician John Calhoun on the right.*

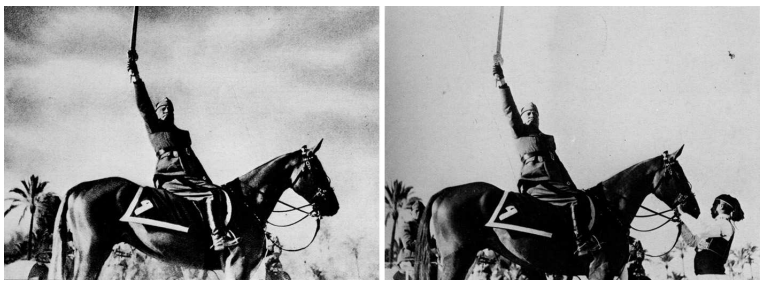


Figure 1.3: *The Benito Mussolini portrait (1942) with the original on the right.*

With the advent of digital cameras, powerful personal computers, and sophisticated photo-editing software, the manipulation of digital images is becoming much more common. Digital images are everywhere: on the covers of magazines, in newspapers, in courtrooms, and all over the internet but, given the ease with which images can be manipulated, we need to know if what we see is real or not. The tools for making forgeries have increased in sophistication, bringing the ability to make forgeries to anyone in fact in recent years, tampered images have affected science, law, politics, the media, and business. The tools for detecting forgeries, on the other hand, are only in the beginning of the development and there is a clear need for these tools in particular in the forensic domain.

Recently, there have been numerous examples of tampered images in newspapers and on magazine covers. Figure 1.4 and 1.5, for example, shows covers from popular magazines where the images have been manipulated.

Figure 1.4: *June 2010. This cover of The Economist shows the President Obama on the Louisiana beach inspecting the oil spill. The original photo, shot by Reuters photographer shows other two persons standing next to the President.*

(a) (b)

Figure 1.5: *a) June 2010. The text "38. BFD." was digitally added to the photo of seven-time Tour de France champion, Lance Armstrong.*
b) April 2005. The picture is a digital composite of two images of the actors Brad Pitt and Angelina Jolie.

Another example is showed in Figure 1.6 where a war photographer doctored a photograph that appeared on the cover of the Los Angeles Times in 2003, creating a picture from a composite of two images. The image in Figure 1.7) show an Iranian missile test appeared in 2008 on the front page of many major newspapers. After the publication of this photo, it was revealed that the second missile from the right was digitally added to the image in order to conceal a missile on the ground that did not fire.

Another picture (Figure 1.8) shows Prime Minister Benjamin Netanyahu (center left) and President Shimon Peres (center right) with members of the Cabinet. The Israeli newspaper Yated Neeman digitally removed two female Cabinet members from the photo and replaced them with male members.

Psicology studies show that people's memories of events can be altered by viewing doctored images and while some images might only tarnish the public opinion of a celebrity (Figure 1.5), in politics (Figure 1.6, 1.7, 1.8) or in science and law, tarnishing the public perception of images, could have more serious implications. In 2007, Missouri University professor R. Michael Roberts and co-authors retracted their paper (*Cdx2 Gene Expression and Trophectoderm Lineage Specification in Mouse Embryos*) published in Science journal after was revealed that images of the paper were doctored.

Figure 1.6: *2003. On the top the digital composite of a soldier appeared on the front page of the Los Angeles Times. On the bottom the original two source images.*

Figure 1.7: *July 2008: Iranian missile test. On the left the tampered image and on the right the orginal.*

Sometimes could be useful in some scenario to prove not only the image authenticity but also to know the origin of images presented as evidence. In law, the Unites States of American Child Pornography Prevention Act of 1996 (CPPA) outlawed virtual child pornography, i.e., images that appear to depict minors engaged in sexual acts but were created by computer. In 2002, the United States Supreme Court declared the CPPA to be in violation of the First Amendment. Their decision was based on the fact that no children are directly harmed in the production of virtual child pornography, and therefore, such images are protected under the right to freedom of speech. A side-effect of this is that people accused of child pornography can claim that the images are computer-generated. Therefore in a child pornography case,

Figure 1.8: *April 2009: Prime Minister Benjamin Netanyahu (center left), President Shimon Peres (center right), along with members of the Cabinet in the original image (up) and in the fake image (bottom).*

one could prove that certain imagery has been obtained using a specific camera and is not a computer-generated image relating the image to a suspect camera. In the same manner as bullet scratches allow forensic examiners to match a gun bullet to a particular barrel with reliability high enough to be accepted in courts, a digital equivalent of bullet scratches should allow reliable matching of a digital image to a digital camera.
Device identification could also be used when digital camcorders are used by pirates in movie theaters to obtain copies of reasonable quality that are subsequently sold on a black market and transcoded to low bit-rates for illegal distribution over the Internet. Forensic methods capable of determining that two clips came from the same camcorder or that two transcoded versions of one movie have a common source will obviously help investigators draw connections between different entities or subjects and may become a crucial piece of evidence in prosecuting the pirates.
Furthermore a forensic analysis could help the investigator to distinguish between an original multimedia content and an illegal copy of it. Different types of acquisition devices are involved in this scenario, from digital cameras, scanners, cellphones, PDAs and camcorders till photorealistic images or videos created with graphic rendering software. In all of these examples,

the authenticity and reliability of images is an issue and there is a need for solutions to address this problem.

1.1 Contributions

In this thesis, various subjects have been studied referring to source device identification and image tampering detection. In particular four novel techniques for *Image Forensics* will be presented in the following chpaters. For each method, we describe the conditions under which it is applicable, give a mathematical model, and bring experiments and results on real images to validate the methodology. In the first technique, presented in Chapter 3, a method to discern between digital camera and scanned images is exploited, and in Chapter 4 an analysis of different denoising filters is carried on with regard to source camera identification. In Chapter 5 the problem of grouping images belonging to a given set and coming from an unknown number of cameras is faced. Then in Chapter 6 the problem of detecting if a feigned image has been created is investigated; in particular, attention has been paid to the case in which an area of an image is copied and then pasted onto another zone to make a duplication or to cancel something that was awkward. In Chapter 7 is briefly disccussed a new multimedia forensics topic regarding the problem of ordering over time the outputs observed from a device (digital camera). This problem is addressed as “temporal forensics”.

Chapter 2

Multimedia Forensics

Multimedia forensics can be defined as the science that tries, by only analysing a particular digital asset, to give an assessment on such a content and to extract information that can be useful to address and support an investigation linked to the scene represented in that specific digital document. The basic idea behind multimedia forensics relies on the observation that both the acquisition process and any post-processing operation leave a distinctive imprint on the data, as a sort of digital fingerprint. The analysis of such a fingerprint may permit to determine image/video origin and to establish digital content authenticity.

Digital crime, together with constantly emerging software technologies, is growing at a rate that far surpasses defensive measures. Sometimes a digital image or a video may be found to be incontrovertible evidence of a crime or of a malevolent action. By looking at a digital content as a digital clue, Multimedia Forensic technologies are introducing a novel methodology for supporting clue analysis and providing an aid for making a decision on a crime. Multimedia forensic researcher community aimed so far at assisting human investigators by giving instruments for the authentication and the analysis of such clues. To better comprehend such issues let firstly introduce some application scenarios. Let's imagine a situation in which the action itself of creating a digital content (e.g. a photograph) implies an illegal action related to the content represented in the data (e.g. child pornography). In such a case, tracing the acquisition device that took that digital asset, can lead the judge to blame the owner of the "guilty" device for that action. Forensic techniques can help in establishing the origin/source of a digital media, making the "incriminated" digital content a valid, silent witness in the

court. A similar approach can be used in a different circumstance, in which a forensic analysis can help the investigator to distinguish between an original multimedia content and an illegal copy of it. Different types of acquisition devices can be involved in this scenario, from digital cameras, scanners, cellphones, PDAs and camcorders till photorealistic images or videos created with graphic rendering software. In this context, the possibility of identifying how that digital document was created may allow to detect illegal copy (e.g. digital cinema video recaptured by a camcorder). A more insidious digital crime is the one that attempts to bias the public opinion through the publication of tampered data. Motivations can spread from joking (e.g. unconvincing loving couple), to changing the context of a situation in which very important people are involved, or to exaggerating/debasing the gravity of a disaster image. Image forensic techniques can give a support in recognizing if, how and possibly where the picture has been forged.

Forensic tools work without any added information, the only features that can be evaluated are the ones intrinsically tied to the digital content. The basic idea behind multimedia forensic analysis relies on the observation that both the acquisition process and any post-processing operation leave a distinctive imprint on the data, as a sort of digital fingerprint. The estimation of such fingerprints really suggests how to evaluate the digital clue, turning it into an actual evidence.

It is the aim of this chapter to present the principles and the motivations of digital forensics (i.e. concerning images and videos), and to describe the main approaches proposed so far for facing the two basic questions: a) what is the source of a digital content? b) is such a digital content authentic or not? The chapter will be organized as it follows. The first section will introduce the reader to the basics of multimedia forensics; the different approaches for obtaining information from a digital content will be presented, as well as the diverse type of digital data that can be usually analyzed; then, the possible application scenarios that can benefit from forensic techniques will be described and an overview over the intrinsic digital fingerprints will be presented. The second and the third sections will be devoted to the analysis of the principal techniques exploited respectively for identifying the acquisition device of digital images and videos, and for assessing the authenticity of digital images. The fourth section will describe some counter-forensic technique applied to hide some modifications on the image in order to avoid the

digital camera identification or a forgery identification.[1]

2.1 Multimedia forensics: principles and motivations

Multimedia forensics can be defined as the science that tries, by analysing a digital asset, to give an assessment on such a content and to extract information that can be useful to address and support an investigation linked to the scene represented in that specific digital document. Multimedia forensics has to be able to develop efficient instruments to deal with the disparate digital devices that can generate images and, above all, with the different processing tools that allows also an unskilled user to manipulate digital goods. Hereafter two basic approaches are introduced, then the various kinds of data that multimedia forensic tools could have to face with are presented. After that, some possible application scenarios where these technologies could be claim to operate are described and finally a wide look to which are the possible digital fingerprints to be searched for in a multimedia content is given.

2.1.1 Possible approaches

When digital images (videos) had to be protected or their authenticity verified or, furthermore, their provenance tracked, the solution generally was to insert in the original data an embedded, usually unperceivable, information that permitted afterwards to determine what was happened, in which part of the content and, in particular application cases, by whom. This kind of techniques that can be grouped under the name of *digital watermarking* 1citebarni, follow an "active" approach, that is it is necessary to operate on the original document which has to be available from the beginning: this requirement is almost always hard to be satisfied. Embedding a watermark into an image, for instance, (see Figure 2.1) can be accomplished by applying some specific slight modifications to the original document I according to the information contained in the watermark W and ,often, to a private key K; after that the watermarked content IW is obtained.

[1] This survey has been published as book chapter in Handbook of Research on Computational Forensics, Digital Crime and Investigation: Methods and Solutions, IGI Global, Hershey, PA, USA, November 2009.

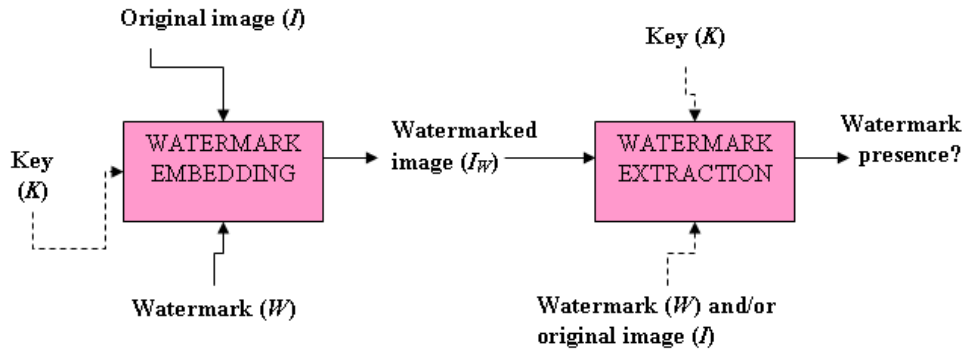


Figure 2.1: *Watermark embedding phase (left) and watermark extraction (right).*

If an assessment has to be performed to check if something has happened on the watermarked image, the detection phase is carried out by passing it, together with the private key K (if the algorithm is not blind the original image is needed too), to the detector that give an answer by re-extracting the watermark W or by comparing a verification parameter with a certain threshold.

For sake of completeness, also the cryptographic approach should be included within "active" method category. Such an approach uses digital signature for verifying author and time of signature and authenticating message contents. A digital signature is achieved by calculating a digest of the digital data by means of a hash function and encrypting it with a private key; such a signed digest is stored together with the image and can be used to prove data integrity or to trace back to its origin. There are some intrinsic weaknesses in this cryptographic approach. Firstly, the signal digest has to be tied to the content itself, e.g. by defining a proper format, and this makes impossible to use a different format, or to authenticate the data after D/A conversion. Secondly, the digest changes as soon as any modification is applied to the signal, making impossible to distinguish malicious versus innocuous modifications. Finally, cryptographic authentication usually does not allow a precise localization of tampering [6].

It is easy to understand that such a-posteriori evaluation can not be performed, for instance, on a common digital content obtained through the Internet (e.g. a video posted on YouTube, an image published on a newspaper web-site and so on). This kind of "active" technologies [7] can be adopted to manage data in a specific application context where additional information casting is feasible but are not able to deal with an open operative environment in which only a detection step is possible.

On the contrary, in this situation a "passive" methodology would be useful; with the term "passive" an approach which tries to make an assessment only having the digital content at disposal is to be intended. It is straightforward to realize that this kind of investigation is harder and has to be founded on the thorough analysis of some intrinsic features that should have/have not been present and are not/are now recognizable inside the observed data [8]. For sake of clarity: when a photomontage, for instance, has been performed to alter the content of a digital photo, to change the meaning of the represented scene, some traces of this operation are left somehow over the "new fake" image. These traces, although unperceivable, can result in the modification of the image structure such as anomalous pixel values (e.g. sequential interpolated values or strange continuous flat values) but also in inconsistencies within the image content itself such as anomalies in the illumination direction or in the presence of slight disproportionate object size with respect to the whole context. These are only some examples of the analysis approaches to be followed; further and deeper details will be discussed in the next sections.

2.1.2 Kinds of digital evidence and their characterization

Digital forensic tools are asked to recover crucial information by analysing digital evidences; their intrinsic features related to the way these documents have been created, stored and managed are important elements to be considered from the very first and, particularly, can determine which investigation methodology is more appropriate.

Most of the digital data digital forensic has to deal with are images: a three-channelled bi-dimensional array (single if grey level image) is all you can get to try to give answers. First of all, if images have been originated by a digital camera framing a real scene, it follows that its content, besides

presenting an intrinsic real structure, will contain all the imperfections and alterations induced by the specific acquiring sensor and by the processing block which generates the final stored file. As evidenced in Figure 2.2, when an image is taken from real life, light is focused by the lenses on the camera sensor which is a 2D array of CCD/CMOS which constitute the picture elements (pixels). Such elements are hit by the photons and convert them into voltage signals which are then sampled by an A/D converter.

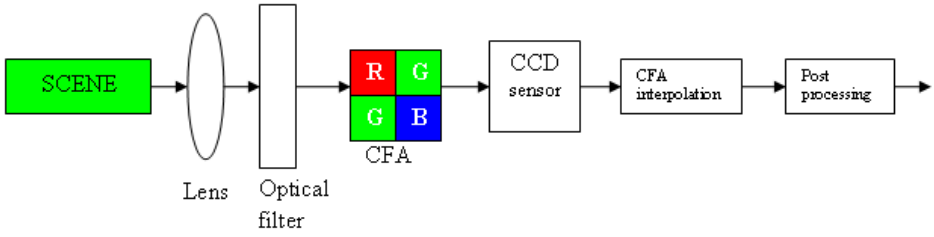


Figure 2.2: *Acquisition process in a photo camera.*

Anyway before reaching the sensor, the rays from the scene are filtered by the CFA (Colour Filter Array) which is a specific colour mosaic that permits to each pixel to gather only one particular colour. The sensor output is successively demosaicked (i.e. interpolated) to obtain all the three colours for each pixel and then this signal undergoes additional processing such as white balance, gamma correction, image enhancement and so on; after that is stored to the camera memory in a customized format, although, for commercial devices, JPEG format is usually preferred.

It is now easier to understand that the characteristics of each operation and the properties of every element, from the framed scene to the final image file, influence the digital data. In literature, in fact, there are techniques that have investigated the presence of a specific CFA [9] within the image texture to go back to the brand of the camera that had taken a certain photo and

other methods which have proposed to study the JPEG quantization coefficients to verify if an image had undergone a second compression thus revealing a possible tampering [10]. On the other side, many are the approaches based on the analysis of the anomalies left by the device over the image such as scratches on the lenses, defective pixels, etc.. In particular, attention has been paid to the sensor noise and among all, dark current, shot noise, thermal noise and so on, PRNU noise (Photo Response Non-Uniformity) is one of the most interesting for forensic applications. PRNU presence is induced by intrinsic disconformities in the manufacturing process of silicon CCD/CMOSs [11]. Such a noise is a 2D systematic fingerprint which characterized each single sensor, that is two cameras of the same brand and model will leave two different traces on the digital contents they acquire. So it is not properly a random noise because it is a deterministic bidimensional template which is superimposed to each taken image.

However images, needing a forensic analysis, can be, not only still, but may also be part of a video sequence; in this circumstance the data to be controlled have a temporal dimension too that has to be taken into account although most of the considerations made for digital photos regarding the presence of PRNU noise pattern and the CFA related to the acquisition phase, can be directly extended to the case of videos [12, 13]. It is anyway fundamental to point out that the huge amount of available data can suffer different kinds of manipulations with respect to static ones, in particular frames can be skipped or interpolated and inserted to modify the meaning and to alter the original duration of the sequence. Furthermore a clip, coming from another recording but of similar content, could be added to the video in a not-annoying manner to change the whole represented story. Forensic analysis has to be concentrated on aspects such as inter-frame PRNU noise correlation and MPEG-X re-coding.

Another kind of images that can constitute a digital evidence to be checked, in addition to those ones acquired with a photo camera or with a camcorder, might come from a scanning operation. This means that a printed document (e.g. the cover of a magazine or a real-life photo) located in a flatbed scanner has been illuminated row by row by a sliding monodimensional sensor array to originate the digital data [14]. The final file format is usually customizable but often is JPEG or PNG. In this case, due to the diversity of the device and to the digitization process, other elements, in addition to those already discussed for cameras, can be considered during

the forensic analysis to highlight possible traces of digital asset misuse. For instance, the presence over the image of a 1-D noise pattern, instead of a bidimensional, could be an indicator of image origin and what's more, the direction (vertical or horizontal) of such mono-dimensional periodicity could evidence which has been the scanning manner. Another interesting aspect to control could be the existence of some pieces of dirt that were settled over the scanner plate or of small scratches over the scanner glass that during acquisition have become integral part of the image itself.

Finally it is worthy to spend some words on another type of images digital forensic tools could have to face with: these are computer-generated images. Many are the software that allow to create digital photorealistic pictures that are undistinguishable with respect to those ones acquired by a camera (http://area.autodesk.com/index.php/fakeorfoto). These systems offer the possibility to build up a completely new image or to arrange a believable photomontage merging parts of a real photo with elements synthetically generated. To do this as much actual as possible various are the instruments that are usable and the superimposition of artificial noise is only one of the shrewdness a skilled user could put in practice to develop his fake content. The basic idea to be followed when dealing with this kind of images is to extract significant features which give an indication of the intrinsic realism of the image.

2.1.3 Application scenarios

It is now interesting to consider which can be the possible application scenarios for digital forensic technologies and which could be the questions they can give answers to. Though in literature many have been the fields where digital forensic tools were call to operate, two are the basic categories of usage: "identification of the source" and "detection of forgeries", these two aspects will be debated in detail in the following sections of this chapter.

With the term "identification of the source" it is intended the forensic procedure to determine which is the origin where the digital image comes from. In particular, it is good to split this issue into two sub-cases. In the first sub-case the aim is to recognize which is the device that has produced that digital asset, that is if the digital content has been generated by a photo-camera (video-camera), by a scanner or was computer-generated. To achieve this target, though different approaches exist, the basic ideas are to search

over the digital image for traces of the specific acquisition process and for the presence/absence of realistic characteristics within the digital data, this last mainly for distinguishing a computer generated image. On the other side, the second sub-case concerns with the individuation, within a certain set of devices, of which one has created that image. For example, taken a group of photo-cameras (scanners or video-cameras) try to discern which camera (brand and model) has taken that picture. Usually to perform this purpose is necessary to previously extract some information featuring each apparatus and this is done by constructing a sort of identifying fingerprint through the analysis of a certain number of digital contents (training set) produced by that device. Well-known procedures are based on SVM (Support Vector Machine) or on noise pattern correlation.

The second principal application scenario for digital forensic is the "detection of forgeries"; in this case it is required to establish if a certain image is authentic or has been artificially created by means of a manipulation to change its content. The aim of this modification could be very disparate ranging from commercial applications like to make an untrue journalistic scoop or to realize a pseudo-realistic advertisement clip, to some others much more crucial ones such as to alter the judgement in a trial where the image has been accepted as digital evidence or to produce satellite photos to assess that nuclear arms are stocked in a certain territory. Anyway it is important to point out that one of the main hurdles to this kind of analysis is the dimension of the forged part with respect to the whole image size. On the contrary, it is not to underestimate that a mimicking action often has to lead to a substantial alteration of the meaning of the represented scene and this is not always achievable with the exchange of a few pixels.

2.1.4 Intrinsic digital fingerprints

Even if forensic technologies are usually applied for different purposes (as previously described), actually it is possible to evidence how a common approach is followed by almost all the forensic algorithms proposed so far, regardless of their application for source identification or tampering detection. In particular, digital forensics is based on the idea that inherent traces (like digital fingerprints) are left behind in a digital media during both the creation phase and any other successively process [15]. By resorting only on the analyzed data, without any previously embedded information (pas-

sive approach) and without the knowledge of the related original data (blind method), forensic techniques capture a set of intrinsic information carried out by the digital asset by means of different analysis methods, i.e. statistical, geometric, etc.

Several kinds of digital fingerprints are taken into account for the forensic analysis, a possible classification of such fingerprints can be made by dividing them in three categories: digital traces left by the in-camera processing and those left by the out-camera processing and the fingerprints related to the features of the framed scene. In particular it is to be intended:

1. in-camera fingerprints: each component in a digital acquisition device modifies the input and leaves intrinsic fingerprints in the final output, due to the specific optical system, color sensor and camera software; furthermore, images and in particular natural images, have general characteristics, regardless of the content, such as inherent noise or behaviour of the luminance or statistical properties that can be seen as inherent fingerprint;

2. out-camera fingerprints: each processing applied to digital media modifies their properties (e.g. statistical, geometrical, etc.) leaving peculiar traces accordingly to the processing itself.

Let us note that previous fingerprints are independent off the content of the analysed data: e.g. the trace left by a given camera is the same even if different subjects have been acquired. On the contrary there is a third fingerprint category considering features related to the content of the image itself, namely:

1. scene fingerprints: the world, the photo coming from, has specific properties depending on the content, like lighting properties, which characterize the reproduced scene.

After choosing the specific fingerprint, generally the procedure is to select some properties of the considered fingerprint, to explore relative parameters, and to make a decision basing on either classification or estimation procedures. In particular, in the case of source identification these traces are usually extracted and then compared with a dataset of possible fingerprints specific for each kind/model/brand of acquisition devices, in order to link

the digital data to the corresponding source. On the other hand, according to the purpose of forgery detection, the idea is to detect non-uniformity or breaking of such fingerprints within the considered data; specifically, the media is usually block wise analysed and for each block the chosen fingerprints or, better, their properties or parameters, are extracted and compared each other. It is obvious that for the source identification only the first category of traces, the in-camera fingerprints, will be taken into account, whereas for integrity verification all the three categories can be exploited.

Next sections will be devoted to the two main purposes digital forensics is exploited for: acquisition device identification and integrity verification; what kind of digital fingerprint is taken into account and how it is used for the specific aim will be debated for providing a general overview of the principal approaches followed by multimedia forensics. In particular, in the next section, focused on the source identification, the so-called in-camera fingerprints are deeply analysed and their characteristics exploited for acquiring information about data origin. While the successive section focuses on tampering detection, by starting from the specific application of in-camera fingerprints to such a task and then the usage of the other two kinds of fingerprints (out-camera fingerprints and scene fingerprints) is debated.

2.2 Techniques for acquisition device identification

Techniques for device identification are focused on assessing digital data origin (images or videos). In particular two are the main aspects to be studied: the first one is to understand which kind of device has generated those digital data (e.g. a scanner, a cell-phone, a digital camera, a camcorder or they are computer-generated) and the second one is to succeed in determining the specific camera or scanner that has acquired such a content, recognizing model and brand (Figure 2.3).

Digital images, can be stored in a variety of formats, such as JPEG, GIF, PNG, TIFF, and the format can be as informative as the image. For example JPEG files contain a well-defined feature set that includes metadata, quantization tables for image compression and lossy compressed data. The metadata describe the source of the image, usually includes the camera type, resolution, focus settings, and other features [16]. Besides when RAW format

Figure 2.3: *The source identification problem.*

is used, the camera creates a header file which contains all of the camera settings, including (depending on the camera) sharpening level, contrast and saturation settings, colour temperature / white balance, and so on. The image is not changed by these settings, they are simply tagged onto the raw image data.

Although such metadata provide a significant amount of information it has some limitations: they can be edited, deleted and false information about the camera type and settings can be inserted. So it is important to provide a reliable source identification regardless of the type of metadata information ; such passive approach will be explored in the following.

This section will be dedicated to the analysis of the principal solutions exploited for identifying the acquisition device of digital images and videos exploring the general structure and sequence of stages of image formation pipeline, grounding on the physics and operations of the device under examination. These techniques aim at analysing those operations in order to find a fingerprint for the device (the so called in-camera fingerprints) in term of the presence of an identifying mark due to the color filter array (CFA) interpolation, sensor imperfections and lens aberration, In this section techniques based on the extraction, from images belonging to different categories (e.g. scanned images, photos, video etc.), of some robust intrinsic features that are typical of a particular devices classes will be explored. Generally these features can be used to train a classifier (e.g. SVM); when training is

performed and whether features grant a good characterization, the system is able to classify the digital asset. Hereafter, it will be shown how all these techniques do not work only for digital cameras but also for scanner and camcorder identification and also to distinguish between a photographic and a computer graphic image.

2.2.1 Color Filter Array and Demosaicking

In digital cameras with single imaging sensors (the most diffuse on the market) the Color Filter Array (CFA) covers the CCD sensor. Several patterns exist for the filter array (see Figure 2.4), the most common array is the Bayer CFA. Since the CFA allows only one color to be measured at each pixel this means that the camera must estimate the missing two color values at each pixel, this estimation process is known as "demosaicking".

Figure 2.4: *Examples of CFA patterns.*

There are several commonly used algorithms for color interpolation and each manufacturer employs a specific algorithm for this purpose. Given an output image I, the techniques for acquisition device identification are focused on finding the color filter array pattern and the color interpolation algorithm employed in internal processing blocks of a digital camera that acquired image I.

One well-known approach [9] assumes to know the CFA used in a digital camera based on the fact that most of commercial cameras use RGB type of CFA with a periodicity of 2x2.

The image I after the CFA sampling becomes:

$$I_s = \begin{cases} I\,(x, y, c)\,,\ if\ t\,(x, y) = c \\ 0, otherwise \end{cases} \tag{2.1}$$

where *t(x,y)* is the CFA pattern and c (colour) can be R, G and B.

Then the intermediate pixel values, corresponding to the points where $I_s\,(x, y, c) = 0$ in (2.1) are interpolated using its neighboring pixel values.

The digital forensic method proposed in [9] , for every CFA pattern t in a search space, estimates the color interpolation coefficients in different types of texture of the image (smooth, horizontal gradient and vertical gradient image regions) through a linear approximation.

Using the final camera output I and the assumed sample pattern t, it is possible to identify the set of locations in each color channel of I that are acquired directly from the sensor array. The remaining pixels are interpolated with a set of linear equations in terms of the colors of the pixel captured directly in each types of region. Then the algorithm reconstructs the input image I using the corresponding coefficients in each regions to obtain estimated final output image $\hat{I}$for all the CFA patterns in the search space. At this point the CFA pattern that minimizes error between I and $\hat{I}$is found by computing a weighted sum of the errors of the three color channels.

The color interpolation coefficients estimated from an image and the proposed CFA can be used as features to identify the camera brand utilized to capture the digital image. So a support vector machine (SVM) classifier is trained and then used to identify the interpolation method concerning different digital camera brands. The camera model is more difficult to detect because the color interpolation coefficients are quite similar among camera models and hence it is likely that the manufacturer uses similar kinds of interpolation methods. Furthermore, others limitations to the method exist: only RGB CFA is considered and then this technique does not permit to distinguishing Super CCD cameras because those digital cameras do not employ a square CFA pattern; moreover there is a misclassification around the smooth regions of the image, in fact similar techniques, such as bicubic interpolation, around smooth region in almost all commercial cameras are used.

As explained before, at each pixel location of a CFA interpolated color image, a single color sample is captured by the camera sensor, while the other two colors are estimated from neighboring ones. As a result, a subset

of samples, within a color channel, is correlated to their neighboring samples. This form of correlation is expressed by the linear model:

$$f(x,y) = \sum_{u,v=-N}^{N} \alpha_{u,v} f(x+u, y+v) \tag{2.2}$$

In the above equation, $\alpha_{u,v}$ are the coefficients of the model parameters and N is the number of correlated pixel. Since the color filters in a CFA are typically arranged in a periodic pattern (see again Figure 4), then a periodic correlation is introduced.

The probability maps of the observed data obtained from the Expectation Maximization (EM) algorithm can be employed to detect if a color image is the result of a CFA interpolation algorithm and the linear coefficients, $\alpha_{u,v}$, returned by the EM algorithm, can be used to distinguish between different CFA interpolation techniques [17, 18].

When observed in the frequency domain, these probability maps yield to peaks at different frequencies with varying magnitudes indicating the structure of correlation between the spatial samples. Then a classifier is designed on the basis of the two sets of features: the set of weighting coefficients obtained from an image, and the peak locations and magnitudes in frequency spectrum.

This method does not work in case of cameras of the same model, because they share the same CFA filter and interpolation algorithm, and also for compressed image or modified image (gamma corrected, smoothed) because these artefacts suppress and remove the spatial correlation between the pixels due to CFA interpolation.

2.2.2 Imaging Sensor Imperfections

This class of approaches for source matching aims at identifying and extracting systematic errors due to imaging sensor, which appear on all images acquired by the sensor in a way independent by the scene content.

There are several sources of imperfections and noise that influence the image acquisition process [19]. When the imaging sensor takes a picture of an absolutely evenly lit scene, the resulting digital image will still exhibit small changes in intensity among individual pixels.

These errors include sensor's pixel defects and pattern noise this last has two major components, namely, fixed pattern noise and photo response

non-uniformity noise (PRNU).

The defective pixels can be used for camera identification as described in [20]. This type of noise, generated by hot or dead pixels, is typically more prevalent in cheap cameras and can be visualized by averaging multiple images from the same camera. However, many cameras post-processing remove these types of noise, then this technique cannot always be used.

So, for a reliable camera identification, the idea is to estimate the pattern noise.

The fixed pattern noise (FPN) refers to pixel-to-pixel differences when the sensor array is not exposed to light (so called dark current) and also depends on exposure and temperature. The FPN is used for source camera identification in [21] but it is an additive noise and some middle to high-end consumer cameras suppress this noise by subtracting a dark frame from every image they take. On the basis of this consideration, photo-response non-uniformity noise (PRNU), that is the dominant part of the pattern noise in natural images, is usually searched for. The most important component of PRNU is the pixel non-uniformity (PNU), which is defined as different sensitivity of pixels to light. The PNU is caused by stochastic inhomogenities present in silicon wafers and other imperfections originated during the sensor manufacturing process. As such, it is not dependent on ambient temperature and appears to be stable over time. Light refraction on dust particles, optical surfaces and properties of the camera optics, which also contribute to the PRNU noise, are generally low spatial frequency components not characterizing the sensor and therefore not usable for source identification. Finally the noise component to be estimated and to be used as intrinsic characteristic of the sensor (fingerprint) is the PNU. It is also possible to suppress this kind of noise using a process called flat fielding [19], in which the pixel values are first corrected for the additive FPN and then divided by a flat field frame obtained by averaging images of a uniformly lit scene, but consumer digital cameras do not flat-field their images because it is difficult to achieve a uniform sensor illumination inside the camera.

To continue the discussion, it's necessary to give a mathematical model of image acquisition process. The digitized output of the sensor l can be expressed in the following form (before any other camera processing occurs):

$$l = k(s + p) + r + d \tag{2.3}$$

where s is the signal if no other noise sources exist, p is the random shot

noise, r is the additive random noise (represented by the read-out noise, etc.) and d is the dark current.

The factor k is close to 1 and captures the PRNU noise, which is a multiplicative noise.

Because details about the processing are not always easily available (they are hard-wired or proprietary), generally is needed to use a simplified model that captures various elements of typical in-camera processing. A more accurate model tailored to a specific camera would likely produce more reliable camera identification results at the cost of increased complexity.

The simplify sensor output model described in [22] results in the following vector form:

$$l \ = \sigma^{\gamma} \cdot [(1 + \Gamma)\, \Upsilon + \Pi]^{\gamma} + \theta_q \tag{2.4}$$

In Equation 2.4, Y is the incident light intensity on the sensor, σ is the color channel gain and γ is the gamma correction factor (typically, γ ò 0.45). The gain factor σ adjusts the pixel intensity level according to the sensitivity of the pixel in the red, green, and blue spectral bands to obtain the correct white balance. The multiplicative factor Γ is a zero-mean noise-like signal responsible for PRNU. Finally, Π is a combination of the other noise sources including the dark current, shot noise, and read-out noise, and $\theta_q$is the quantization noise.

Assuming that either the camera that took the image is available to the forensic analyst or at least some other (non-tampered) images taken by the camera are available, the PRNU term Γ, can be estimated from a set of N images taken by the camera. To improve the SNR between the PRNU term and observed data l, a host signal rejection is performed by subtracting (pixel by pixel) the denoised version *(ld)* of l, who can be obtained by using a denoising filter usually implemented through wavelet based algorithm [1].

$$Z = l - l_d \tag{2.5}$$

Since the image content is significantly suppressed in the noise residual Z, the PRNU can be better estimate from Z than from l, so Z is designated as the reference pattern and serves as an intrinsic signature of the camera. To identify the source camera, the noise pattern from an image is correlated with the known reference patterns from a set of cameras. The camera corresponding to the reference pattern giving maximum correlation is chosen to

be the source camera that acquired that image.

This type of approach is used also for video source identification [12] by estimating the PRNU from a video segment and then calculating the correlation with the reference pattern from a different segment of a video clip. The method described above shows poor performance when digital image are cropped, scaled or digital magnified so an improved method for source camera identification based on joint estimation and detection of the camera photo response non uniformity has been developed in [23]. The detector is obtained using the generalized likelihood ratio test and has the form of a cross-correlation maximized over the parameters of the geometrical transform.

With regard to the identification between synthetic image and photographic image a method is described in [24], based on the observation that in computer generated images occurs a lack of the sensor's pattern noise artefacts due to the software generation of the image.

Furthermore a technique based on PRNU estimation, for classification of scanned and non-scanned images, is outlined in [14, 25], based on the difference in the dimension of the sensor array (scanner sensor is a one dimensional sensor array, see previous section). This technique extracts a row reference noise pattern from a single scanned image by averaging the extracted noise (via denoising) over all rows and then a procedure like [22, 26] is used, based on the computation of correlation between the scanner reference pattern and the noise pattern from an image.

2.2.3 Lens Aberration

Due to the design and manufacturing process, lenses produce different kinds of aberrations in images. Generally two of them are investigated to solve the problem of source device identification: lens radial distortion [27] and chromatic aberration [28].

To reduce manufacturing cost, most of digital cameras are equipped with lenses having almost spherical surfaces that introduce radial distortions.

The radial distortion causes straight lines in the object space rendered as curved lines on camera sensor and it occurs when there is a change in transverse magnification$M_t$with increasing distance from the optical axis. The degree and the order of compensation of such a distortion vary from one manufacturer to another or even in different camera models by the same

manufacturer. As a result, lenses from different cameras leave unique imprints on the captured pictures.

The lens radial distortion can be written as:

$$r_u = r_d + k_1 r_d^3 + k_2 r_d^5 \tag{2.6}$$

wherer_u andr_d are the undistorted radius and distorted radius respectively. The radius is the radial distance$\sqrt{x_2 + y_2}$of a point (x, y) from the center of distortion (the centre of an image). The goal in the method proposed in [27] is to find the distortion parameters $k_1$andk_2 that constitute the fingerprint to identify source camera following the Devernay's straight line method.

However this method fails if there are no straight lines in the image and also if two cameras of the same model are compared. Besides it is also possible to operate a software correction in order to correct the radial distortion on an image.

The second type of aberration investigated to solve the source identification problem is the chromatic aberration. Chromatic aberration is the phenomenon where light of different wavelenghts fail to converge at the same position of the focal plane. There are two kind of chromatic aberration: longitudinal aberration that causes different wavelenghts to focus at different distances from the lens, while lateral aberration is attributed at different positions on the sensor. In both cases, chromatic aberration leads to various forms of color imperfections in the image. Only lateral chromatic aberration is taken into consideration in the method described in [28] for source identification. Chromatic aberration causes misalignment between the RGB channels so the task is to estimate the distorted parameters to compensate for the distortion maximizing the mutual information among the color channels. Then these parameters are used in [28] to identify source cell phone through the use of a SVM classifier.

2.2.4 Others Approaches

There are other approaches for source identification using a set of suitable digital data intrinsic features designed to classify a device model. These features can be statistical, geometrical and color features.

In [29] a set of features are calculated, they are composed by suitably chosen image quality metrics (IQM) evaluated between an input image and its filtered version using a low-pass Gaussian filter, and integrated with color

features (deviation from gray, inter-band correlation, gamma factor), and wavelet coefficient statistics. These features are used to construct multi-class classifier with images coming from different cameras, but it is demonstrated that this approach does not work well with cameras with similar CCD and it requires images of the same content and resolution.

Another group of selected features is based on the assumption that proprietary CFA interpolation algorithm leaves correlations across adjacent bitplanes of an image. Binary similarity measures (BSM) are metrics used to measure such a similarity. In [30] the authors differentiate between cell-phone camera models by using BSM features in conjunction with IQM features. In the approach described in [31], High-Order Wavelet Statistic (HOWS) features are added to the features used in [30] to distinguish among various brands of cell-phone cameras.

Other techniques exist to solve the classification problem between synthetic and "real" images. The method in [32] proposes a wavelet based statistical model to extract features from the characteristic functions of wavelet coefficient histograms. The previous approach is then extended in [33] by proposing new features to detect the use of Bayer color filter array during demosaicking [17, 18]. These features are incorporated with the features in [34] that capture the statistical regularities of natural images in terms of statistics of four level discrete wavelet transform coefficients.

A new set of features is taken into account for scanner identification in [35] because, generally, features are extracted without specifically taking the scanning process into consideration. The same features, with the addition of color interpolation coefficients, are proposed to identify images produced by cameras, cell-phone, scanners and computer graphics [36]. These features have been chosen in particular to distinguish camera form scanner because the CCD line sensor in a scanner consists of three lines for each color (red, green, blue), so in a scanner acquisition process no color interpolation is needed.

Another set of features has been built in [25] for classifying scanner, computer generated and digital camera due to the physical characteristic of the image sensor. In fact for a scanner, the fixed component of the noise should be nearly identical for all the rows of a scanned image due to mono dimensional image sensor, and for the same reason should be different for all the columns. Then the statistics of row correlation will differ from those of column correlation. Row correlation is defined as the correlation of each

row of the image with the estimated row reference pattern calculated as average of the noise of the reference image over all rows. So the first order statistics (mean, median, mode, maximum and minimum) and the higher order statistics (variance, kurtosis and skewness) of the row correlation and column correlation are used to generate the features vector for each image and also a measure of similarity among the rows or columns of the reference pattern noise are considered (Khanna, 2007 b) to design a SVM classifier.

2.3 Techniques for assessing image integrity

Information integrity is fundamental in a trial: a verdict must be returned after considering a set of evidences and the authenticity of such proofs should be assured before making a decision. On one hand witnesses and their assertions constitute a type of evidence; on the other hand, concrete objects, e.g. a weapon, represent another type of proof, so to speak "real" evidence. In this latter category can be included all the information belonging to the crime scene, and such information have been often captured and stored by means of pictures. If pictures are just representative of the real world, then they can be considered as authentic evidences. But, it is clear that the advent of digital pictures and relative ease of digital image processing makes today this authenticity uncertain. In this scenario, an efficient assessment of the integrity of digital information, and in particular of digital images, plays a central role.

But, what does integrity mean? In a strong sense, the image must be only the outcome of an acquisition of a real world scene, without any successively processing; in a wide sense, the image must accordingly represent a real world scene and even if some processing has been probably applied, the "meaning" of the scene must not be altered.

Once evidence passes from the real world of three dimensional objects to a digital image, we lose the origin of information and we can not trust any more what we are seeing, even if the content is advertised as real. Several image processing tools are nowadays easily usable for almost everybody; let only consider that Adobe PhotoShop is already licensed to many millions of users worldwide. With such programs, a great deal of operations is allowed to affect digital photographic files: person images can be moved in different contexts; objects can be deleted from scenes; particular details can be cloned within the photograph; computer graphic objects can be added to the real

scene. All these manipulations become more and more sophisticated thus making the alteration virtually imperceptible; furthermore, establishing the authenticity of images is a key point for being able to use digital images as critical evidence.
Digital forensics assume that images are intrinsically characterized by specific pattern due to the creation process and to any other process suffered after image creation. To properly individuate possible modifications, the image forensic approach considers that such intrinsic fingerprints inside images are distinguishable due to the different applied image processing, or that the original traces have been altered due to a tampering, thus losing their uniformity. So, different digital fingerprints are taken into account and studying their characteristics it is possible to verify if an image has undergone some tampering and even detect the suffered processing. Referring to the wide sense meaning of integrity (i.e. the digital photograph is a congruous representation of the captured "real" world), a lot of processing non-affecting the semantic (e.g. JPEG compression or recompression, brightness adjustment, gamma correction, etc.) can be erroneously revealed as tampering. Therefore, detection of image alteration does not necessarily prove malicious tampering, but surely questions about the content of the image and helps for further analysis.
In the following, we are going to discuss the technological approaches proposed in literature so far for verifying digital image authenticity; this discussion is structured again according to the classification of digital fingerprints previously introduced in this chapter where the three kinds of traces are categorized: in-camera fingerprints (described for their exploitation in source identification), out-camera fingerprints and scene fingerprints. Specifically, in the first and third case, forensic techniques search for some breaking or inconsistencies of such traces, whereas in the second case fingerprints are used for identifying a specific processing. As already mentioned, detection of image processing does not necessarily prove malicious tampering, but surely proves that some manipulation occurred after image creation.

Because of the great variety of existing methodologies devoted to this purpose, we have decided to provide only some hints of each analyzed technique, to allow the interested reader to get useful information and to possibly deepen his study by following the bibliographic references.

2.3.1 In-camera fingerprint breaking

Basically, the acquisition process is analysed and peculiarities left by some component of the chain are considered as intrinsic fingerprints (in-camera fingerprints) that characterize the kind or even the model or brand of acquisition devices. In particular, in the previous section three main components (namely color filter array, sensors and lens) are considered with their related fingerprints, that are:

1. the Color Filter Array (CFA) and its related demosaicking process;
2. the sensor imperfection and its related pattern noise;
3. the lens aberration and its related chromatic aberration.

On the basis of the previous analysis, we now consider how the traces left by such components can be exploited for tampering detection.
In the case of *CFA* the correlations between pixels introduced by the specific algorithm for the color interpolation are analysed in order to verify if these properties are broken in certain areas, thus revealing possible tampering [37, 15]. The works in [38, 11] propose a method to detect the *camera pattern noise* present in a given image: the inconsistency of camera pattern noise in some regions of digital image reveals the non integrity of the content; the proposed approach requires either the camera which produced the image or a set of images produced by the same camera, thus making such an algorithm non blind. Regarding the lens aberration, in [39] the authors consider in particular the *chromatic aberration* that leads to various forms of color imperfections in the image: when these alterations fail to be consistent across the image, a tampering can be supposed to be happened.
Besides the above mentioned fingerprints, there are other in-camera traces that have been used for integrity verification. Basically, also for such algorithms a block-based analysis is computed for evidencing the coherence/incoherence of the extracted parameters on the whole image.
The image irradiance (light energy incident on the image sensors) is related to the image intensity (the final output image) by a non-linear camera response function (*CRF*), that is a characteristic of each camera. The estimation of the CRF on different regions of the analysed image and the evaluation of consistency/inconsistency between such estimated CRFs, provides a good

method for deciding if the image is likely to be authentic or spliced [40, 41, 42].

The last step of the acquisition process is usually a *JPEG compression* to reduce storage space of the output image. Such a compression leaves unique fingerprints due to the particular quantization matrix used by the specific camera, and serves as a "fragile watermark" enabling the detection of changes within the image. In [43] authors propose to detect possible manipulations by investigating the compatibility of 8×8 pixel blocks with a given quantization matrix; whereas in [44] an algorithm is developed for automatically locating the tampered regions.

The discrepancy in the signal-to-noise ratio (*SNR*) across the image can also be considered as a sign for possible tampering. Digital images have an inherent amount of noise introduced either by the imaging process or digital compression, and such a noise is typically uniform across the entire image. If two images with different noise levels are spliced together, or if small amounts of noise are locally added to conceal traces of tampering, hence changes in the SNR across the image can be used as evidence of tampering [8].

A different in-camera fingerprint regards the luminance non-linearity, introduced during the acquisition chain in order to improve the perceptual quality of the output digital images; parameters of this non-linearity are dynamically chosen and depend on the camera and the scene, but they are typically constant on the image. The presence of several distinct non-linearities across an image can reveal the non integrity of the content. In [8] it is described how luminance non-linearities introduce specific correlations in the Fourier domain, and how these correlations can be estimated and used for tampering detection.

Finally, another approach proposed in [45] consider that the camera lens often have an optical low-pass property for the purpose of anti-aliasing; hence, when an image is spliced onto another, it is likely that sharp edges are introduced into the tampered content, and that such edge transitions invalidate the low-pass behaviour. Some parameters, representing the optical low-pass property, are extracted by means of statistical methods and are used for image integrity verification.

2.3.2 Out-camera processing identification

A class of forensic algorithms have been proposed for identifying some pro-

cessing applied after image creation, to reveal possible tampering operations. Firstly, for generating convincing digital image forgeries, it is often necessary to resize, rotate, stretch some portions of the manipulated images, thus leading to apply a final resampling step. Although a resampling process does not typically leave perceivable artefacts, it anyway introduces specific periodic correlations between image pixels. For instance, when the image is upsampled, some of the pixel values are directly obtained from the smaller version of the image, and the remaining pixels are interpolated and, thus, they appear highly correlated with its neighbors. The authors in [46] show how to detect a discrete approximation of the applied resampling rate in an image region. The approach relies on the detection of the introduced correlation patterns; since each pattern (based on the probability of each signal sample to be correlated to its neighboring samples) is not in a biunique relation with a resampling rate, the matching could not be uniquely identified. Another method for detecting interpolation has been proposed in [47], where authors observe a periodicity in the variance function of the interpolated signal. Authors in [48] analytically describe the periodic properties of an interpolated signal as well as its derivatives, thus providing also a theoretical support for the methods in [46] and [47]. The method allows the direct estimation of the resampling parameters such as the scaling factors, rotation angles and skewing factors.

Another fundamental processing to be considered is compression. Image tampering usually requires to make use of common photo-editing software: original images, often stored in JPEG format, are manipulated by the editing tools and then they are re-saved using again the JPEG format; hence the resulting tampered images have been wholly or in part, double compressed. While double compression does not necessarily prove malicious tampering, it raises suspicions that the image may be not authentic; as a matter of fact, double JPEG identification has acquired special attention in digital forensic literature, as it may serve as an useful forensics clue. Double JPEG compression often introduces specific correlations between the discrete cosine transform (DCT) coefficients of image blocks that are not present in single compressed images. These correlations can be detected and quantified by analyzing the double quantization effect of two JPEG compressions with different quality factors. Such effect is identified in the exhibition of periodic peaks and valleys in the histograms of the DCT coefficients. Not only the presence of a double compression can be estimated but also the compression

quality that have been used [10, 8] as well as the specific doctored parts [44]. On the other hand, the works in [49] and [50] exploit the JPEG "blockiness" artefacts in order to detect a double compression. The authors in [49] evaluate the Blocking Artefact Characteristic Matrix (BACM) of an image which exhibits a symmetrical shape and regularity for a single JPEG compression; they show how this regularity can be destroyed by a successively non aligned compression. Fan [50] proposes a method to determine whether a non compressed image has been previously JPEG compressed, and further to estimate which quantization matrix has been used. The original intention of such an approach was the removal of JPEG artefacts; however, it can serve as an image forensic tool by also revealing the presence of a double JPEG compression. The method assumes that if there is no compression the pixel differences across blocks should be similar to those within blocks (thus non showing any blockiness artefacts) while they should be different due to block artefacts if the image has been compressed. Finally, in [51] it is also found that the distribution of the first digit of the JPEG DCT coefficients can be used to distinguish a singly JPEG compressed image from a double compressed one. A single compressed image is characterized by a distribution of its DCT coefficients that follows the Benford's law distribution; whereas, as soon as another compression is applied, the coefficients do not follow this law anymore.

One of the main common image tampering is splicing. It is defined as a simple joining of portions coming from two or more different images. In [52] some image features, particularly sensitive to splicing operations, have been extracted and used for designing a classifier. A different technique for detecting splicing searches for the presence of abrupt discontinuities in the image [53]. Several other techniques estimate the camera response function from different regions of an image to detect splicing and possibly other manipulations [42, 8]. The authors in [54] observe that the spliced image may be characterized by a number of sharp transitions such as lines, edges and corners; hence, they found a parameter as a sensitive measure of these sharp transitions, and used it for splicing detection.

Another common tampering is object removal: an image region containing objects that have to be erased, is replaced by another region of the same image. This type of operation is called copy-move or region-duplication. Since there is similar information (e.g. texture, noise and color) inside the same image, it is hard to identify these forgeries via visual inspection. Furthermore,

several post-processing (such as adding noise, blurring, lossy compression) may be performed on such tampered images, thus making the detection of forgery significantly harder. Works in [55, 49, 56] are all based on block matching: firstly, the image is divided into small blocks and some features are extracted for each block; then, by comparing such features for different blocks, it is possible to identify duplicated regions.
Several works in the tampering detection literature try to define the properties of a manipulated image in terms of the distortions it goes through, and using such analysis to present methods for detecting manipulated images. In doing so, some works assume that creating a tampered image involves a series of processing operations; they propose identifying such manipulations by extracting certain salient features that would help distinguish such tampering from authentic data. Image manipulations, such as contrast changes, gamma correction, and other image nonlinearities have been modeled and used to identify them [57]. More generally, in [58], image operations, such as resampling, JPEG compression, and adding of noise, are modeled as linear operators and estimated by linear image deconvolution. In the frequency domain a "natural" signal has weak higher-order statistical correlations. The authors in [59] observed that "un-natural" correlations are introduced if this signal is passed through a non-linearity (which would almost surely occur in the creation of a forgery).

2.3.3 Scene characteristic inconsistencies

Some works have proposed to use as fingerprints the *light properties* directly derived from the scene. In particular, Johnson and Farid base their works on the idea that splicing together different images (that are the acquisition of different scenes) means likely to create a new content where light inconsistencies are present.
In [60, 61] the authors consider to estimate the direction of the light source, both in a simplified case [60] and in complex lighting environments [61]: if the image is supposed to be a composition of more images, hence the lighting direction is computed more than once in different positions of the image; by comparing such directions it is possible to verify whether inconsistencies are present thus revealing the suffered digital tampering.
Lighting direction can be also estimated by considering that the light source produces specular highlights on the eyes of people present in the scene. Au-

thors in [62] propose to compute the direction of a light source by analyzing the different highlights within an image, and by detecting inconsistencies in lighting they are able to reveal possible tampering in some part of the content. Furthermore authors evidence how it would be possible to measure from highlights also the shape and the color of the light source (besides its location), and how these parameters could help in exposing digital forgeries. By considering specific images where eyes are present , in [63] it is shown how to estimate the camera's principal point (i.e. the projection of the camera center onto the image plane) from the analysis of person's eyes within an image. Such a principal point depends on intrinsic and extrinsic camera parameters and it is proposed to be adopted as a fingerprint, whose inconsistency across an image can be used as evidence of tampering.

2.4 Counter-Forensics

The question of trustworthiness of digital image forensic arises because most pubblications still lack rigorous discussions of robustness against counterfeiters. Forensic methods might benefit from research on countermeasures in a similar way as reasoning about attacks in multimedia security in general is useful to improve security. In this sense attacks on image forensic algorithms can be understood as schemes to sistematically mislead the detection methods. In general, such attacks can be assigned to one of the following three objectives: the camouflage of malicious post-processing or tampering of an image, the suppression of correct image origin identification and furthermore the forgery of image origin. Only few papers are presented on this topic, because research on this theme is only in its infancy. In [64, 65, 66] the authors described how to deceive two very important and useful algorithm the resampling detector proposed by Popescu and Farid [46] and the digital camera identification method by Fridrich and Goljan [38]. Another interesting work is presented by Fridrich et al. [67] where a methods is developed to reveal counter-forensic activities in which an attacker estimates the camera fingerprint from a set of images and pastes it onto an image from a different camera with the intent to introduce a false alarm and frame an innocent victim.

Chapter 3

Distinguishing between camera and scanned images

Distinguishing the kind of sensor which has acquired a digital image could be crucial in many scenarios where digital forensic techniques are called to give answers. In this chapter a new methodology which permits to determine if a digital photo has been taken by a camera or has been scanned by a scanner is proposed. Such a technique exploits the specific geometrical features of the sensor pattern noise introduced by the sensor in both cases and by resorting to a frequency analysis can infer if a periodicity is present and consequently which is the origin of the digital content. Experimental results are presented to support the theoretical framework.[1]

The chapter lay-out is the following: Section 3.1 introduces a characterization of the sensor pattern noise and the periodicity is discussed, in Section 3.2 the proposed methodology is presented and in Section 3.3 describes thresholds selection based on ROC curves. In Section 3.4 some experimental results are brought to support theoretical thesis.

3.1 Sensor pattern noise characterization

PRNU (Photo Response Non-Uniformity) noise is quite well-known as being an effective instrument for sensor identification because it is deterministically

[1] This work has been published in the International Journal of Digital Crime and Forensics (IJDCF), Volume 2, Number 1, Jan-Mar 2010 and also has obtained the best paper award in the International Conference e-Forensics, Adelaide, South Australia, 2009.

generated over each digital image it acquires. Such a noise is therefore an intrinsic characteristic of that specific sensor. The extraction of this noise is usually accomplished by denoising filters [1] and information it contains are used to assess something on the sensor characteristics. If we focus our attention on the acquisition process, it is easy to comprehend that when a photo is taken by a digital camera, basically a PRNU with a bi-dimensional structure is superimposed to it; on the contrary, when a digital image is created by means of a scanning operation the sensor array which slides over the to-be-acquired asset located on the scanner plate leaves its mono-dimensional fingerprint row by row during scanning. So in the last case, it is expected that a certain periodicity of the 1-D noise signal is evidenced along the scanning direction. This behavior should be absent in the camera case and this difference can be investigated to discern between images coming from the two different kinds of device. Being $R(i,j)$ with $1 \leq i \leq N$ and $1 \leq j \leq M$, the noise extracted by the scanned image of size $N \times M$, and assuming i (row) as scanning direction, it can, at least ideally, be expected that all the rows are equal (see Equation 7.2).

$$R(i,j) = R(k,j) \quad \forall \ 1 \leq j \leq M, 1 \leq i,k \leq N \tag{3.1}$$

So if a 1-D signal, $\mathbf{S}$ of $N \times M$ samples, is constructed by concatenating all the rows, it happens that $\mathbf{S}$ is a periodical signal of period M (Equation 7.3).

$$\mathbf{S} = [R(1,1), \cdots, R(1,M), \cdots, R(N,1), \cdots, R(N,M)] \tag{3.2}$$

It is also worthy to point out that if the 1-D signal is mounted along columns direction (i.e. this would be right assuming that j is the scanning direction), $\mathbf{S}$ is not periodical anymore, but it is constituted by diverse constant steps each of length M. A periodical signal such as $\mathbf{S}$, represented in Equation 7.3, contains a number of repetitions equal to N and therefore will have basically a frequency spectrum made by equispaced spikes. Such spikes will be spaced of $(N \times M)/M = N$ and will be weighted by the spectrum of the basic replica of the signal. So most of the energy of such a signal is located in these spikes. Obviously this is what should happen, in practice the 1-D signal will be corrupted and its periodical structure altered. Consequently the spectral spikes will be reduced and their magnitude partially spread over the other frequencies. If it is still possible to individuate such

peaks, it will be simple to distinguish between a scanned image and a digital photo.

3.2 The proposed methodology

According to the idea presented in Section 3.1, let us describe in detail which is the proposed methodology to achieve that aim. The to-be-checked image I (size $N \times M$) is denoise filtered [1] obtaining I_d which is subtracted to the initial image to extract the sensor pattern noise R (see Equation 3.3).

$$R = I - I_d \tag{3.3}$$

To improve the possible presence of the deterministic contribution due to the 1-D PRNU pattern noise, R is divided into non-overlapping stripes (both horizontally and vertically, because both possible scanning directions have to be taken into account) and then all the different rows (columns) belonging to a stripe are averaged according to Equation 3.4 where L is the width of the stripe.

$$R_r(k) = \frac{1}{L}\sum_{i=1}^{L} R[i + (k-1)L] \quad 1 \leq k \leq N/L \tag{3.4}$$

After that two new noise images, named *bar codes*, respectively R_r (size $N/L \times M$) and R_c (size $N \times M/L$), have been obtained; R_r and R_c have the same number of samples. If an image has been scanned in the row direction, for instance, it is expected that R_r will be composed by equal (ideally) rows, on the other side such a characterization can not be expected in the column direction for R_c and, above all, for an image coming from a digital camera (both directions): this circumstance is presented in Figure 5.1. *Bar codes* are then used to create the mono-dimensional signal by concatenating respectively rows of R_r and columns of R_c and then periodicity is checked. Sometimes to reduce randomness a low pass filtering operation (usually a median filter) is applied to bar codes, along the rows and the columns separately, before constructing 1-D signals.

For the sake of clarity, let us call S_r and S_c the two mono-dimensional signal, obtained as previously described, from R_r and R_c respectively. DFT (Discrete Fourier Transform) is applied to both these signals and the magnitude of the coefficients is considered. After that a selection is carried out

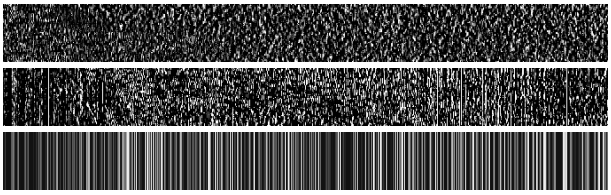


Figure 3.1: Bar codes of size $N/L \times M$ (scanning direction = row): camera image (top), scanned image (center) and ideal bar code for a scanned image (bottom).

on the basis of the following criterion: amplitude values above a threshold T_1 (see Equation 3.5 where α is a weighting factor usually set to 0.4) and at the same time located in the expected positions within the spectrum (see Section 3.1) are taken.

$$T_1 = \alpha * max(max(abs(DFT(S_r))), max(abs(DFT(S_c)))) \tag{3.5}$$

In the end all the values satisfying the previous selection criterion are added, separately for row and column cases, yielding to two energy factors, F_r and F_c respectively and their ratio $RATIO = F_r/F_c$ is computed. If the digital image has been scanned in the row direction, a high value of $RATIO$ is expected (if the scanning direction has been along columns $RATIO$ will be very small), otherwise if the image has been taken by a digital camera the two energy factors should be comparable and a value of $RATIO$ around one is foreseen. Doing so it is possible not only distinguishing between images coming from a scanner or from a camera but, in the scanner case, determining the scanning direction. To improve robustness, this technique is applied to all the three image channels (R, G, B) and three energy contributions are collected in each factor F_r and F_c.

3.3 Analysis of thresholds through ROC curves

As seen in Section 3, the threshold T_1, that is used to evaluate energy of DFT of signals S_r and S_c, depends upon α parameter, besides there is another threshold T_2, for the $RATIO$ value, that makes possible to distinguish between images taken from scanners or digital cameras. Proper choice of these two parameters is a key problem to adequately control discrimination. To find optimal value for T_1 and T_2, is possible to use ROC (Receive Operat-

ing Characteristic or Relative Operating Characteristic) Curve. To introduce ROC Curve is necessary to define two new parameters:

1. *Se* (Sensitivity): the fraction of images taken from a scanner correctly identified as such.

2. *Sp* (Specificity): the fraction of images taken from a digital camera that are correctly identified as such.

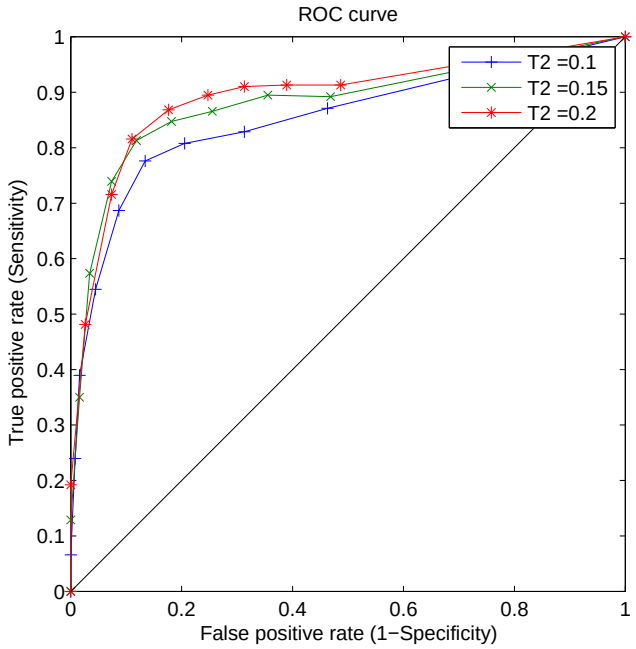


Figure 3.2: *Roc Curves.*

Finding optimal thresholds is not limited to the statistical minimization of wrong classification, but it is also related to the minimization of the FRR (False Rejection Rate) for scanner images or digital camera images. ROC Curve permits to analyze more values of the thresholds to determine which obtains the best results. The ROC Curves analysis is performed through the function that binds the probability of True Positive to recognize scanned images (Se) and the probability to obtain a False Positive ($1 - Sp$). The relationship between these parameters can be represented by plotting Se on the y-axis and ($1 - Sp$) on the x-axis (see Figure 3.2). A single confusion

	Camera	Scanner
Camera	S_p	$1 - S_p$
Scanner	$1 - S_e$	S_e

Table 3.1: *Confusion matrix.*

matrix (see Table **??**) thus produces a single point in ROC space. A ROC curve is formed from a sequence of such points, including (0,0) and (1,1).

To determine the best value of α for T_1 is necessary to plot multiple ROC curves for a certain range of T_2. To get results for ROC curves a training-set composed by 380 images taken from different scanners and 380 images taken from different digital cameras, diverse from the images of test-set used in Section 5 for the experimental tests, have been provided. The training-set has been tested by selecting for T_2 three values $(0.1, 0.15, 0.2)$ and for each of this thresholds, the parameter α ranges in $[0.1, 0.9]$ with steps of 0.1. This determines the ROC curves in Figure 2.

The area under a ROC curve (AUC) quantifies the overall ability of the test to discriminate between scanner and digital camera images. A truly useless test (no better to identify true positive than flipping a coin) has a relative area of 0.5. A perfect test (one that has zero false positive and zero false negative) has a relative area of 1. Real tests will present after that an area between these two values. As it can be noticed, the greater AUC is obtained with T_2 equal to 0.2

Next step is to analyze the single ROC Curve (see Figure 3.3). A point in ROC space dominates another one if it has a higher true positive rate and a lower false positive rate. So the best value for α is the closest point to (0,1); in this case, it is achieved for α equal to 0.4.

Finally on the basis of such an analysis, in the experimental tests, the values of parameters have been set to $\alpha = 0.4$ and $T_2 = 0.2$ respectively.

3.4 Experimental results

Experimental tests have been carried out to support the theoretical framework. Digital images coming from 4 different scanners (Epson Expression XL 10000 2400x4200 dpi, HP Scanjet 8300 4800x4800 dpi, HP Deskjet F4180 1200x2400 dpi, Brother DCP 7010 600x2.400 dpi) and from 7 commercial cameras (Canon DIGITAL IXUS i ZOOM, Nikon COOLPIX L12, Fuji

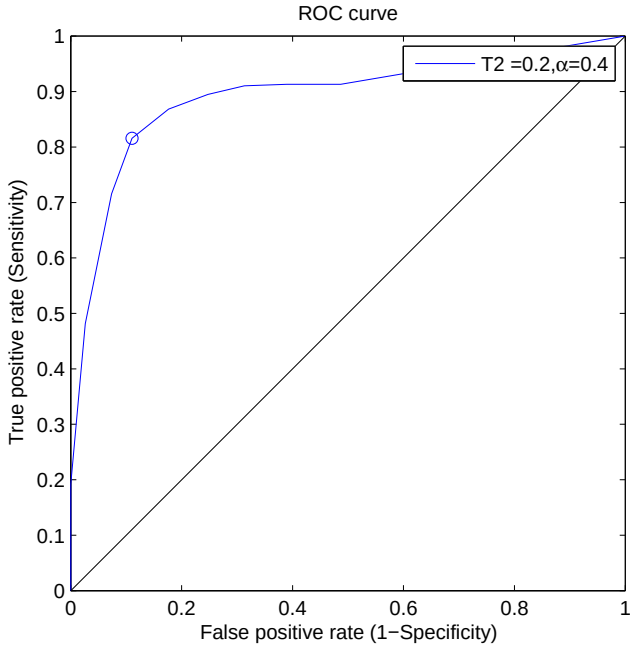
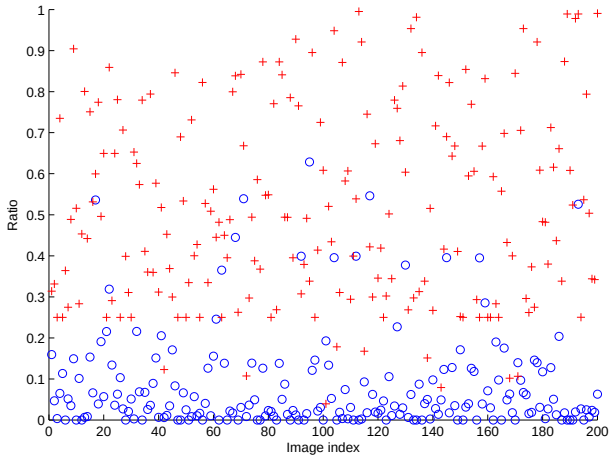
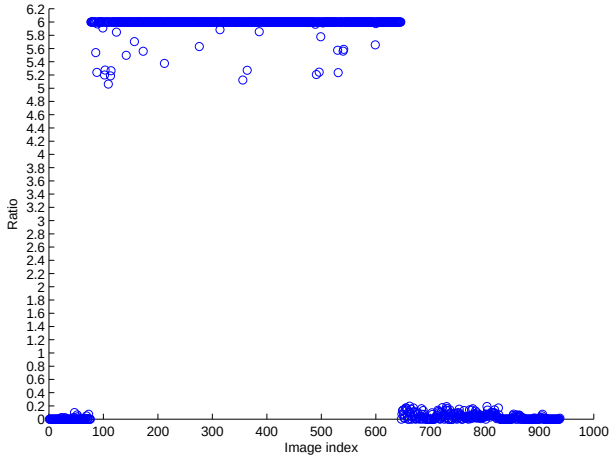


Figure 3.3: *The selected ROC Curve.*

Finepix F10, HP Photosmart C935, Nikon D80, Samsung VP-MS11, Sony DSC-P200) have been acquired in TIFF and JPEG format. Because of the diverse size of the contents, the analysis have been done by dividing them into images of fixed dimension $N \times M$ (1024×768). Obtained results have confirmed theoretical assumptions as it can be seen in Figure 5.2 (a) where $RATIO$ values are plotted and a separate clustering is observed (for sake of clarity when $RATIO$ was over 1 the inverse was taken, due to this, information about scanning direction is lost). In Figure 5.2 (b), only scanned images, correctly detected, are figured: in this case inversion of $RATIO$ has not been done and, to make visualization easier, high values are saturated at 6. It is simply to distinguish the two different scanning directions individuated by high and low values of $RATIO$; in particular it is interesting to note the left and the right side of the plot related to column scanning direction and the central part related to row direction. In Figure 5.3 the statistical distribution of $RATIO$ for 1000 camera images (a) and 1000 scanned ones (b) are pictured where, in this case, higher values have been saturated at 50; a strong concentration is evidenced on the tails of the graph for the scanner case. Finally, a massive test has been carried out on a data set of



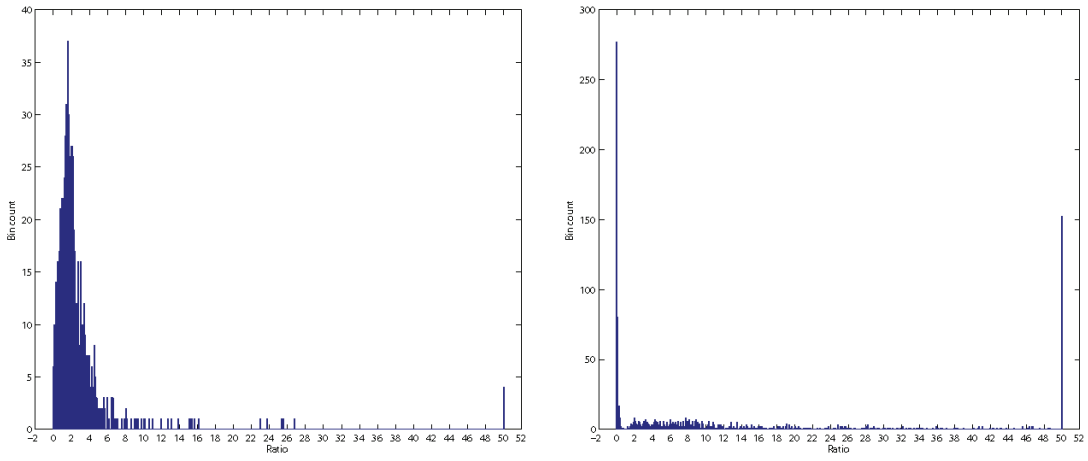
(a)



(b)

Figure 3.4: Energy $RATIO$ for 200 scanned (circle) and 200 camera (cross) images (a). Energy $RATIO$ only for 950 scanned images, correctly detected: scanning directions are evidenced (b).

2000 images (half scanned images and half photos) by setting a threshold at 0.2 with $RATIO$ normalized between 0 and 1 (as done for Figure 5.2 (a)): percentages are presented in the rows of Table 3.2 (left). In Table 3.2 (right) percentages related to the scanning directions in the scanner successful cases (85.35% of Table 3.2 left) are reported.



(a) (b)

Figure 3.5: Statistical distribution of $RATIO$: camera (a) and scanned images (b).

	Camera	Scanner
Camera	89.74%	10.26%
Scanner	14.65%	85.35%

	Row	Column
Row	100.00%	0.00%
Column	0.00%	100.00%

Table 3.2: *Confusion matrix for scanned and camera images over a data set of* 2000 *images (left) and scanning direction recovery for scanner correct answers (right).*

3.4.1 Detecting cut-and-paste forgeries

In this subsection results concerning authenticity verification against cut-and-paste tampering are presented.

Figure 3.6: *(Left) Original scanned and (Right) forged image.*

The cut-and-paste tampering method (also known as splicing) consists

of joining image parts, which come from different images, that might be captured by using different devices. In the case when the tampered images were created by joining a digital camera picture with a scanned image, or viceversa, the proposed algorithm can also be used to identify such forgeries. In addition to this it is also possible to identify another type of forgery attack, such as joining parts of two digital scanner images with different scanning direction. An example is shown in Figure 3.6, where it has been created a tampered picture by inserting a part of a digital camera image coming from a Nikon E4600 within an original scanned image (HP Deskjet F4180, 600 dpi) of size 2380×3550.

To identify the tampered region of the picture, the image is controlled, according to the proposed technique, by dividing it into investigation blocks of size 384×512, starting from the top-left corner. 36 sub-images are obtained which almost covers the whole image (a second check could be done by starting from the bottom-right corner) and the factor $RATIO$ in each block is calculated. Since the original image has been scanned along column direction, it is expected that the Energy $RATIO$ of each block will be very small (almost zero). Whereas a $RATIO$ with values over 0.2, indicating that some parts come from a camera, are foreseen in the tampered region ($RATIO$ inversion is performed as explained in sub-section 3.4). In Figure 3.7 (a) and (b), the $RATIO$ values of each block for the original image and for the tampered one are shown respectively: each bar of the 3D graph corresponds at one of the 36 blocks composing the image. Blocks with higher values than the decision threshold T_2 set to 0.2 are evidenced as tampered blocks in the modified area. Plenty of experiments were performed, creating different tampering images with different sizes of the forged areas: the technique is able to identify a forged patch that involves the 7% (on average) of the original image. Besides, the percentage ratio between the tampered area and the block size, used for investigation, should be around 40%.

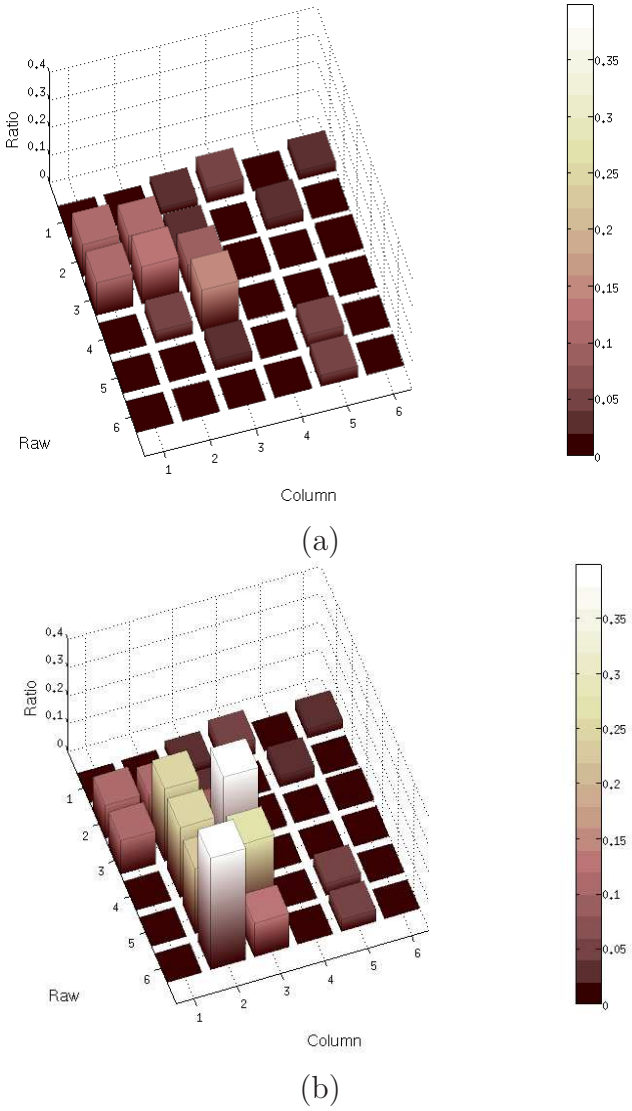


Figure 3.7: *The RATIO values of the original image (a) and of the tampered one (b), shown in the Figure 3.6.*

Chapter 4

Analysis of denoising filters in source camera identification

Identification of the source that has generated a digital content is considered one of the main open issues in multimedia forensics community. The extraction of photo-response non-uniformity (PRNU) noise has been so far indicated as a mean to identify sensor fingerprint. Such a fingerprint can be estimated from multiple images taken by the same camera by means of a denoising filtering

In this chapter a theoretical and experimental comparative analysis of different wavelet denoising filters to estimate the PRNU in order to solve the digital camera identification problem is presented. Two denoising filters are used operating in the wavelet domain and based on different noise models. The first is the filter proposed in [1] and used in [11] and the second filter is a MMSE filter operating in the undecimated wavelet domain [2]. Introducing this kind of filter we make an assumption that the digital camera noise is considered as dependent on the sensed signal, while using the filter described in [1] a signal-independent noise model is supposed.
The filter in [2] is used for the first time in the digital forensic domain to solve the problem of source camera identification, generally it is adopted for speckle and film-grain noise removal in coherent radiation imaging systems including ultrasound, infrared and laser imaging and synthetic aperture radar (SAR).
The paper layout is the following: in Section 4.1 the two denoising filters are introduced, in Section 4.2 we describe the digital camera sensor output model that will be used to derive the estimation of PRNU and the noise

models for the two filters will be discussed. Some experimental results are presented to evaluate the denoising filters performances in Section 4.3.[1]

4.1 Denoising Filters

According to PRNU methodology, it is crucial to analyze the type of denoising filter to be used for the extraction of such a noise. In this work we have decided to evaluate two denoising filters described in detail hereafter: a spatially adaptive statistical modelling of wavelet coefficients filter [1] (Mihcak's Filter) and a MMSE filter operating in the undecimated wavelet domain [2] (Argenti's Filter). The first one adopts a simple additive noise model, on the contrary the second one is based on a signal dependent noise model.

For sake of completeness a simple low-pass filter in the wavelets domain (LP Filter) has been considered too, to provide a performance lower bound during the experimental tests. In this case, after a 4 level Discrete Wavelets Transform (DWT), all the detail coefficients are set to zero and the Inverse Discrete Wavelets Transform (IDWT) is performed to reconstruct the denoised image. The extreme simplicity of this filter is inversely proportional to its accuracy, because setting to zero the coefficients of detail equally removes noise and details that are part of the content of the image. Therefore, the results obtained when we used this filter are presumably coarser.

4.1.1 Mihcak's Filter [1]

This filter is based on a spatially adaptive statistical modelling of wavelet coefficients; such noisy coefficients $G(k)$ are considered as the addition of the noise-free image $X(k)$ (a locally stationary i.i.d. signal with zero mean) and the noise component $n(k)$ (a stationary white Gaussian noise with known variance σ_n^2). The target is to retrieve the original image coefficients as well as possible from the noisy observation. By using a local Wiener filter (Equation (4.1)) we obtain an estimate of the denoised image in the wavelet domain and then apply the IDWT (Inverse DWT).

$$\hat{X}(k) = \frac{\sigma_x^2(k)}{\sigma_x^2(k) + \sigma_n^2} G(k) \tag{4.1}$$

[1] This work has been published in International Journal of Digital Crime and Forensics (IJDCF), Volume 2, Number 2.

However, we can not use the true signal variance $\sigma_x^2(k)$ since it is unknown, but only an estimate $\hat{\sigma}_x^2(k)$ achieved by previously using a MAP (Maximum A-posteriori Probability) approach on noisy wavelet coefficients.

4.1.2 Argenti's Filter [2]

Unlike the filter seen before this filter is based on a signal-dependent noise model (see Equation 4.2):

$$\mathbf{I} = \mathbf{I}_o + [\mathbf{I}_o]^\alpha \cdot \mathbf{U} + \mathbf{W}, \tag{4.2}$$

where $\mathbf{I}$ and $\mathbf{I}_o$ represent the noisy and noise-free images respectively, while $\mathbf{U}$ states for a stationary zero-mean uncorrelated random process independent of $\mathbf{I}_o$ and $\mathbf{W}$ takes into account of electronics noise (zero-mean white and gaussian). The term α is the exponent that rules the dependence of noise from the signal. It is a parametric model which meets different situations of acquisition [68]. The parameters to be estimated are: α, σ_U^2 which is the variance of $\mathbf{U}$ and σ_W^2 which is the variance of electronic noise $\mathbf{W}$, that can simply be estimated from black image area. The denoising method is based on MMSE filtering in undecimated wavelet domain: after the estimation of the parameters α and σ_U^2 in the spatial domain, the undecimated wavelet transform of the image is computed and then a MMSE filtering in this domain is applied according to the supplied parameters. IDWT to reconstruct the estimated noise-free image is finally performed.

The estimation of α and σ_U

As described above two are the parameters to be estimated in the noise model (Equation (4.2)): α and σ_U^2. In [69] has been proposed an iterative algorithm to estimate these parameters which utilizes an adaptive filter (a MMSE noise filter in the spatial domain). After simple calculation [69], it is possible to derive the relationship among $\tilde{\sigma}_I$, the image $\mathbf{I}$ and σ_U expressed in Equation (4.3) which is valid on homogeneous pixels:

$$\log[\tilde{\sigma}_\mathbf{I}] = \alpha \cdot \log\{E[\mathbf{I}]\} + \log(\sigma_u). \tag{4.3}$$

So on homogeneous pixels, the ensemble statistics of $\mathbf{I}$ are aligned along a straight line having α as a slope and $\log(\sigma_U)$ as intercept. At each step of the algorithm, the α and σ_U estimate are substituted in the MMSE spatial filter

in order to obtain the noise free image on which the homogeneous pixels are selected through an homogeneity equation described in detail in [69]. On these homogeneous pixels a log scatter plot is computed, the regression line is estimated and then the α and σ_U are found.

4.2 Digital Camera Sensor Output Model

Digital camera acquisition process is well-known as being composed by different processes such as signal quantization, white balance, color and gamma correction, filtering and usually JPEG compression. This variety of effects, together with the diversities due to the specific kind of camera, determine that a precise modelling is difficult to be achieved. In [11] a quite complete model, which takes into account most of the components relevant for forensic task, is introduced. Such a model is reported in Equation (4.4), where $\mathbf{I}$ is the 2-D sensor output (noisy image), g and γ are the gain factor and the gamma correction respectively, and $\mathbf{Y}$ is the 2-D incident light:

$$\mathbf{I} = g^{\gamma} \cdot [(1 + \mathbf{K})\mathbf{Y} + \mathbf{\Lambda}]^{\gamma} + \mathbf{\Theta_q}. \tag{4.4}$$

The term that is useful for the forensic analysis is $\mathbf{K}$ which represents a zero-mean noise-like signal that is the PRNU (Photo Response Non-Uniformity) (i.e. the 2-D sensor fingerprint deterministically superimposed to each taken digital image), while $\mathbf{\Theta_q}$ is the quantization noise and $\mathbf{\Lambda}$ takes into account a combination of different noise sources.

According to the discussion presented in [11], this expression can be simplified to get to a more concise representation (see Equation (4.5)), where $\mathbf{I}_o$ is the noise-free sensor output, $\mathbf{K}_1 = \mathbf{K} \cdot \gamma$ is basically considered again as the PRNU and $\mathbf{\Theta}$ is an ensemble of independent random noise components.

$$\mathbf{I} = \mathbf{I}_o + \mathbf{I}_o \cdot \mathbf{K}_1 + \mathbf{\Theta} \tag{4.5}$$

This expression points out an additive-multiplicative relation between the signal without noise and the noise terms. An estimate $\hat{\mathbf{I}}_o = F_M(\mathbf{I})$ of the denoised image $\mathbf{I}_o$ is usually obtained by a wavelet-based denoising filter F_M [1], though such a filter is built on an additive noise model as explained in Section 4.1.1. It is immediate to comprehend that Equation (4.2) coincides with Equation (4.5) ($\mathbf{U}$ and $\mathbf{W}$ are the same of $\mathbf{K_1}$ and of $\mathbf{\Theta}$ respectively) except for the term α ($|\alpha| \leq 1$) which determines signal-dependency. When α

Filter Type	n.	Nikon E4600	Samsung MS11	Olympus FE120	Sony S650	Nikon L12	**Concord 2000**
	30	-1.714	0.735	-0.234	0.778	0.262	**67.969**
	31	0.083	-0.160	0.469	-0.056	-0.265	**83.186**
	32	-1.007	0.593	-0.254	0.090	0.147	**67.926**
	33	-0.722	-0.522	0.411	-0.158	-0.456	**39.619**
Low Pass	34	-1.815	0.700	0.322	0.883	1.037	**43.593**
	35	0.613	-1.261	-0.028	-0.340	-0.444	**68.18**
	36	-0.280	0.292	-0.539	0.294	-0.229	**69.173**
	37	0.477	0.016	0.347	-0.082	0.341	**99.602**
	38	0.416	-0.013	-0.001	-0.239	0.481	**63.028**
	30	1.210	-0.487	0.365	0.173	-1.997	**101.070**
	31	-0.370	-1.152	0.263	-0.880	-1.157	**98.416**
	32	0.190	0.923	0.171	0.619	0.043	**100.710**
	33	-1.486	1.226	-0.524	0.595	0.026	**74.502**
Mihcak	34	1.154	-0.621	0.031	1.368	0.449	**70.787**
	35	0.288	-0.594	0.917	-0.645	0.440	**105.400**
	36	0.166	0.470	-0.736	0.001	-0.064	**102.320**
	37	0.219	0.946	-0.048	0.185	0.736	**145.380**
	38	0.525	0.948	-0.282	0.679	0.996	**92.319**
	30	0.884	-0.469	0.026	0.334	-0.471	**111.530**
	31	-3.362	-4.128	3.466	-1.883	-1.879	**111.290**
	32	0.046	1.355	-1.608	1.026	0.787	**102.050**
	33	-0.591	-0.238	-0.547	-0.162	-0.959	**84.691**
Argenti	34	1.292	-0.762	-0.549	-1.179	-0.720	**79.884**
	35	0.174	-0.423	0.252	-0.421	-0.577	**113.380**
	36	-0.046	-1.253	-0.212	-1.235	-0.060	**105.320**
	37	1.291	0.051	-0.839	1.217	-0.629	**143.020**
	38	1.556	0.216	-1.395	0.889	1.211	**96.836**

Table 4.1: *Correlation values (values are to be scaled by* 10^{-3}*) for a selection of* 9 *test images (*30 *to* 38*) from a Concord 2000 digital camera calculated with the fingerprints of* 6 *cameras (Concord 2000 included).*

is equal to 1 for purely multiplicative noise the two models are identical. On the basis of this consideration, it is interesting to analyze how this difference in modelling can influence filtering and consequently PRNU detection.

The two digital filters F_M and F_A will yield two estimates $F_M(\mathbf{I})$ and $F_A(\mathbf{I})$, and when are tested against signal-dependent generated noisy images, results achieved in denoising operation are generally superior with F_A filter (e.g. 2 or 3 dB of PSNR improvement), as expected. This witnesses the goodness of the Argenti's filter when the noise model is exactly matched.
When the noise-free image is obtained, the PRNU noise is computed, at least in a rough approach, by subtracting from the noisy image the denoised one. The more accurate the denoised image estimate, the more reliable the fingerprint extraction so high relevence is given to the kind denoising filter used. The sensor fingerprint $\mathbf{N}$ is obtained, as indicated in Equation (4.6),

by suppressing the scene content:

$$\mathbf{N} = \mathbf{I} - \hat{\mathbf{I}}_o. \tag{4.6}$$

Successively a refinement of the fingerprint is carried out by averaging the results got over a set of M training images (usually M is around 50). This operation yields to delete different noise components that are present on the acquired images but which are not systematic like PRNU.

	LP		Mihcak		Argenti	
Camera	t (10^{-3})	FRR	t (10^{-3})	FRR	t (10^{-3})	FRR
Nikon E4600	3.0	$3x10^{-2}$	3.0	$8.11x10^{-3}$	9.3	$8.11x10^{-3}$
Samsung MS11	15.5	$2x10^{-2}$	4.6	$1.8x10^{-10}$	9.9	$8x10^{-12}$
Olympus FE120	4.2	$2.8x10^{-2}$	2.6	$1.2x10^{-2}$	9.9	$8x10^{-4}$
Sony S650	4.9	$2.6x10^{-2}$	2.0	$3.1x10^{-3}$	7.7	$1.8x10^{-2}$
Nikon L12	5.6	1.1810^{-1}	4.1	$8.8x10^{-3}$	8.4	$9.4x10^{-3}$
Canon DI50	5.7	5.210^{-1}	4.2	$4.5x10^{-2}$	7.7	$4.7x10^{-2}$
Nikon D40x	2.1	1.7610^{-1}	2.4	$7x10^{-3}$	4.8	$1.5x10^{-2}$
Canon Diiz	7.7	2.7210^{-1}	4.5	$9.3x10^{-2}$	5.2	$5.7x10^{-2}$
HP PSC935	4.6	4.510^{-1}	4.1	$1.9x10^{-10}$	5.0	$7x10^{-2}$
Concord 2000	3.3	$1.3x10^{-2}$	3.7	$5x10^{-4}$	5.8	$9x10^{-4}$

Table 4.2: *Thresholds t and FRR for all 10 cameras with a FAR=10^{-3} for the three different denoising filters.*

4.3 Experimental results

In the first part of this section the denoising filters performances are discussed in relation with the digital camera identification. In the second part of this section experimental measures of the model parameters associated to the Argenti's filter are reported and analyzed.

4.3.1 Denoising filters performances

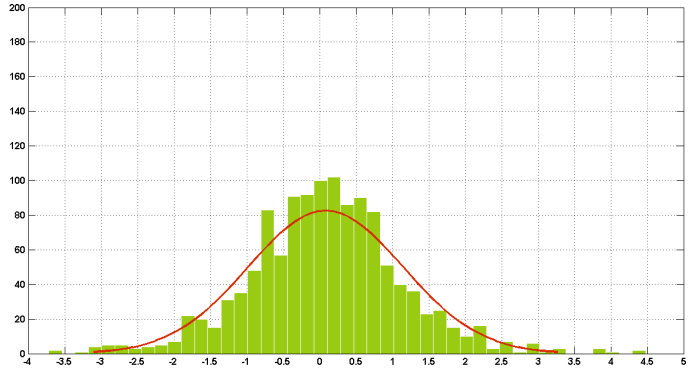
In this section experimental results for digital camera identification, carried out to compare the three filters (LP, Mihcak and Argenti) used to estimate the PRNU noise are collected and analyzed. The data set is composed by images coming from 10 digital cameras of various brand and model taken by generic users in different kinds of settings. We have created the fingerprint

for each camera in the data set, averaging residual noises from 40 images; the remaining photos have composed the test-set (approximately 250 images for each camera). For each camera we obtained three fingerprints, one for each denoising filter under investigation. The correlation between each fingeprint and the residual noises of the test images is performed.

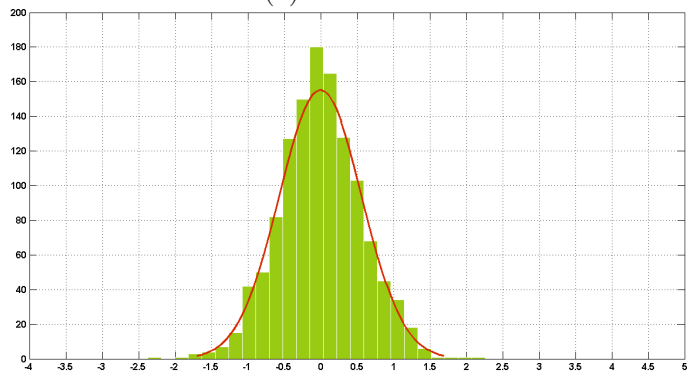
In Table 4.1 a numerical example of the correlation values for a selection of images from a Concord 2000 is shown. Each fingerprint calculated for the Nikon E4600, Samsung MS11 etc., through the three filters under examination (Low Pass, Mihcak and Argenti) is compared with the residual noise of a selection of Concord 2000 test images (from 30 to 38). It is worth to point out that the correlation values in the last column of the Table 4.1 have the higher values, so the images taken by the Concord 2000 are correctly identified as belonging to Concord 2000 digital camera. Moreover it is interesting to observe that higher values of the last column are encountered when the correlation is made between the fingerprint and the PRNU noise residual calculated with the Argenti filter (see the lower part of the Table 4.1).
To decide if an image has been acquired or not by a specific camera we introduced a statistical threshold for the correlation value. To calculate the threshold we used the Neyman-Pearson approach based on two parameters: the False Acceptance Ratio (FAR) and False Rejection Ratio (FRR).
The FAR establishes a limit to the number of cases in which an image is wrongly identified as related to a given fingerprint. The FRR is the rate that indicates the number of images that, though related to the given fingerprint, are not recognized as such. With this method we set an *a priori* FAR and we found the threshold that minimize FRR. We suppose that the distribution of the correlation between the fingerprint of the camera C_0 and the noise residuals coming from images taken by different cameras is Generalized Gaussian (see Equation (4.7)).

$$f(x;\delta,\beta,\mu) = \frac{1}{2\delta\Gamma(1+1/\beta)} e^{-(\frac{|x-\mu|}{\delta})^{\beta}} \tag{4.7}$$

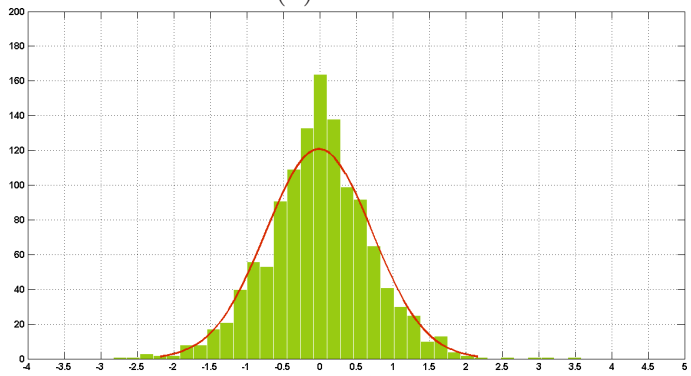
In Figure 4.1 the distribution of correlation between the Nikon D40x with noise residual from a selection of images taken by the others cameras in the database (except the Nikon D40x) is shown. It is possible to fit the data with a Generalized Gaussian distribution centered close to zero. Furthermore, the standard deviation is bigger in the Low Pass filter case and decrease in the other two filters. So it's possible to consider the standard deviation as a



(a) Low Pass



(b) Mihcak



(c) Argenti

Figure 4.1: *Distribution of the correlation values between Nikon D40x fingerprint with residual noises taken by a random selection of* 300 *images belonging to different cameras. The continuous line is the Generalized Gaussian fitting.*

performance marker of the three filter, and it is possible to presume that Argenti's and Mihcak's filter will show better results. The method of mo-

ments [22] is used to estimate the parameters of Equation (4.7) and then we calculate the cumulative density function of $f(x; \delta, \beta, \mu)$ over all the cameras at disposal, except C_0. By using the Neyman-Pearson approach we determine the threshold by minimizing the probability of rejection, given an upper bound on the FAR $= 10^{-3}$. In Table 4.2 the decision thresholds and the FRR computed for each denoising filter relatively to the 10 test cameras are shown.
The LP filter has the worst behavior as obviously expected. The other two filters showed a comparable behavior; in fact in most cases the value of FRR has the same order of magnitude though Argenti's filter has a significative lower FRR for Samsung MS11 and Olympus FE120. However Argenti's filter does not exhibit a considerable improvement in the results of camera identification compared to Mihcak's filter. According to our analysis, this is mainly due to the sensibility of the filter itself to the reliability of the parameters estimation (see Section 4.3.2). In fact we noted, by acting on noisy images generated by introducing a speckle noise, that filter performances drastically decreased, when an uncorrect estimation was done, specifically for the parameter α.
In Figure 5.4 the correlation values for images from a Olympus FE120 with 5 fingerprints of various cameras are pictured. The distributions of the correlation values in all the three cases are always well separated; in fact the higher values are those related to the correlation between the noise residual of the Olympus FE120 images and its fingerprint. In the Mihcak and Argenti filter cases (Figure 5.4 (b),(c)) the two classes are better clustered than in Figure 5.4 (a). This result confirms that using a denoising filter adequate at the noise model there is an improvement in the performance of the camera identification method.

4.3.2 About α and σ_U estimate in the Argenti's filter

The Argenti's filter proposes, as said in Section 4.1.2, an iterative estimate of α and σ_U in the parametric noise model (Equation (4.2)). So some tests to check the reliability of such estimation have been performed. We consider a noise free computer generated image (Figure 4.3), then we corrupted this image with a noise in order to achieve a $SNR = 3dB$, driven by the parameters α and σ_U. Then using the estimation algorithm proposed in 4.1.2 we obtained the $\hat{\alpha}$ and $\hat{\sigma_U}$ estimated values. In Table 4.3 the results of this test

are listed: in the first and the second columns there are the actual α and σ_U values while in the third and the forth there are the corresponding estimated values obtained by implementing the algorithm proposed in [69]. In general the estimate of each couple of value (α, σ_U) seems to be consistent with the real ones.

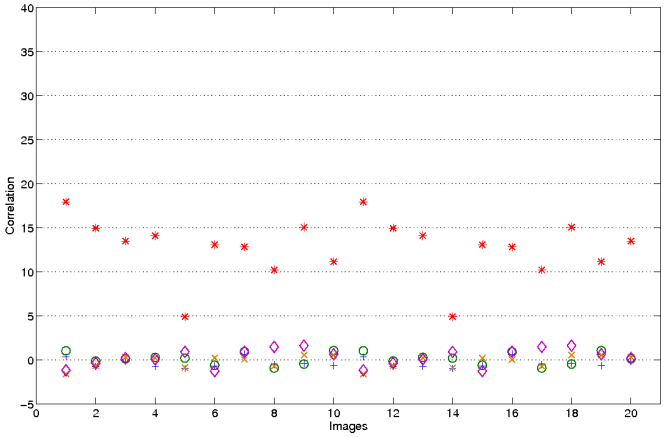
α	σ_U	$\hat{\alpha}$	$\hat{\sigma_U}$
-0.80	1340.66	-0.77	1187.47
-0.70	885.20	-0.66	751.36
-0.60	578.87	-0.55	461.65
-0.50	375.11	-0.45	298.65
-0.40	241.01	-0.35	188.70
-0.30	153.63	-0.25	121.34
-0.20	97.22	-0.16	80.27
-0.10	61.12	-0.08	54.00
0.00	38.19	0.01	36.31
0.10	23.74	0.09	24.70
0.20	14.68	0.17	16.78
0.30	9.04	0.24	11.67
0.40	5.55	0.32	7.84
0.50	3.39	0.40	5.35
0.60	2.07	0.48	3.57
0.70	1.25	0.57	2.36
0.80	0.76	0.65	1.54

Table 4.3: *The real α and σ_U and their estimate $\hat{\alpha}$ and $\hat{\sigma_U}$ over different measures.*

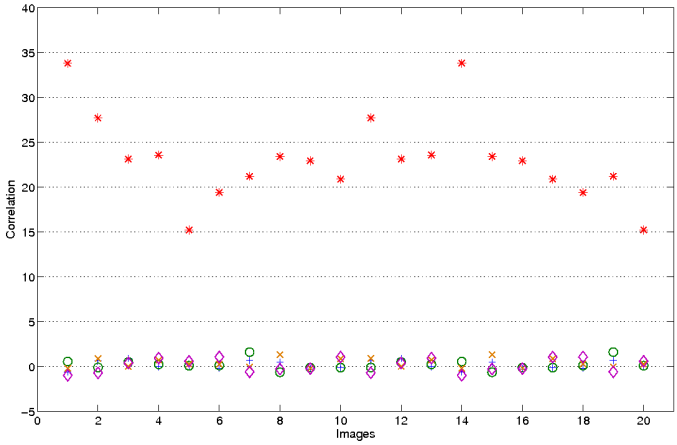
Furthermore we considered the estimate of these parameters in relation to the correlation value obtained from the fingerprint and the residual noise when the Argenti's denoising filter is used. We calculated the first estimate (α^1 and σ_U^1) of the parameters for each photo taken by a certain camera C.

We computed new α and σ_U values calculated in the range of [-50%, +50%] from the initial value (121 values are considered in total). Then

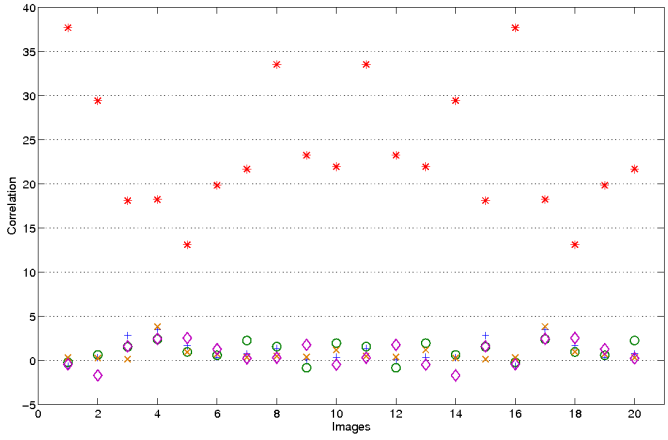
we calculated the residual noises for each of the 121 couples and then the correlation of them with the fingerprint of the camera C is measured. In the majority of the observed cases the correlation value does not improve using the 121 values of α and σ_U instead the initial one. In Figure 4.4 an example of this situation for Nikon E4600 is presented. The values of (α, σ_U) in the (x, y) axes, and in z axes the value of the correlation are reported. The higher value of correlation is in the central point of the graph ($x = 0$, $y = 0$) that corresponds at the initial estimate of the two parameters. According to these observations we used the first estimate of the α and σ_U parameters for the computation of the PRNU noise. So it is necessary to find a new technique to estimate α and σ_U parameters in order to improve their reliability.



(a) Low Pass



(b) Mihcak



(c) Argenti

Figure 4.2: *Correlation values of residual noises (values are to be scaled by* 10^{-3}*) of* 20 *images coming from an Olympus FE120 with* 5 *fingerprints. Legend:* $+$ *Nikon E4600,* $\circ$ *Samsung MS11,* $*$ *Olympus FE120,* $\times$ *Sony S650,* $\diamond$ *Nikon L12*

Figure 4.3: *A computer graphics image "Room".*

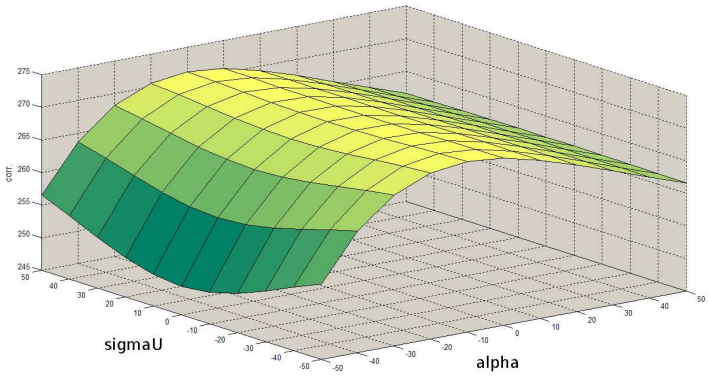


Figure 4.4: *Trend of the correlation values with respect to (α, σ_U) for a Nikon E4600.*

Chapter 5

Fast Image Clustering of Unknown Source Images

Succeeding in determining information about the origin of a digital image is a basic issue of multimedia forensics. It is easy to understand that in many application scenarios information at disposal are very limited; this is the case when, given a set of N images, we want to establish if they belong to M different cameras where M is less or, at most, equal to N, without having any knowledge about the source cameras. In this paper a new technique which aims at blindly clustering a given set of N digital images is presented. Such a technique is based on a pre-existing one [70] and improves it both in terms of error probability and of computational efficiency. The system is able, in an unsupervised and fast manner, to group photos without any initial information about their membership. Sensor pattern noise is extracted by each image as reference and the successive classification is performed by means of a hierarchical clustering procedure. Experimental results have been carried out to verify theoretical expectations and to witness the improvements with respect to the other technique. Tests have also been done in different operative circumstances (e.g. asymmetric distribution of the images within each cluster), obtaining satisfactory results.[1]

The chapter is organized as it follows: Section 5.1 describes the PRNU enhancer, while Section 6.2.2 presents the new clustering procedure; in Section 5.3 experimental results are presented.

[1] This work has been presented in the International Workshop on Information Forensics and Security, Seattle, 2010.

5.1 PRNU Enhancer

Since Photo Response Non-Uniformity (PRNU) is part of the high-frequency components of the image's signal, it can be used the model in Equation (5.1), in order to extract the noise n from an image I:

$$n = I - F(I) \tag{5.1}$$

where F, is a wavelet-based denoising filter [71] that filters out the sensor pattern noise of the image.

However, not all the high-frequency components of an image are responsible for the sensor pattern noise: in fact, scene details also contribute to these components and their magnitude is generally greater than that of PRNU [72, 73]. So the noise n should be cleaned from scene details to improve the system performance. This task becomes extremely critical when we take only a small block of the image into account: on the one part a small block helps by reducing the computational time, on the other part it can lose a lot of information. In the work presented in [72] it has been developed a function that aims to filter out scene details, based on the following idea: scene details contribute to the very strong signal components, so the stronger a signal component (in n), the more it should be attenuated. According to this consideration, a new kind of enhancer has been developed. The noise-enhancing function gives bigger weighting factors to the weak components of n in the DWT domain, and viceversa, and it is described by the formulae in Equation (5.2):

$$n_e = \begin{cases} 0 & n(i,j) < -\alpha \\ -\cos(\frac{n(i,j)\pi}{2\alpha}) & -\alpha \leq n(i,j) \leq 0 \\ \cos(\frac{n(i,j)\pi}{2\alpha}) & 0 < n(i,j) \leq \alpha \\ 0 & n(i,j) > \alpha \end{cases} \tag{5.2}$$

where α is a parameter that represents the cut-value between the PRNU components and the scene details (see Figure 5.1).

The selection of the parameter α has been achieved by means of a set of source camera identification experiments. On a set of 1200 photos, taken from 6 different cameras (i.e. 200 for each), small blocks of 128×128 pixels cropped from the original photos were used to evaluate the performance when varying the parameter α in order to get the best one. Six reference PR-

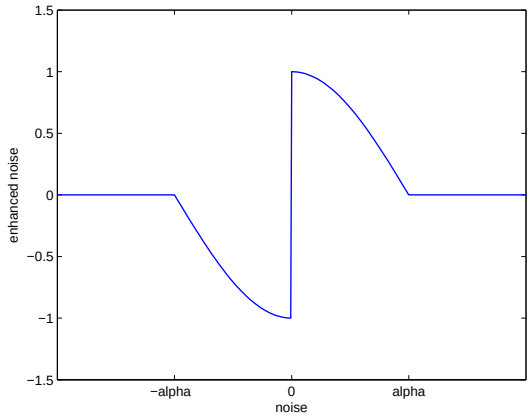


Figure 5.1: Enhancement function

NUs were generated by calculating the average of the noises extracted from 50 photos taken by each digital camera, without applying any enhancing function. Then, these fingerprints were used to classify a test-set (made by 600 photos, 100 from each camera, different from those used for the reference PRNUs) by simply calculating the correlation between the current noise (extracted with the use of the enhancing function) and each of the six reference PRNUs and deeming the image as taken by the camera corresponding to the maximum of the correlation values. The best classification performance was achieved with $\alpha \in [0.05, 0.0575]$ with 504/600 correct classifications, corresponding to an 84% percentage (see Figure 5.2) though the trend is stable starting from $\alpha = 0.04$.

The performance of this test without the use of an enhancing function is only 215/600 (35.8%): that's what it is expected from the previous considerations.

5.2 Fast Unsupervised Clustering

The aim of this method is to quickly classify a generic group of photos taken by different cameras, in a completely unsupervised mode. Starting from the method proposed in [70], it has been attempted to improve performance in terms of computation speed and accuracy estimation. To do this a different clustering algorithm based on *hierarchical clustering* has been introduced [74]. Hierarchical clustering outputs a hierarchy of clusters which may be

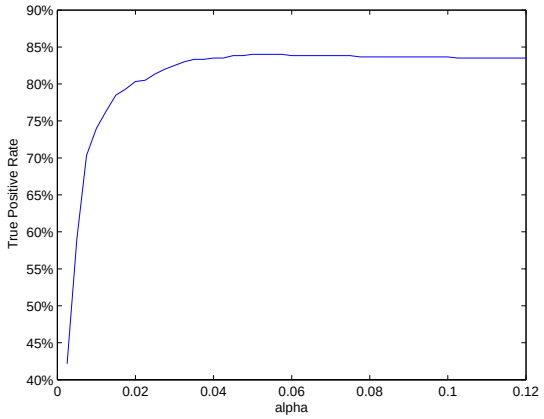


Figure 5.2: TPR vs α factor

represented by a tree-like two-dimensional structure known as dendrogram, which illustrates the fusions (agglomerative clustering or bottom-up) or divisions (divisive clustering or top-down) made at each successive stage of analysis: the root of the dendrogram is a single cluster containing all the elements, and the leaves correspond to the individual elements. Agglomerative hierarchical clustering procedure, that has been used, over a set of N data produces a series of partitions of the data $P_0, \ldots, P_{N-1}$: the first P_0 consists of N single object clusters, while the last P_{N-1} consists of a single group containing all the N elements. So the procedure merge pairs of clusters at each step until all clusters have been merged into a single one that contains all the data; in other words, the actual number of clusters K ranges from N to 1. Any valid metric may be used as a measure of similarity between pairs of elements: since it was facing the problem of clustering a set of images (more precisely, the noises extracted from them and then enhanced), the best choice is the correlation measure. The choice of which clusters to merge at each step is determined by a linkage criterion, which is a function of the pairwise distances (correlations in our case) between the noises. Several experiments have been done over various linkage criterion and it was found that the "average linkage method" is the more appropriate for the issue that has been examined. The average linkage method establishes that the distance (or similarity) between two clusters is the average of all the distances (or similarities) between pairs of elements, taken each from the respective cluster. So the similarity $H(A, B)$ between the two clusters A

and B is calculated according to Equation (5.3):

$$H(A,B) = \frac{1}{\|A\|\,\|B\|} \sum_{n_i \in A, n_j \in B} corr(n_i, n_j) \tag{5.3}$$

where $corr(n_i, n_j)$ is the normalized correlation (Equation (5.4)):

$$corr(n_i, n_j) = \frac{(n_i - \bar{n}_i)(n_j - \bar{n}_j)}{\|n_i - \bar{n}_i\|\,\|n_j - \bar{n}_j\|} \tag{5.4}$$

while $\|A\|$ and $\|B\|$ are the cardinalities of the considered clusters. It is worthy to say that $\mathbf{H}$ will be a symmetric matrix with ones on the main diagonal, whose elements $H(k,l)$ represent the similarity between clusters k and l. The initial matrix $\mathbf{H}$ is a $N \times N$ matrix that contains the simple correlations among the noises $n_1, \ldots, n_N$, then it is updated by deleting rows/columns related to the clusters that have been merged by adding rows/-columns related to the new merged cluster and by recalculating the similarity values between the new cluster and all the remaining ones.
Hierarchical clustering does not require a pre-specified number of clusters. However, in this application we want a partition of disjoint clusters just as in flat clustering: in this case, the hierarchy needs to be cut at some point. Different criteria can be used to determine the cutting point: the criterion based on the *silhouette coefficient* has been used. The use of silhouette coefficient combines both the measures of cohesion (inside clusters) and separation (among clusters). For each noise n_i, the coefficient s_i is simply calculated as in Equation (5.5):

$$s_i = b_i - a_i \tag{5.5}$$

- a_i (cohesion): the average correlation of n_i to all other noises in the same cluster.
- b_i (separation): the average correlation of n_i to all other noises in each of the other clusters, taking the average value with respect to all clusters.

For instance, a very negative value of s_i means that the separation value b_i is highly negative and the cohesion (a_i) is very positive: this indicates that what has been merged is really correlated. So the procedure aims at the smallest possible value of the silhouette coefficient to achieve a good

clustering. We apply this calculation at each loop of the algorithm and at every noise in the data we are examining: more precisely, at the iteration q it is calculated a global measure of the silhouette coefficient SC_q (see Equation (5.6)) by averaging the coefficients related to each noise that belong to a certain cluster and taking the average value with respect to all the current K-clusters.

$$SC_q = \frac{1}{N} \sum_{i=1}^{N} s_i \tag{5.6}$$

Then it is found the minimum coefficient over the $N-1$ obtained and the corresponding index q^* is chosen as the iteration that has to be taken as the last to be executed. According to this, clustering should be done again with the found stop condition; however it has been used a shrewdness to save execution time that consists in saving at each loop the current partition P_q, and then selecting the optimal clustering by simply using the partition P_{q*}. Here is the pseudo-code of the algorithm adopted:

1. Initialization: $K \Leftarrow N$, calculate similarity matrix $\mathbf{H} \in \mathbb{R}^{N \times N}$
2. Loop over $q \Leftarrow 1 \, to \, N-1$
   (a) Search for the pair of clusters $\langle U, V \rangle$ that match the greatest similarity
   (b) Delete from $\mathbf{H}$ the rows and the columns referred to clusters $\langle U, V \rangle$
   (c) Update $\mathbf{H}$ by calculating the new similarity values between the new cluster $Z \Leftarrow \langle U, V \rangle$ and the remaining clusters
   (d) $K \Leftarrow K - 1$
   (e) Calculate the silhouette coefficient SC_q
   (f) Save the current partition P_q
3. Calculate the minimum value of the silhouette coefficients: $q^* \Leftarrow \min_q(SC_q)$
4. Get the optimal partition by selecting the one relative to the iteration q^*, that is the partition P_{q^*}.

At the end of the clustering procedure, the number of clusters M is obtained, that is supposed to be exactly the real number of devices which generated

the given N images of the training set. For each of the obtained M clusters, a reference noise is calculated (as the centroid of the cluster) simply by averaging all the noises belonging to that cluster. The centroids of the clusters provided by the mentioned procedure are then used as the trained classifier to group the images belonging to the test set. The classification is very simple: it consists on comparing the similarity of the current image (taken from the test set) to each of the centroids, and then classify the image to the cluster whose centroid is that which provides the greatest similarity.

TPR vs Size	128 x 128	128 x 256	256 x 256	256 x 512	512 x 512	1024 x 1024	1536 x 2048
Proposed method no enhancer	24.1%	27.3%	24.3%	26.3%	26.5%	38.9%	98.5%
Proposed method with enhancer	51.7%	79.3%	87.3%	96.3%	98.0%	98.7%	99.8%
Method in [70]	52.3%	80.0%	87.7%	96.1%	97.3%	98.8%	NA

Table 5.1: Training phase (or clustering phase): TPR (True Positive Rate) for the proposed method (with and without enhancer) and for the method presented in [70].

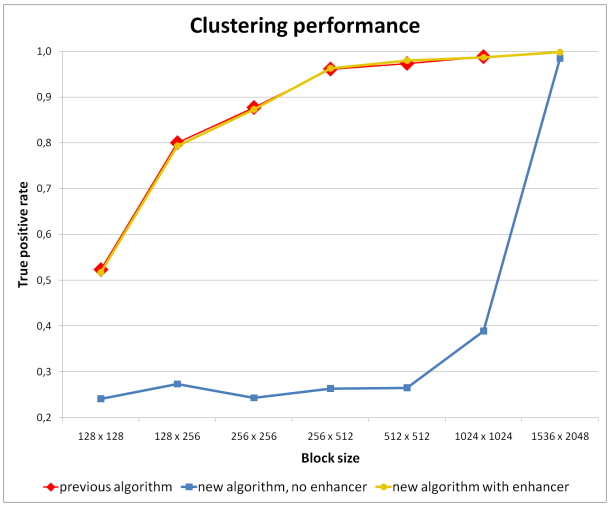
Time vs Size	128 x 128	128 x 256	256 x 256	256 x 512	512 x 512	1024 x 1024	1536 x 2048
Proposed method with enhancer	876	1053	1108	1146	1452	3166	5104
Method in [70]	16183	16200	16284	16437	12215	13896	Not tested

Table 5.2: Training phase: comparison between the proposed method and the method in [70] in terms of execution time (in seconds). The test has been run on *Intel Q6600 quad core, 4Gb RAM, Linux os.*

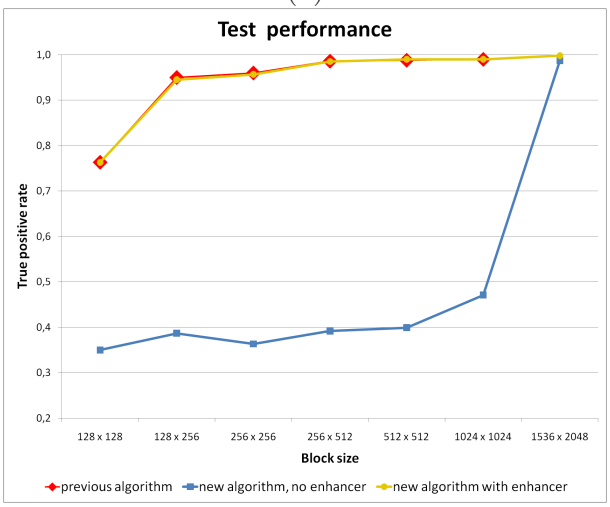
5.3 Experiments

To verify the performances of the presented blind clustering procedure the system has been tested on a dataset populated by 1200 photos at different resolutions (from 3MP to 12MP) taken by six cameras in different time period (200 photos for each camera), here are the cameras: Canon EOS400D (10MP), Canon Digital Ixus i zoom (5MP), Canon Digital Ixus II (3MP), Panasonic DMC-FX12 (7MP), Sony DMC-LZ5 (6MP), FujiFilm FinePix J20 (12MP).

Various experiments have been carried out in different operative conditions:



(a)



(b)

Figure 5.3: Training phase (clustering) (a) and test phase (b) performances.

- image blocks of different sizes, from 128×128 to 1536×2048 pixels;
- using or not a PRNU enhancer;
- training set and test set with a symmetric or an asymmetric distribution of the images within each cluster.

In the first experiment, it has been used a uniform distribution of the images (same number of images for every cameras) both for the training set

and for the test set. The dimension of the training set was of 300 images (50 pictures per camera), while the dimension of the test set was of 600 (100 pictures per camera). Furthermore, not to get misleading evaluation of performances, overlapping between the two data-set has been avoided. Then the TPR (True Positive Rate) and the processing time for the proposed algorithm (with and without enhancer) are compared with the method in [70]. The TPR related to the training phase (or clustering phase) obtained by varying the image block dimension is reported in Table 5.1. The TPR achieved for the proposed algorithm (with enhancer) (second row in Table 5.1) and for the algorithm in [70] (third row in Table 5.1) are comparable for all image block sizes but in our algorithm is possible to reach higher image block resolution and therefore better TPR (last column in Table 5.1). This is due to the fact that our method performs better in term of time execution as reported in Table 5.2 (about 14 minutes against more than 4 hours for the smaller block size in the first column of Table 5.2). After that, a testing phase has been carried out in order to evaluate the performances of the proposed method and the results are shown in Figure 5.3b. The results are comparable in term of TPR to the results obtained during the training phase (or clustering phase) as reported in Figure 5.3a. It is possible to point out that using enhancer becomes less important when image block of higher resolution are taken in account, that is when approaching the actual image resolution. In the second experiment, the robustness of our algorithm with respect to a non uniform data set has been tested and a comparison with the algorithm in [70] has been made. Five different groups have been created for the training set while the test set is the same as before. For each group the difference for the number of pictures belonging to each camera is increased, as we can see in Table 5.3.

Group	Cam 1	Cam 2	Cam 3	Cam 4	Cam 5	Cam 6
A	55	45	55	45	55	45
B	60	40	60	40	60	40
C	70	30	70	30	70	30
D	80	20	80	20	80	20
E	90	10	90	10	90	10

Table 5.3: Non uniform data set distribution.

In Table 5.4 the TPR for training and test phase related to the proposed method is reported and in Table 5.5 the results obtained for the method in [70] are shown.

TPR vs Size	A	B	C	D	E
1024x1024	95.2%	92.0%	84.8%	76.8%	68.4%
512x512	89.3%	87.6%	80.0%	71.0%	56.0%
256x256	82.1%	76.7%	75.3%	66.5%	53.0%

(a)

TPR vs Size	A	B	C	D	E
1024x1024	97.8%	96.3%	94.7%	91.2%	86.5%
512x512	93.3%	91.0%	87.6%	84.9%	82.6%
256x256	89.8%	84.4%	82.5%	73.1%	61.1%

(b)

Table 5.4: The TPR in the training (a) and testing (b) phase for the proposed method.

TPR vs Size	A	B	C	D	E
512x512	89.7%	87.3%	79.0%	69.3%	49.7%
256x256	81.9%	77.4%	74.9%	66.5%	46.3%

(a)

TPR vs Size	A	B	C	D	E
512x512	94.0%	91.6%	86.6%	79.8%	56.2%
256x256	89.8%	83.9%	81.6%	72.1%	50.9%

(b)

Table 5.5: The TPR in the training (a) and testing (b) phase for the method in [70].

The two algorithms have similar performances related to the group A, B and C, while the new algorithm shows better performances especially for non

uniform datasets (groups D, E). Furthermore, with the proposed algorithm is possible to evaluate the performance for the image block resolution 1024×1024 achieving a TPR of 86.5% even for the group E. This kind of experiment is hardly feasible for the algorithm in [70] due to its high computational time as shown in Table 5.2.

Chapter 6

A SIFT-based forensic method for copy-move attack detection

The other main multimedia forensics topic is about image tampering detection [75], assessing the authenticity or not of a digital image. Information integrity is fundamental in a trial but it is clear that the advent of digital pictures and relative ease of digital image processing makes today this authenticity uncertain. Examples of this problem that recently appeared in several newspapers and TV news, are given in Figure 6.1 and in Figure 6.2.

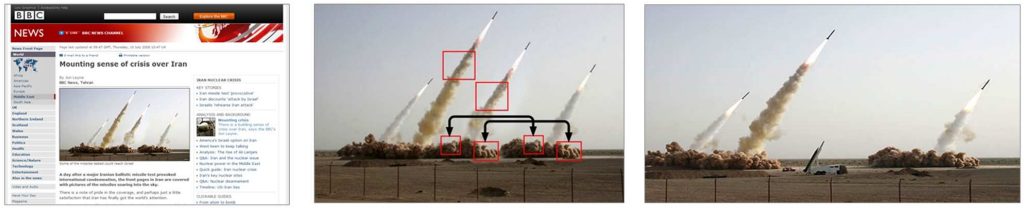


Figure 6.1: *An example of image tampering appeared on press in July 2008. The feigned image (on the left) shows four Iranian missiles but only three of them are real (image on the right).*

Modifying a digital image, to change the meaning of what is represented in it, could be crucial when it is used in a court of law, where images are presented as basic evidences to influence the judgement.Furthermore, it would be interesting, once established that something has happened, to understand what: if an object or a person has been covered, if a part of the image has been cloned, if something has been copied from another image or, even more, if a combination of these processes have been carried out. In particular, when

Figure 6.2: *A close look at this picture shows that many elements of this picture are cloned over and over.*

an attacker creates his feigned image by cloning an area of the image onto another zone (copy-move attack) to cancel something that was awkward, he is often obliged to apply a geometric transformation to satisfactorily achieve his aim. In this paper this issue is investigated, individuating if the copy-move tampering has taken place and estimating the parameters of the transformation occurred (i.e. horizontal and vertical translation, scaling factors, rotation angle). On the basis of the work proposed in [76], a new methodology which answers to this requirement is presented hereafter. Such a technique is based on Scale Invariant Features Transform (SIFT) [77] algorithm, which is used to robustly detect and describe clusters of points belonging to cloned areas. Successively, these points are exploited to reconstruct the parameters of the occurred geometric transformation. The proposed technique has also been tested against splicing attack, that is when an image block is duplicated onto another different image. In fact, in a context where the source image is available (e.g. the forensic analyst has to check a suspect dataset which contains both the source and the destination image) this methodology can still be applied.

The chapter is structured as follows: in Section 6.1 related works regarding copy-move forgery detection techniques are presented and SIFT technique is introduced. In Section 6.2 the proposed method is discussed in

its three main stages and experimental results on forgery detection performances and on applied transformation parameters estimation are presented in Section 6.3.[1]

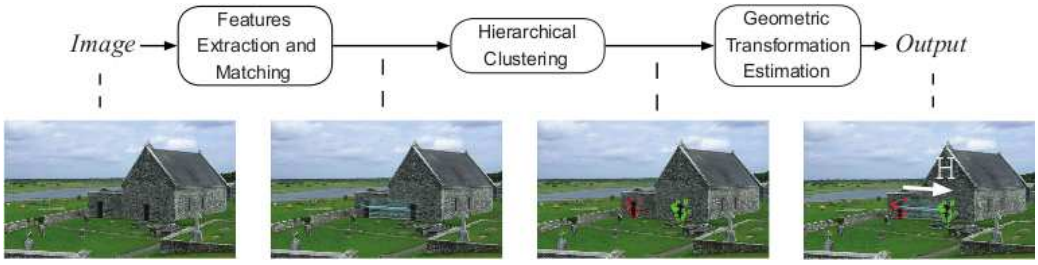


Figure 6.3: *Overview of the proposed system. SIFT matched pairs and clusters.*

6.1 SIFT Features for Image Forensics

One of the most common image manipulations is to clone (copy and paste) portions of the image to conceal a person or an object in the pictured scene. When this is done with care and retouch tools are used, it can be difficult to detect cloning visually. Moreover since the copied parts are from the same images some components (e.g noise and color) will be compatible with the rest of the image and thus will not be detectable using methods that look for incompatibilities in statistical measures in different parts of the image [78, 79]. Furthermore, since the cloned regions can be of any shape and location, it is computationally impossible to search all possible image locations and sizes with an exhaustive search as pointed out in [55]. The problem of copy-move forgery detection has been faced by proposing different approaches each of these based on the same concept: a copy-move forgery introduces a *correlation* between the original image area and the pasted one. Several methods search this dependence dividing the image into overlapping blocks and then applying a feature extraction process in order to represent the image blocks by using a low dimensional representation. For example in [49] the the averages of red, green and blue components respectively are chosen as three features together with other four computed on overlapping blocks by calculating the energy distribution of luminance along four differ-

[1] This work has been presented in the International Conference in Acoustic Speech and Signal Processing, Dallas TX, USA, 2010.

ent directions. In [80] the features are represented by the SVD (Singular Value Decomposition) performed on low-frequency coefficients of the block-based DWT (Discrete Wavelet Transform). The authors in [48] propose a block representation by blur invariants. Their specific aim is to find features invariant to the presence of blur artifacts that a falsifier can apply to make detection of forgery more difficult. Then they used PCA (Principal Component Analysis) to reduce the number of features and a k-tree to identify the interested regions. In [81] authors present a technique to detect cloning when the copied part has been modified using two specific tools, the Photoshop healing brush and the Poisson cloning. Others two algorithms [55] and [56] based on using low dimensional representation of blocks and fast sorting to improve efficiency have been developed to detect copy-move image regions. In particular, the authors in [55] apply a discrete cosine transform (DCT) to the block. Duplicated regions are then detected by lexicographically sorting the DCT block coefficients and grouping similar blocks with the same spatial offset in the image. While in [56] the authors apply a principal component analysis (PCA) on image blocks to yield a reduced-dimension representation. Duplicated regions are again detected by lexicographically sorting and grouping all of the image blocks. A related approach is the method in [82] where a Fourier Mellin Transform is applied on each block. A forgery decision is made when there are more then a given number of blocks that are connected to each other and the distance between block pairs is the same. To create a convincing forgery, it is often necessary to resize, rotate, or stretch portions of an image. For example, when creating a composition of two objects, one object may have to be resized to match the relative heights. This process requires re-sampling of the original image introducing specific periodic correlations between neighboring pixels. The presence of these correlations due to the re-sampling can be used to detect that something happened to the image [83] but not to detect the specific manipulation.

So a good copy-move forgery detection should be robust to some types of transformations as rotation and scaling and also to some manipulations including JPEG compression, Gaussian noise addition and gamma correction. Most of the existing methods do not deal with all these manipulations and are often computationally prohibitive. In particular the method in [56] is not able to detect scaling or rotation transformation, whereas with the methods in [55] and [82] only small variations in rotation and scaling are identifiable as reported in [84]. The authors in [85] make an attempt to overcome this

problem solving copy-move identification when only rotation of the copied area takes place by using the Zernike moments. This issue is also discussed in [86] where rotation transformation and JPEG compression and Gaussian noise manipulations are analyzed to understand how they could affect the copy-move detection. Authors in [87] instead propose a method to detect duplicated and transformed regions through the use of a block description invariant to reflection and rotation such as the log-polar block representation summed along its angle axis. Finally a comparison among some of copy-move methods described above has been reported in [88] evaluating the performance of each methods with and without geometric transformation applied to the copied patch.
Nowadays visual local features (e.g SIFT Scale Invariant Features Transform) have been widely used for the particular tasks of image retrieval and object detection and recognition, due to their robustness to several geometrical transformations (e.g. rotation and scaling), occlusions and clutter. More recently few attempts have been done to apply this kind of features also in the digital forensics domain; just as an example, SIFT features have been used for fingerprint detection [89] and shoeprint image retrieval [90].
Two preliminary works on copy-move forgery detection based on SIFT features has been recently proposed by Huang *et al.* [91] and Pan *et al.* [92], they report experimental results only on a few example images and they do not provide any estimation of the parameters of the forgery transformation. In this scenario is placed the proposed method that on the basis of our previous work in [76] is able to detect and then estimate the geometrical transformation occurred in a copy-move forgery.

6.1.1 Review on SIFT method

In particular the methods based on visual local features typically start with a detection step, in which interest points are localized, then robust local descriptors are built so as to be invariant with respect to orientation, scale and affine transformations. Mikolajczyk and Schmid provide a comprehensive analysis of several local descriptors in [93] while local affine region detectors are surveyed in [94]. Their work confirm that Scale Invariant Features Transform (SIFT) [77] are a good solution because of their high performance and relatively low computational cost. This method can be roughly summerized as the following four steps: *i)* scale-space extrema detection; *ii)* keypoint

localization; *iii)* assignment of one (or more) canonical orientation; *iv)* generation of keypoint descriptors.

In other words, given an input image, SIFT features are detected at different scales by using a scale-space representation implemented as an image pyramid. The pyramid levels are obtained by Gaussian smoothing and subsampling of the image resolution while interest points are selected as local extrema (min/max) in the scale-space. These keypoints, also referred as $\mathbf{x}_i$, are extracted by applying a computable approximation of the Laplacian of Gaussian (LoG). In particular, the SIFT algorithm approximates LoG by iteratively computing the difference between two nearby scales in the scale-space. This idea is referred to as the Difference of Gaussians (DoG) approach.

In order to guarantee invariance to rotations, the algorithm assigns to each keypoint a canonical orientation o. To determine this orientation, a gradient orientation histogram is computed in the neighborhood of the keypoint. The basic idea is to assign a consistent canonical orientation to each keypoint, based on local image properties (computed at the level of the image pyramid where the keypoint was detected). The keypoint descriptor can be represented relatively to this orientation and therefore achieve robustness to image rotation.

Once these keypoints are detected, and canonical orientations are assigned, SIFT descriptors are computed at their locations in both image plane and scale-space. Each descriptor consists in a histogram $\mathbf{S}$ of 128 elements, obtained from a 16×16 pixels area around the corresponding keypoint. This area is selected using the coordinates (x, y) of the keypoint as the center and its canonical orientation as the origin axis. The contribution of each pixel is obtained by calculating image gradient magnitude and orientation in scale-space and the histogram is computed as the local statistics of gradient orientations (considering 8 bins) in 4×4 sub-patches.

Summarizing the above, given an image I, this procedure ends with a list of N keypoints each of which is completely described by the following informations:

$$\mathbf{x}_i = \{x, y, \sigma, o, \mathbf{S}\}, \tag{6.1}$$

where (x, y) are the coordinates in the image plane, σ is the scale of the keypoint (related to the level of the image-pyramid used to compute the descriptor), o is the canonical orientation (used to achieve rotation invariance) and $\mathbf{S}$ is the final SIFT descriptor.

6.2 The proposed method

The proposed approach is based on SIFT technology to extract robust features which can allow to discover if a part of an image was copy-moved and furthermore which geometrical transformation was applied. In fact, the copied part has basically the same appearance of the original one, thus keypoints extracted in the forged region will be quite similar to the originals. Therefore matching among SIFT features can be adopted for the task to determine a possible tampering. A simple schematization of the three steps the whole system is based on is shown in Figure 6.3: the first phase consists of SIFT features extraction and keypoint matching, the second one is devoted to cluster such keypoints and assess forgeries detection, and the third one is in charge to estimate the occurred geometric transformation, if a tampering has been individuated.

6.2.1 SIFT features extraction and keypoints matching

Given a to-be-checked image, a set of keypoints $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ with their SIFT descriptors $\{\mathbf{S}_1, \ldots, \mathbf{S}_n\}$ is extracted (see sub-section 6.1.1). A matching operation is performed in the SIFT space among $\mathbf{S}_i$ vectors of each keypoint to identify similar local patches in the image. The best candidate match for each keypoint $\mathbf{x}_i$ is found by identifying its nearest neighbor from the other $n-1$ keypoints, which is the keypoint with the minimum Euclidean distance in the SIFT space. In order to decide for a matching between two keypoints (i.e. "are these two descriptors the same or not?"), simply evaluating the distance between two descriptors with respect to a global threshold does not perform well. This is due to the high-dimensionality of the feature space (128) in which some descriptors are much more discriminative than others [77]. We can obtain a more effective measure by using the ratio between the distance of the closest neighbor to that of the second-closest one, and comparing it with a threshold. For seek of clarity, given a keypoint we define a similarity vector $\mathbf{D} = \{d_1, d_2, \ldots, d_{n-1}\}$ that represents the sorted euclidean distances with respect to the other descriptors. The keypoint is matched only if the constraint in Equation 6.2 is satisfied:

$$\frac{d_1}{d_2} < T \tag{6.2}$$

the threshold T is usually fixed to 0.6 (this choice is suggested in [77]). By iterating on each keypoint belonging to $\mathbf{X}$, we can obtain the set of matched points. All the matched keypoints are held, instead isolated ones are discarded and no more considered in the forensic analysis (see the second test image from left in Figure 6.3): already at this stage a draft idea of the authenticity of the image is provided. But it can happen that images, really containing areas with very similar texture, can yield to matched keypoints that might induce false alarms: the following two steps of the proposed methodology tries to reduce this possibility. On the other side, it is worthy to point out that can occur the case where no matched keypoints are obtained, mainly because, unfortunately, salient features are not revealed in the forged patch (e.g. when an object is hidden with a flat patch): anyway this is a very well-known open issue in SIFT-related scientific literature.

6.2.2 Clustering and forgeries detection

To identify possible cloned areas, an *agglomerative hierarchical clustering* [95] is performed on spatial locations of the matched points. Hierarchical clustering creates a hierarchy of clusters which may be represented in a tree structure. The algorithm starts by assigning each keypoint to a cluster; after that it computes all the reciprocal spatial distances among clusters, finds the closest pair of clusters and then merges them into a single cluster. Such computation is iteratively repeated till a final merging situation is achieved. The way such final merging can be accomplished is basically conditioned both by the linkage method adopted and by the threshold used to stop clusters' grouping. Several linkage methods exist in literature and experimental tests to evaluate their performances and to estimate the cut-off threshold T_h have been carried out (see subsection 6.3.1 for a detailed description of such experiments). In particular, three different linkage methods, briefly described in the following, have been taken into account: *Single*, *Centroid* and *Ward's* linkage. Given two clusters P and Q containing n_P and n_Q objects respectively and indicated with $\mathbf{x}_{Pi}$ and $\mathbf{x}_{Pj}$ the i^{th} and the j^{th} object respectively in the clusters P and Q, the diverse linkage method operates as it follows:

- *Single* linkage uses the smallest euclidean distance between objects in

the two clusters:

$$dist(P,Q) = \min(\|\mathbf{x}_{Pi}, \mathbf{x}_{Qj}\|_2),\ i = [1, n_P], j = [1, n_Q].$$

- *Centroid* linkage uses the euclidean distance between the centroids of the two clusters:
$$dist(P,Q) = \|\overline{\mathbf{x}}_P - \overline{\mathbf{x}}_Q\|_2$$
where
$$\overline{\mathbf{x}}_P = \frac{1}{n_P}\sum_{i=1}^{n_P}\mathbf{x}_{Pi} \quad \text{and} \quad \overline{\mathbf{x}}_Q = \frac{1}{n_Q}\sum_{i=1}^{n_Q}\mathbf{x}_{Qi}.$$

- *Ward's* linkage evaluates the increment/decrement in the "error sum of squares" (ESS) after merging the two clusters into a single one with respect to case of the two separated clusters:
$$ESS(P) = \sum_{i=1}^{n_P}|\mathbf{x}_{Pi} - \overline{\mathbf{x}}_P|^2$$
$$\Delta_{dist}(P,Q) = ESS(PQ) - [ESS(P) + ESS(Q)]$$
where $\overline{\mathbf{x}}_P$ is the centroid again and PQ indicates the combined cluster.

According to the adopted linkage method, a specific tree structure is obtained. In addition to this, the proper choice of the threshold T_h to determine where to cut the tree and consequently which is the final number of clusters is crucial. The parameter which is utilized to be compared with T_h is the "inconsistency coefficient" (IC) which characterizes each clustering operation; the higher the value of this coefficient, the less similar the objects connected by the link, thus when it exceeds the threshold T_h clustering stops. IC takes basically into account the average distance among clusters and does not allow to join clusters spatially too far at that level of hierarchy. It is easy to understand that an appropriate assumption of T_h hardly influences tampering detection performances. At the end of clustering procedure, however clusters which do not contain a significant number (more than three) of matched keypoints are eliminated. On this basis, to optimize detection performances and consequently to the carried out experimental tests (see again subsection 6.3.1), it has been established to consider that an image has been altered by a copy-move attack, if the method detects two (or more) clusters with at least three pairs of matched points that link a cluster to

another one. This aspect has been investigated and this assumption grants a good trade-off between the need to provide a low false alarm rate and the necessity to deal with flat duplicated regions.

6.2.3 Geometric transformation estimation

When an image has been classified as non-authentic, the proposed method allows to determine which is the geometrical transformation occurred between the original area and its copy-moved version. Let the matched point coordinates be, for the two areas, $\tilde{\mathbf{x}}_i = (x, y, 1)^{\mathtt{T}}$ and $\tilde{\mathbf{x}}'_i = (x', y', 1)^{\mathtt{T}}$ respectively, their geometric relationships can be defined by an affine homography which is represented by a 3×3 matrix $\mathtt{H}$ as:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \mathtt{H} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{6.3}$$

This matrix can be computed by resorting at three matched points at least. In particular, we determine $\mathtt{H}$ by using Maximum Likelihood estimation of the homography [96]. This method seeks homography $\mathtt{H}$ and pairs of perfectly matched points $\hat{\mathbf{x}}_i$ and $\hat{\mathbf{x}}'_i$ that minimizes the total error function as in Equation 6.4:

$$\sum_i \left[d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 + d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2 \right] \text{ subject to } \hat{\mathbf{x}}'_i = \mathtt{H}\hat{\mathbf{x}}_i \; \forall i. \tag{6.4}$$

However mismatched points (*outliers*) can severely disturb the estimated homography. For this purpose we perform the previous estimation by applying the RANdom SAmple Consensus algorithm (RANSAC) [97]. Such algorithm randomly selects a set (in our case three pairs of points) from the matched points and estimates the homography H, then all the remained points are transformed according to H and compared in terms of distance with respect to their corresponding matched ones. If this distance is under or above a certain threshold β, they are catalogued as *inliers* or *outliers* respectively. After a pre-defined number N_{iter} of iterations, the estimated transformation which is associated with the higher number of inliers is chosen. In our experimental tests, N_{iter} has been set to 1000 and the threshold β to 0.05; this is due to the fact that we used a standard method of normalization of the data for homography estimation. The points are translated so

that their centroid is at the origin and then they are scaled so that the average distance from the origin is equal to $\sqrt{2}$. This transformation is applied to both of the two areas $\mathbf{x}_i$ and $\mathbf{x}'_i$ independently.

Once the affine homography is found, rotation and scaling transformations can be computed by its decomposition, while translation can be determined by considering the centroids of the two matched clusters. In particular, $\mathtt{H}$ can be represented as:

$$\mathtt{H} = \begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{0}^{\mathtt{T}} & 1 \end{bmatrix} \quad \text{where} \quad \mathtt{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}. \tag{6.5}$$

The matrix $\mathtt{A}$ is the composition of rotation and non-isotropic scaling transformations. In fact, it can always be decomposed as

$$\mathtt{A} = \mathtt{R}(\theta)(\mathtt{R}(-\Phi)\mathtt{SR}(\Phi)) \tag{6.6}$$

where $\mathtt{R}(\theta)$ and $\mathtt{R}(\Phi)$ are rotations by θ and Φ respectively, and $\mathtt{S} = \mathtt{diag}(s_1, s_2)$ is a diagonal matrix for the scaling transformation. Hence, the $\mathtt{A}$ defines the concatenation of a rotation by Φ, a scaling by s_1 and s_2 respectively in the rotated x and y directions; a rotation back by $-\Phi$; and finally another rotation by θ. This decomposition is computed directly by the SVD (Singular Value Decomposition). In fact, the matrix $\mathtt{A}$ can be also rewritten as: $\mathtt{A} = \mathtt{USV}^{\mathtt{T}} = (\mathtt{UV}^{\mathtt{T}})(\mathtt{VSV}^{\mathtt{T}}) = \mathtt{R}(\theta)(\mathtt{R}(-\Phi)\mathtt{SR}(\Phi))$ since $\mathtt{U}$ and $\mathtt{V}$ are orthogonal matrices.

6.3 Experimental results

In this Section some of the experimental results carried out to evaluate the proposed methodology are provided. In particular, two are the main kinds of the tests: firstly, on a small dataset named DB220, a benchmarking of the technique is done to properly set the operative threshold T_h and to compare it with other methods known in literature; secondly, on a larger dataset named DB2000, a complete evaluation is carried out by testing the system against different types of modifications.

Both datasets are composed by images with different contents coming from the Columbia photographic images repository [98] and from a personal collection. The first dataset DB220 is composed by 220 images: 110 are tampered images and 110 are originals. The images resolution varies from

722×480 to 800×600 pixels and the size of the forged patch covers, on the average, 1.2% of the whole image. The images have been tampered by coping and pasting an image part of different dimension over another area of the same photo by applying diverse transformations (see hereafter). The second dataset DB2000 is composed by 2000 photos of 2048×1536 pixels (3M pixels) and the forgery is, on the average, 1.12% of the whole image: so it is again quite small and similar to the DB200 dataset case. This aspect is very crucial because the cloned patch size drastically influences the performances of the SIFT search, obviously the greater the cloned area the higher the number of possible SIFT keypoints. Furthermore, to reproduce as much as possible a practical situation, the number of original and altered images belonging to the DB2000 dataset is not the same: 1300 original images and 700 tampered images have been taken. The forged images are obtained, in both the datasets, by randomly selecting (both as location and as dimension) an image area (squared or rectangular) and copy-pasting it over the image after having applied a number of different attacks such as translation, rotation, scale (symmetric/asymmetric) or a combination of them. Table 6.1 and Table 6.2 summarize the geometric transformations for the applied attacks for the first DB220 dataset (10 attacks, from A to J in Table 6.1) and for the second DB2000 (14 attacks, from a to o in Table 6.2) respectively. In particular, for each attack, is reported the rotation θ expressed in degrees and the scaling factors s_x, s_y applied to the x and y axis of the cloned image part (e.g. in the attack G, the x and y axes are scaled by 30%, and no rotation is performed).

Attack	θ °	s_x	s_y
A	0	1	1
B	10	1	1
C	20	1	1
D	30	1	1
E	40	1	1

Attack	θ °	s_x	s_y
F	0	1.2	1.2
G	0	1.3	1.3
H	0	1.4	1.2
I	10	1.2	1.2
J	20	1.4	1.2

Table 6.1: *The 10 different combinations of geometric transformations applied to the original patch for the DB220 dataset.*

Attack	θ °	s_x	s_y
a	0	1	1
b	0	0.5	0.5
c	0	0.7	0.7
d	0	1.2	1.2
e	0	1.6	1.6
f	0	2	2
g	0	1.6	1.2

Attack	θ °	s_x	s_y
h	0	1.2	1.6
i	5	1	1
j	30	1	1
l	70	1	1
m	90	1	1
n	40	1.1	1.6
o	30	0.7	0.9

Table 6.2: *The 14 different combinations of geometric transformations applied to the original patch for the DB2000 dataset.*

6.3.1 Threshold settings for forgeries detection

As said before, in this subsection, first of all, the proposed method is analyzed to determine the best settings for the cut-off threshold T_h introduced in Section 6.2.2 according to the chosen linkage method. Such values will be set up for the successive phase of experiments and comparisons. To address this problem, the following experiment has been set-up applying a *4*-fold cross-validation process: from the database of 220 images (DB220), 165, that is 3/4 of the image set, (82 tampered and 83 original) have been randomly chosen to perform a training to find the best threshold T_h for each of the three considered linkage methods (*Single*, *Centroid*, *Ward's*); the remaining 55 images (1/4 of the whole set) have been used in a successive testing phase to evaluate detection performances of the proposed technique. During training, the threshold T_h ranged in the interval [0.8, 3] with steps of 0.2. The experiment was repeated 4 times, by cyclically exchanging the four image sub-sets belonging to the training (3 sub-sets) and to the testing set (1 sub-set), and the results have been averaged. Detection performances have been measured in terms of True Positive Rate (TPR) and False Positive Rate (FPR), where TPR is the fraction of tampered images correctly identified as such and FPR is the fraction of original images that are not correctly identified as such, that is:

- TPR = $\frac{images\ detected\ as\ forged\ being\ really\ forged}{forged\ images}$
- FPR = $\frac{images\ detected\ as\ forged\ being\ instead\ original}{original\ images}$

Again we underline that has been assumed to consider that an image has been altered by a copy-move attack, if the method detects two (or more) clusters with at least three pairs of matched points that link a cluster to another one (as debated in subsection 6.2.2).

In Table 6.3, for each linkage method, the TPR and the FPR obtained during the training phase are reported with respect to the threshold T_h which varies in the established range.

	Single		*Centroid*		*Ward's*	
T_h	FPR(%)	TPR(%)	FPR(%)	TPR(%)	FPR(%)	TPR(%)
0.8	2.729	41.827	1.822	23.626	0.911	10.906
1	5.455	70.001	4.547	56.373	3.636	32.739
1.2	8.180	89.994	7.273	90	7.273	82.714
1.4	8.180	95.456	8.180	95.456	7.273	90.905
1.6	***8.180***	***98.185***	7.273	97.274	8.180	97.274
1.8	7.269	96.360	***8.180***	***98.182***	9.088	99.089
2	6.362	91.820	7.269	95.456	9.088	100
2.2	5.451	82.721	5.451	92.723	***8.177***	***100***
2.4	4.544	63.639	4.544	84.536	7.269	96.364
2.6	2.726	48.185	2.729	70.897	7.273	89.998
2.8	0.911	22.726	1.822	46.360	3.640	78.170
3	0.911	15.461	0.911	18.179	3.640	61.813

Table 6.3: *Training phase: TPR and FPR values (in percentage) for each metric with respect to T_h.*

The goal was to minimize the FPR while maintaining a very high TPR; as it can be seen FPR is almost always very low, on the contrary TPR is very variable, so the optimal threshold T_h has been chosen, as evidenced in Table 6.3, for the maximum value of TPR that means 1.6 for the *Single* linkage method, 1.8 for the *Centroid* and 2.2 for the *Ward's* linkage. Finally, on the basis of such analysis, the test phase has been launched for the best metrics by using the T_h previously obtained in the training phase. The final detection results, again averaged on the 4 repetitions, are reported in Table 6.4. These results show that the proposed method performs satisfactorily providing a low FPR though maintaining an high rate of correct tampering detection basically for all the used linkage method, though *Ward's* metric seems to be slightly better. It is possible to conclude that the choice of

linkage method is not so fundamental while T_h setting is crucial.

	Single	*Centroid*	*Ward's*
FPR (%)	8.16	8.16	8
TPR (%)	98.21	98.17	100

Table 6.4: *Test phase on DB220 dataset: detection results in terms of FPR and TPR.*

Furthermore, for the cases of correctly detected forged images, the estimation of the geometric transformation parameters which bring the original patch onto the forged one has also been computed. The Mean Absolute Error (MAE) between each of the true values of the transformation parameters and the estimated ones, again averaged on all the images (correctly detected as forged) of the 4 repetitions are reported in Table 6.5.

MAE (t_x)	MAE (t_y)	MAE (θ)	MAE (s_x)	MAE (s_y)
4.04	2.48	0.94	0.021	0.015

Table 6.5: *Transformation parameters estimation errors for the DB220 (*Single *linkage method with $T_h = 1.6$, as previously underlined other metrics give similar performances). The values t_x and t_y are expressed in pixels while θ in degrees.*

A	t_x	$\hat{t}_x$	$\lvert e\rvert$	t_y	$\hat{t}_y$	$\lvert e\rvert$	θ	$\hat{\theta}$	$\lvert e\rvert$	s_x	$\hat{s}_x$	$\lvert e\rvert$	s_y	$\hat{s}_y$	$\lvert e\rvert$
A	304	304.02	0.02	80.5	81.01	0.51	0	0.040	0.040	1	1.004	0.004	1	0.998	0.002
B	304	305.20	1.20	80.5	82.42	1.92	10	9.963	0.037	1	1.001	0.001	1	0.999	0.001
C	304	305.55	1.55	80.5	82.64	2.14	20	20.009	0.009	1	1.006	0.006	1	0.998	0.002
D	304	305.04	1.04	80.5	82.49	1.99	30	30.092	0.092	1	1.002	0.002	1	0.998	0.002
E	304	306.08	2,08	80.5	78.43	2.07	40	39.932	0.067	1	1.007	0.007	1	1.004	0.004
F	304	304.88	0.88	80.5	80.41	0.09	0	0.080	0.080	1.2	1.202	0.002	1.2	1.198	0.002
G	304	305.07	1.07	80.5	79.87	0.63	0	0.108	0.108	1.3	1.304	0.004	1.3	1.303	0.003
H	304	305.78	1.78	80.5	80.18	0.32	0	0.037	0.037	1.4	1.403	0.003	1.2	1.206	0.006
I	304	305.23	1.23	80.5	81.76	1.26	10	9.910	0.090	1.2	1.203	0.003	1.2	1.201	0.001
J	304	305.02	1.02	80.5	80.82	0.32	20	20.067	0.067	1.4	1.404	0.004	1.2	1.198	0.002

Table 6.6: *Transformation parameters estimation on image* Cars*. The values t_x and t_y are expressed in pixels while θ in degrees.*

Results show an high degree of precision in the estimate of the various parameters of the affine transformation. In addition to this, in Table 6.6, as example, for one of the test image belonging to the DB220, named *Cars* (see Figure 6.4 top-left corner), each transformation parameter (the original

value applied to the patch, the estimated one and the absolute error ($|e|$)) is reported in detail. It can be observed how reliable the estimate is, specifically for the scale parameters and also for an asymmetric scaling combined with a rotation.

Qualitative evaluation

Hereafter, some experimental results on images where a copy-move attack has been performed by taking into account the context are reported. In this case the patch is selected according to the specific goal to be achieved and, above all, transformed by paying attention to perfectly conceal the occurred modification. Alterations are not recognizable at least at a first rough watch and a forensic tool could help in investigation action. In Figure 6.4, six of these specific cases are pictured by presenting the tampered image and the corresponding one where SIFT matches, extracted by the proposed method, are highlighted. It is interesting to point out how the cloned patches are individuated; the case of a multiple cloning, as evidenced in the top-right corner where the goose is copy-moved twice (different scale factor have been applied), is well managed too. In particular, it is worthy to notice how the technique still works though the number of keypoints is reduced when the goose (top-left) is also down-scaled. Another interesting situation concerns the individuation of a cloned patch for the image named *Dune* (middle of third and fourth rows) where, though the duplicated area is quite flat, the method is able to detect a sufficient number of matched keypoints. On the contrary, an opposite case is registered for the image named *Santorini* (right of third and fourth rows), where a very high amount of matched keypoints is obtained; now the cloned block is very textured and though it has undergone a geometrical transformation to be properly adapted to the context, the SIFT algorithm is so robust not to be disturbed.

Copy-move methods comparison

The proposed approach has been compared with the implementations both of the method presented in [55], based on DCT (Discrete Cosine Transform),

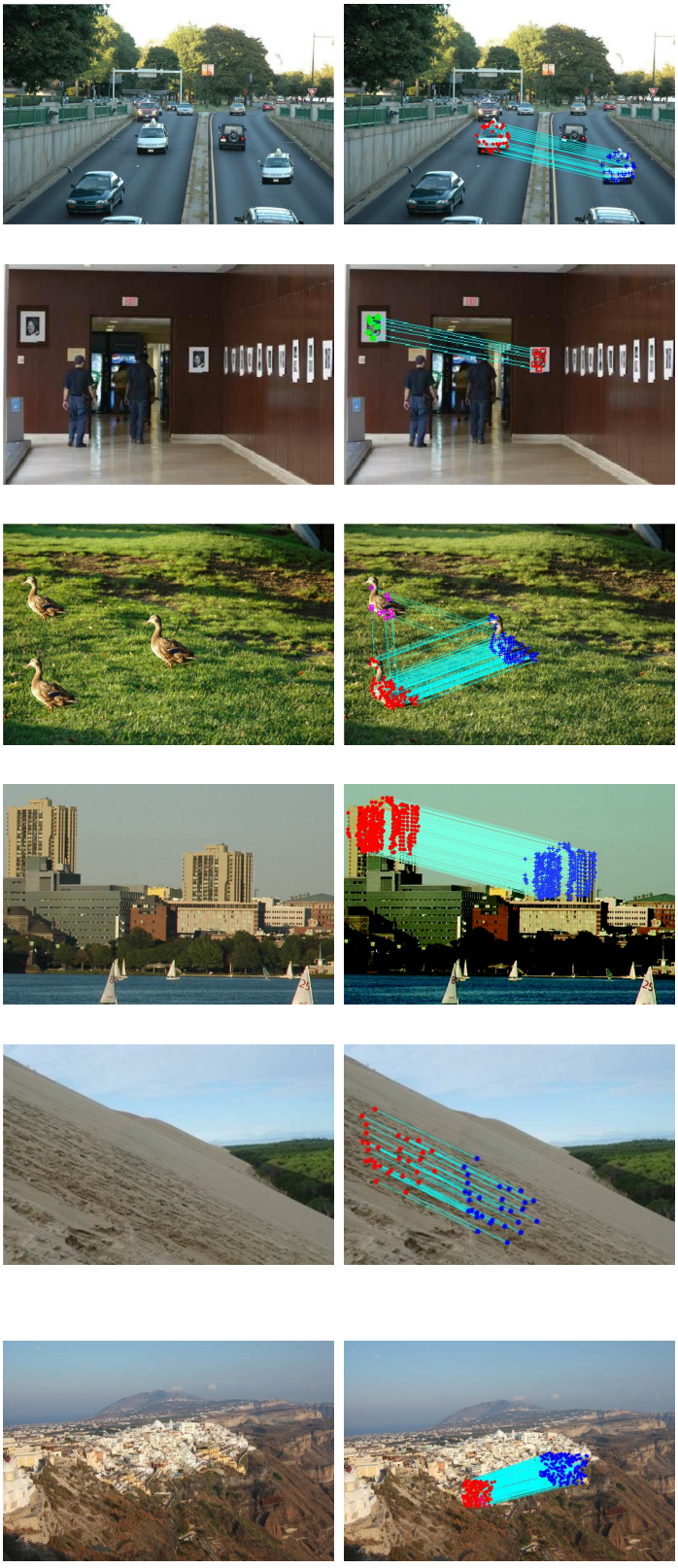


Figure 6.4: *Some examples of tampered images are pictured in the first and the third rows, the corresponding detection results are reported in the second and in the forth rows.*

and of the technique introduced in [56], based on PCA (Principal Component Analysis), (both have been briefly described in Section 6.1). The input parameters required by the two methods are set as it follows: $b = 16$ (number of pixels per block), $N_n = 5$ (number of neighborhood rows to search in the lexicographically sorted matrix), $N_f = 1000$ (threshold for the minimum frequency) and $N_d = 22$ (threshold to determine a duplicated block). These parameters are used in both the algorithms, while $e = 0.01$ (fraction of the ignored variance along the principle axes after PCA is computed) and $Q = 256$ (number of the quantization bins) are only used for the method in [15]. For the proposed technique the *Ward's* linkage with a threshold $T_h = 2.2$ has been assumed. The experimental test has been launched over the whole DB220 image database by using a machine *Intel Q6600 quad core, 4Gb RAM, linux os* and the FPR, TPR and the processing time have been evaluated. Table 6.7 shows the detection performances and the processing time on average (in seconds) for an image relatively to each methodology.

Method	FPR (%)	TPR (%)	time
[55]	84	89	294.69
[56]	86	87	70.97
Proposed	8	100	4.94

Table 6.7: *TPR, FPR values (%) and processing time (one image averagely) for each method.*

The results point out that the proposed method performs better with respect to the others methods; in fact the processing time (per image) is on average about 5 seconds, whereas the other two take more than 1 minute and almost 5 minutes respectively. Furthermore DCT and PCA methods, though presenting an acceptable TPR, fail when a decision about original images is required (high FPR values in Table 6.7). Anyway this is basically due to the incapacity of such methods to properly deal with cases where a geometrical transformation which is not just a translation is applied to the copy-moved patch. For the specific case of simple patch translation FPR is 0% for all the three methods.

6.3.2 Test on a large dataset

In this Section, experimental results obtained on a larger dataset, named DB2000, to verify the behavior of the proposed technique are presented; detection performances and geometric transformation parameters estimation are investigated as well. Furthermore some tests to check the robustness of the method against usual operations such as JPEG compression or noise addition, an image can undergo, have been carried out; such kinds of processing have been considered as applied both to the whole forged image and only to the altered image patch.

	Single		*Centroid*		*Ward's*	
T_h	FPR(%)	TPR(%)	FPR(%)	TPR(%)	FPR(%)	TPR(%)
0,8	3,41	51,86	1,69	32,29	0,54	11,43
1	5,56	70,19	4,92	62,43	3	51,29
1,2	10,28	89,95	10,31	87,43	9,54	83,86
1,4	10,95	91,24	12,15	90,14	11,62	88,43
1,6	**10,97**	**93**	13,23	93,57	13,15	93,14
1,8	9,46	91	**12,46**	**93,43**	14,54	93,86
2	7,46	84,43	11,23	92,29	13,85	93,86
2,2	4,79	72,38	9,00	89,43	**11,62**	**93,43**
2,4	2,72	54,43	6,46	78,43	9,85	91,29
2,6	1,00	29,14	3,23	62,86	8,46	87,71
2,8	0,21	19,86	1,23	40,86	5,62	79,43
3	0,08	12,86	0,38	23,29	3,38	67,43

Table 6.8: *Training phase on DB2000 dataset: TPR and FPR values (in percentage) for each metric with respect to T_h.*

The dataset DB2000 is composed, as already said, by 2000 images (JPEG quality factor equal to 100) subdivided in 1300 originals and 700 tampered images. The tampered images have been created by using 50 images undergone to 14 different transformations (see Table 6.2). First of all, we have tried to set up again an experiment for the determination of the best threshold T_h, according to the three linkage methods, as done in subsection 6.3.1 for the DB220; this has been made to further check if the established thresholds were correct. To do that 1500 images are used for the training phase (975 original and 525 tampered) and the remaining 500 (325 original and

175 tampered) are used for the test phase; a 4-fold repetitions test has been carried out and, as before, averaged results are listed in Table 6.8. It can be observed that a similar behavior to that obtained with DB220 is registered and, above all, that the values chosen in subsection 6.3.1 for T_h (1.6 for *Single*, 1.8 for *Centroid* and 2.2 for *Ward's*) still grant about the higher performances in terms of TPR and FPR. After this, the test phase is launched by setting such values for T_h and in Table 6.9 the detection rates are reported demonstrating both the effectiveness of the proposed method which achieves a TPR around 93% for all the three metrics and its robustness obtaining again performances very coherent to those presented in Table 6.8 for these fixed thresholds.

	Single	*Centroid*	*Ward's*
FPR (%)	10.99	12.45	11.61
TPR (%)	92.99	93.23	93.42

Table 6.9: *Test phase on DB2000 dataset: detection results in terms of FPR and TPR obtained with $T_h = 1.6$, $T_h = 1.8$ and $T_h = 2.2$ for the three linkage methods respectively.*

Going into detail, in Figure 6.5 the number of errors for each attack is listed with regard to tampered images not detected as such. The most critical attacks seem to be the f ($\theta = 0°$, $s_x = 2$ and $s_y = 2$) and the n ($\theta = 40°$, $s_x = 1.1$ and $s_y = 1.6$) which increase twice the patch dimension and apply a 40 degrees rotation combined with a consistent variation on scale respectively. The histogram in Figure 6.5 shows that these two kinds of attacks generate everyone around the 30% of the total errors.

In Table 6.10 are then reported the estimate errors for the geometric transformation parameters averaged on all the 500 test images. The Mean Absolute Error (MAE) still remains small enough although the transformations applied to the images in this circumstance for DB2000 dataset are more challenging with respect to the case of DB220 dataset.

MAE(t_x)	MAE(t_y)	MAE(θ)	MAE(s_x)	MAE(s_y)
22.49	8.49	1.55	0.27	0.2

Table 6.10: *Transformation parameters estimation errors. The values t_x and t_y are expressed in pixels while θ in degrees.*

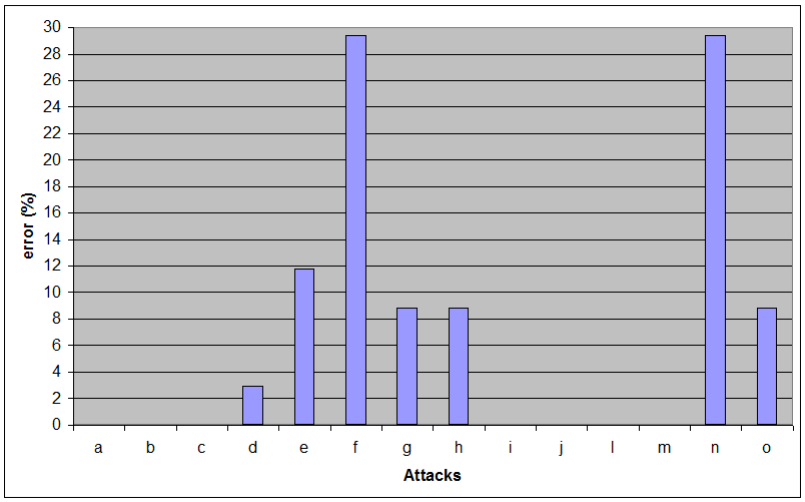


Figure 6.5: *Error analysis of tampered images mis-detection for each different attack (in percentage).*

JPEG compression and noise addition

The proposed methodology has also been tested in terms of detection performances from a robustness point of view; in particular, the impact of JPEG compression and then of noise addition on all the 2000 images of the DB2000 dataset has been investigated. In the first experiment all the images which were originally in the JPEG format (quality factor of 100), have been compressed in JPEG format with a decreasing quality factor of 75, 50, 40 and 20. In Table 6.11 (top) the FPR and TPR (*Ward's* linkage method with $T_h = 2.2$) for all the diverse JPEG quality factors are presented; it can be seen that FPR is practically stable while the TPR tends to slightly diminish when image quality decreases. In the second experiment, in the same way as before, the images of DB2000 dataset are distorted by adding a Gaussian noise obtaining different final decreasing signal-noise-ratios (SNR) of 50, 40, 30 and 20 db, that is noisy images are obtained by adding white Gaussian noise to the image with a JPEG quality factor of 100. In Table 6.11 (bottom), obtained results are shown and it can be noticed that the TPR is over 90% till a SNR of 30 dB while FPR is again quite stable, though it seems to even improve.

JPEG quality	FPR	TPR
100	11.61	93.42
75	12.07	93.42
50	11.15	93.16
40	11.38	92.14
20	10.46	87.15
SNR (dB)	FPR	TPR
50	11.46	93.71
40	11.69	94.14
30	11.46	92.00
20	8.15	82.42

Table 6.11: *Detection performances against JPEG compression (top) and noise addition (bottom)*

.

JPEG compression, noise addition, gamma correction on copied patch

The duplicated patch are often modified by applying some further processing such as brightness/contrast adjustment, gamma correction, noise addition and so on, in order to adjust the patch with respect to the image area where it has to be located. So to explore this scenario the following experiment has been made. Starting from 10 original images, a block is randomly (as explained before) selected for each of them and 4 geometric transformations (a, d, j and o from Table 6.2) are applied to every of these patches. Furthermore, before pasting them, 4 different gamma corrections with values [2.2, 1.4, 0.7, 0.45] are applied to each single block. Finally, 160 tampered images are obtained. In the same way, the final stage of gamma correction is firstly substituted by JPEG compression with different quality factors [75, 50, 40, 20] and secondly by Gaussian noise addition with SNR (dB) equal to 50, 40, 30, 20. For every case, 160 fake images have been created. So for each of the three situations (gamma correction, JPEG compression and noise addition), a dataset composed by 160 fake images and by 350 original ones randomly taken from the DB2000 database is built. Hereafter, in Table 6.12, performances in terms of TPR and FPR are reported.

These experiments show that the proposed method maintains its level

Kind of processing	FPR	TPR
Gamma correction	9.23	99.37
JPEG	11.38	100.00
SNR (dB)	12.00	100.00

Table 6.12: *Detection performances against gamma correction, JPEG compression and noise addition applied to the duplicated and geometrically transformed patch.*

of accuracy though some diverse kinds of post-processing are applied to the duplicated patch in addition to a geometric transformation, to adapt it to the image context where it is pasted.

Image splicing

Though the proposed technique has been presented to operate in a copy-move attack scenario, it can also be utilized in a context where a splicing operation has occurred. With the term splicing attack is intended that a part of an image is grabbed and, possibly after having been adapted (geometric transformed and/or enhanced), pasted onto another one to build a new fake image. In most of the cases only the final fake photo is available to the forensic analyst for inspection, the source one is often undeterminable; because of this, the SIFT matching procedure, which is the core of the proposed method could not take place and would seem that there is no room for it in such circumstance. Anyway this is not always true in practice! In fact, often, the analyst is required to give an assessment over a dataset of images for example belonging to a specific person under judgement, or that have been found in a hard disk or a pen drive, and so on. In this operative scenario, it can happen that the source image used to create a fraudulent content belongs to the image collection at disposal. It is easy to understand that the proposed method can be adopted again to determine both if within the to-be-checked collection there is a false image containing an "external" patch and, above all, where it comes from. It is interesting to highlight that succeeding in detecting such link could help investigation activities. To prove that the proposed technique can be used in such a scenario the following experimental test has been set up.

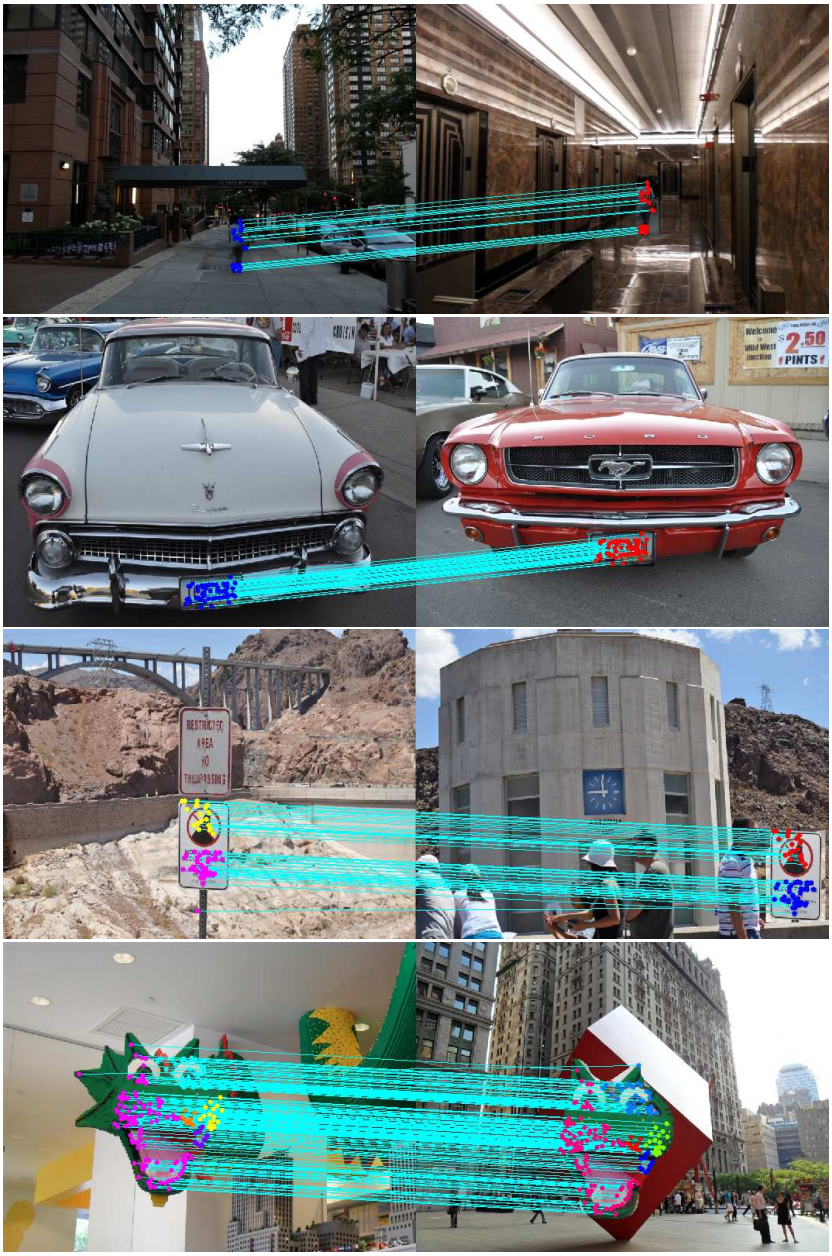


Figure 6.6: *Examples of correct detection of splicing attack.*

Figure 6.7: *An example of wrong detection of splicing attack.*

A subset of 100 images (96 original and 4 tampered with) taken from a private collection with size of 800×600 pixels has been selected. In particular, the 4 fake images have been created by pasting a patch that was cut from another image belonging to the other original 96. The proposed technique has been launched to analyze all the possible pairs of photos ($\frac{n\cdot(n-1)}{2} = \frac{100\cdot 99}{2} = 4950$) within the dataset looking for duplicated areas. To allow to the presented algorithm to perform as it is the pair of images to be checked are considered as a single image with a double number of columns (size equal to $N \times 2M$); due to this fact, the detection threshold T_h has been moved up to 3.4 (it was 2.2 in the previous experimental tests of this Section) for the *Ward's* linkage method which was chosen for this specific experiment. In Table 6.13 performances on FPR and TPR are reported.

Splicing attack	FPR (%)	TPR (%)
	0.04	100.00

Table 6.13: *Detection performances against splicing attack (in percentage).*

The method is able to correctly reveal all the four fake pairs as expected determining a link between the possible original image and the forged one, though it can not distinguish the source from the destination as well-known if other tools are not adopted. The procedure also detect as suspected two other innocent couples of images incurring in false alarms. In Figure 6.6 the four cases of splicing attack detection are pictured, while in Figure 6.7 one of the false alarm is illustrated. In this last circumstance, it is immediate to understand that the error is induced by the presence of the same objects (the posters over the wall of the wooden box) in both the photos taken in the same real context. However this could be the actual situation that might happen in

practical scenario (e.g. establishing possible relations among photos acquired in similar environments).

Chapter 7

Temporal forensics

In a number of forensic applications could be not sufficient to identify the imaging device but could be useful providing a temporal localization of the image. For example, consider a situation where a digital camera changed ownership and some images are found to be associated with illicit content. In this situation an estimate of the time when the images with illicit content were captured can help the investigation to establish a connection between the camera and the owner.
In order to capture the temporal characteristics of a multimedia device it is necessary model the temporal evolution of the sensor device considering the following sensor output model:

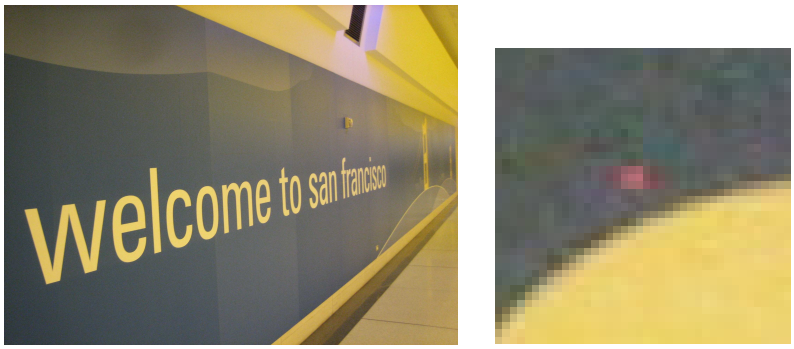
$$Y = I + IK + \theta \tag{7.1}$$

In the model Y is the output from the sensor, I is the input, IK is the term responsible for the PRNU and θ is a collection of all the others sensor noise. According to information avalaible in literature, the PRNU factor is quite stable in time so it is suitable for camera identification (2.2.2) but not for determining an approximate age of images. In the model (7.1) other systematic sensor artifacts exist, they are called Fixed Pattern Noise (FPN) that evolve in time and could be potentially used for this task. In particular defects in the FPN are the following:

- *Stuck pixel*. This type of pixel is always on and it is independent of the incident light, exposure, and other camera settings, and is usually easily visible in photographs (Figure 7.1). This means that any given pixel will stay red, blue or green, and will not change when attempting

to display an image. $y = c$, where c is a constant independent of exposure time or light intensity.

- *Partially Stuck pixel.* $y = I[p] + c$. The response of pixel p is offset by a constant when compared to a properly working pixel.
- *Hot pixel.* Hot pixels are defects due to leakage of electrons. Their output increases with the exposure in time, temperature and ISO settings. $y = I[p] + D[p]\tau + c$ where τ is the exposure time and $D[p]$ represents the effect of dark current integrated at the output of that particular pixel.



(a) (b)

Figure 7.1: *A stuck pixel.*

In particular pixel defects (Figure 7.2) occur randomly in an image and independently of each other. The main cause of new defects is the environmental stress and primarily cosmic ray radiation. In fact the pixels change faster at high altitudes or during airplane trips where cosmic radiation is stronger. Furthermore once defect occurs it becomes a permanent part of the sensor.

7.1 Identification of defective pixels

The first task to accomplish in order to temporally localize images in time is the identification of defective pixels. Then a mathematical model is exploited and the temporal information is extracted from the image and the localization is accomplished. It's worth to point out that in the following

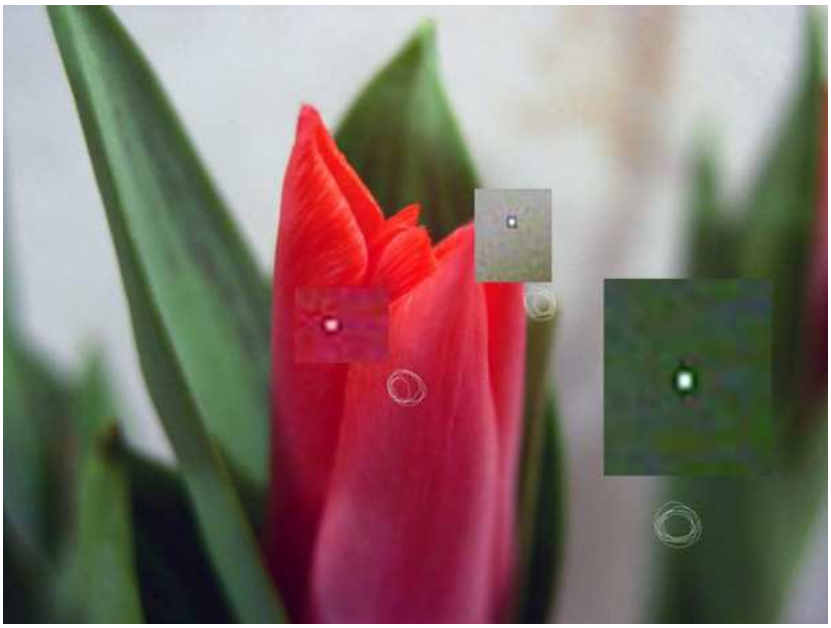


Figure 7.2: *Defective pixels.*

only some cues are given to solve the problem because the argument is still under investigation.
In general fully stuck or partially stuck pixels are extracted through laboratory calibration considering RAW data and subjecting the camera to uniform illumination at increasing intensities with short exposure duration. Hot pixels are revealed, instead, subjecting the camera to darkfield calibration, under no illumination. These tests repeated after few months revealed additional hot and stuck pixels in the defects map of a digital camera sensor while the behaviour of the initial defects remained about the same. In [99] is proposed a defect tracing algorithm that utilizes Bayesian statistics to automatically detect the presence and absence of defects by searching through an image set. At the beginning of the defect detection, the detector is provided with the defect map which specifies the i,j location of the pixel defects as explained before.

In image forensics, it is very rare to operate in a supervised environment so it's impossible to create a defects maps of a digital camera sensor because the camera could not be available or in other case there is not enough time to build a defects maps (sometimes months). On the other side , due to the argument complexity, has been analysed only a particular scenario of the

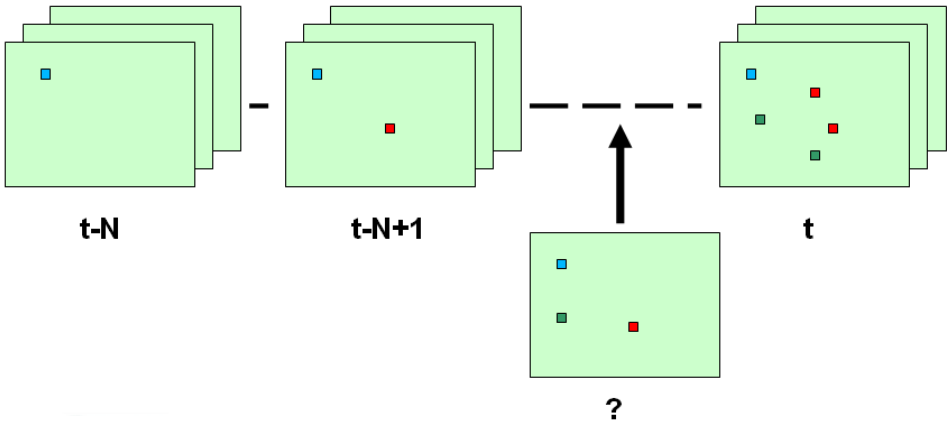


Figure 7.3: *The scenario.*

problem assuming a preordered image set available to the analyst and the objective is to localize one image respect to the others (Figure 7.3).

On the basis of the paper in [99], it is possible to determine the presence of candidate defects (especially hot pixel) in the following manner. A good pixel will measure the amount of incident light that strikes the pixel. The output of a defective pixel from each image capture is simplified to Equation 7.2, where I[p] represents the incident light at the pixel, and the others components can be treated as a dark offset denoted by Δ.

$$y = I[p] + \Delta \tag{7.2}$$

To determine the presence of defects, we first interpolate everywhere in the image to estimate the expected pixel value base on its neighbors values. Then, we compare each pixel's actual value with the interpolated value to find if the deviation is caused by the presence of a defect, or is the result of an interpolation error. Let z denote the interpolated pixel value and y denote the actual pixel output value.
Then, the error is computed as $e = y - z$, and the function $pE(e)$ is the Probability Distribution Function (PDF) of the interpolation error over the whole image (it will differ from one image to another). When y is the output of a good pixel, the error is due to interpolation error and should be approximately zero with a good interpolation scheme. On the other hand, if y is a hot pixel, the error will be approximately Δ.

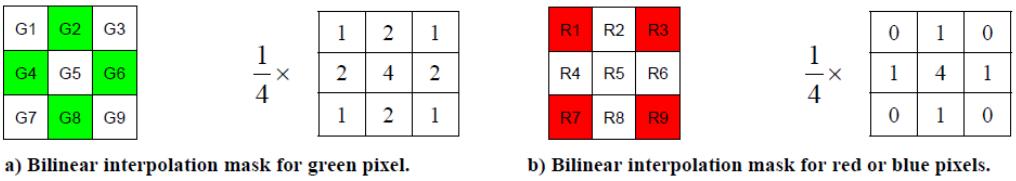


Figure 7.4: *Bayer Pattern.*

Most image sensors use a Color Filter Array (CFA) (most CFAs use the Bayer pattern) such that each pixel only captures one of three color channels: red, green, or blue.
The demosaicing algorithm is applied to the color matrix to recover the two missing color channels at each pixel. Because most demosaicing algorithms require interpolation to recover the missing channels, the presence of defects will contribute a false estimate to the interpolation.
For this study, will model the demosaicing algorithm with the simple bilinear interpolation using the filter masks shown in Figure 7.4. Each interpolation will replace the missing channel by the average of the 4 nearest neighboring pixels from the same color channel.

At this point we select from PDF of the interpolation error over the whole image the pixel value for which

$$pE(e) > abs(m) + 2 * sigma \tag{7.3}$$

where m and $sigma$ are respectively the PDF average and the standard deviation.

After that the selected pixel are sorted by error value and the first k elements with bigger error value will be taken in account.

This selection procedure is applied to multiple images belonging to the first part of the set (older images) and the last part of the set (newer images). Through a voting procedure a set of defective pixel is selected for each group and then two set of pixels are compared and the presence of the same pixel is evidenced. Furthermore the same pixels are deleted and then the others are selected as defective pixels changing in time.
Another method has been studied to identifying defective pixel, in particular stuck pixels. The luminance variation of each pixel in the image has been considered. Give a certain pixel (i, j) the luminance is a mono-dimensional signal in time $I(i, j, t)$. Applying the DFT to the mono-dimensional signal we should obtain no frequency variation if the pixel under observation is a

stuck pixel since the stuck pixel gives the same output for each image under all illuminations.
After identifying defective pixel with both of these technique the idea is to implement a maximum likelihood estimator of FPN in order to describe, given a set of images, the evolution in time of each pixel from no defective to defective pixel. Then it could be possible to place an image in a sequence knowing the behaviour of each pixel in time. [1]

[1] This is a preliminary work done in collaboration with the Professor J. Fridrich at the Binghamton University (USA) during the exchange PhD visiting program from April to October 2010.

Chapter 8

Conclusion and Open Issue

Nowadays, digital visual data have gained high relevance in nearly every aspect of our life and represent one of the main source of information that can bias common opinion. In particular scenarios, such as the forensic one, visual information can be used as possible evidence in a trial thus influencing the final verdict. In such a situation, it is fundamental to know the origin and the history of such data in order to be assured that opinion coming from such information has not been manipulated. In the last years, a new science, referred as multimedia forensics, has been proposed aiming at providing information on a digital asset, by means of the analysis of intrinsic fingerprints that characterize the data during its life. In particular, the analysis of these patterns may lead to identify image and video origin and to establish data integrity.

In this thesis, principles and motivations of digital forensics have been discussed and new methods in Image Forensics have been presented. All the proposed techniques can be sketched as a forensic tool that extracts, from the considered data, some digital fingerprints, and that, by exploring some properties of such patterns, is able to make a decision based on either classification or estimation procedure. In particular, the output of such a tool can provide information on the acquisition device that has produced the visual content as well as on the possible suffered tampering.

The research community is showing an increasing interest for such technologies thus leading to new exciting challenges for the solution of many open issues in the next future.

One of these for example is to create a common framework for multimedia forensics because many of the existing digital forensic techniques are

bright and groundbreaking but none of them by itself offers a stand alone solution for the considered problem (i.e. the source identification and the verification of information integrity). Furthermore, the user intervention is often desirable for validating the final results: for example, let us consider the estimation of image tampering, that without any user intervention is quite impossible, since even if an out camera processing is detected, often only a human interpreter can decide if the purpose of the modification is malicious or not.

The validation of digital forensic approaches for integrity verification, seems to be missing of a common framework, regarding both image databases and performance measures, such as accuracy, robustness, security.

An image database is fundamental for the evaluation of a proposed algorithm; furthermore, a common dataset provides an unified platform for the research community to compare various algorithms. Actually, several datasets are available for the research community [98], but there are some open issues that call for a benchmark dataset. For instance, the experiments involving the camera characteristics require a dataset of images acquired by a diverse models of camera, at various acquisition settings. Furthermore, in order to facilitate the evaluation of the image forgery detection techniques using the images produced by the state-of-the-art image forgery creation techniques, a dataset of these images would be necessary. Therefore, further effort on producing and standardizing the additional benchmark dataset is needed.

A different analysis on performances of forensic algorithms comes from the security point of view. By increasing the possible solutions for forgery identification, also malevolent people, aiming at modifying digital content, increase their attention for overcoming detection of tampering processing. Hence, deepened the analysis of forensic algorithms from the security point of view would be an interesting open issue to be addressed in the future.

Another future trend to be considered is the improvement of the use of image source imperfections as fingerprint to solve the problem of source identification. Review of the modern literature on this argument shows that good experimental results are obtained but reliable identification seems impossible if all the acquisition process and post-processing steps are not taken into account, so further investigations are necessary. Future research should focus on definition of new model for the acquisition process in order to better estimate the anomalies left by intrinsic disconformities in the manufacturing

process of silicon sensor of a camera. Since this fingerprint is not a random noise but a deterministic template, which is superimposed to each taken image, should be necessary to define and use new denoising filters that grant the suppression of the image content and take into account the different kind of sensor device.

Another interesting topic to investigate is the "temporal forensic", described in Section 7), not yet completely addressed by scientific community.

Chapter 9

Publications

International Journals

- I. Amerini, G. Ballocca, R. Becarelli, R. Borri, R. Caldelli, F. Filippini, "A DVB-MHP web browser to pursue convergence between Digital Terrestrial Television and Internet", in Multimedia Tools and Applications, 2009.

- R. Caldelli, I. Amerini, F. Picchioni, "A DFT-based analysis to discern between camera and scanned images", in International Journal of Digital Crime and Forensics (IJDCF), Volume 2, Number 1, e-Forensics 2009 Special Edition, Jan-Mar 2010

- I. Amerini, R. Caldelli, V. Cappellini, F. Picchioni, A. Piva, "Estimate of PRNU noise based on different noise models for source camera identification", in International Journal of Digital Crime and Forensics (IJDCF), Volume 2, Number 2.

International Conferences

- R. Caldelli, I. Amerini, and F. Picchioni, "Distinguishing between camera and scanned images by means of frequency analysis," in Proceedings of e-Forensics 2009: The International Conference on Forensic Applications and Techniques in Telecommunications, Information and

Multimedia, Adelaide, South Australia, January 19-21, 2009. [best paper]

- I. Amerini, R. Becarelli, R. Ballocca, R. Borri, R. Caldelli, F. Filippini, "Integration between Digital Terrestrial Television and Internet by means of a DVB-MHP web browser", 5th International Conference on Web Information Systems and Technologies, Lisboa, Portugal, 23-26 March, 2009.

- R. Caldelli, I. Amerini, F. Picchioni, V. Cappellini, "Multimedia forensics: technologies for imagery investigation", EVA 2009 Florence, pp.87-92, Florence, Italy, 28-30 April 2009.

- I. Amerini, R. Caldelli, V. Cappellini, F. Picchioni, and A. Piva, "Analysis of Denoising Filters for Photo Response Non Uniformity Noise Extraction in Source Camera Identification", 16th Int. Conference on Digital Signal Process, Santorini, Greece, July 5-7, 2009.

- I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo, G. Serra, "Geometric tampering estimation by means of a SIFT-based forensic analysis", International Conference in Acoustic Speech and Signal Processing, Dallas TX, USA, March 14-19, 2010.

- R. Caldelli, I. Amerini, F. Picchioni, and M. Innocenti, "Fast Image Clustering of unknown source images", International Workshop on Information Forensics and Security, Seattle, December 12-15, 2010.

Book chapters, technical reports

- R. Caldelli, I. Amerini, F. Picchioni, A. De Rosa, F. Uccheddu, "Multimedia forensic techniques for acquisition device identification and digital image authentication" , book chapter in Handbook of Research on Computational Forensics, Digital Crime and Investigation: Methods and Solutions, IGI Global, Hershey, PA. USA.November 2009.

Acknowledgements

Un grazie a tutte le persone che ho avuto l'oppurtunitá di conoscere durante questi tre anni di dottorato.
Grazie alla mia famiglia, a Clarissa e a Gian Marco.
Un grazie a tutti gli amici del LCI-MICC passati e presenti e un ringraziamento particolare a Roberto Caldelli per avermi seguita e spronata durante questi tre anni di ricerca.

Bibliography

[1] M. Mihcak, I. Kozintsev, and K. Ramchandran, "Spatially adaptive statistical modeling of wavelet image coefficients and its application to denoising," in *Proc. IEEE ICASSP*, vol. 6, 1999, pp. 3253–3256.

[2] F. Argenti, G. Torricelli, and L. Alparone, "Mmse filtering of generalised signal-dependent noise in spatial and shift-invariant wavelet domains," *Signal Process Journal*, vol. 86, no. 8, pp. 2056–2066, 2006.

[3] "A road map for digital forensics research," DFRWS, Tech. Rep., 2001.

[4] E. Casey, *Digital Evidence and Computer Crime, Second Edition.* Elsevier, 2004. [Online]. Available: ISBN0-12-163104-4

[5] R. Bohme, F. Freiling, T. Gloe, and M. Kirchner, "Multimedia forensics is not computer forensics," in *Third International Workshop on Computational Forensics*, Z. J. Geradts, K. Y. Franke, and C. J. Veenman, Eds., 2009, pp. 90–103.

[6] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography.* CRC Press, 2001. [Online]. Available: http://www.cacr.math.uwaterloo.ca/hac/

[7] P. B. J. Fridrich, "Secure digital camera," in *Digital Forensic Research Workshop*, 2004.

[8] A. C. Popescu and H. Farid, "Statistical tools for digital forensics," in *Proc. of IHW*, 2006.

[9] M. A. Swaminathan and K. J. R. Liu, "Non-intrusive forensics analysis of visual sensors using output images," in *Proc. of IEEE ICIP*, 2006.

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

[11] M. Chen, J. Fridrich, M. Goljan, and J. Lukas, "Determining image origin and integrity using sensor noise," *IEEE Trans. on Information Forensics and Security*, vol. 3(1), pp. 74–90, 2008.

[12] J. F. M. Chen and J. L. M. Goljan, "Source digital camcorder identification using sensor photo response non-uniformity," in *SPIE*, 2007.

[13] N. Mondaini, R. Caldelli, A. Piva, and V. C. M. Barni, "Detection of malevolent changes in digital video for forensic applications," in *Proc. SPIE*, vol. 6505, 65050T, 2007.

[14] N. Khanna, A. Mikkilineni, G. T.-C. Chiu, J. P. Allebach, and E. J. Delp, "Scanner identification using sensor pattern noise," in *Proc. SPIE*, vol. 6505, 65051K, 2007.

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

[16] K. Cohen, "Digital still camera forensics," *SMALL SCALE DIGITAL DEVICE FORENSICS JOURNAL*, vol. '1 no. 1.

[17] H. T. S. S. Bayram and N. Memon, "Source camera identification based on cfa interpolation," in *Proc. of IEEE ICIP*, 2005.

[18] S. Bayram, S. H. T., and N. Memon, "Improvements on source camera-model identification based on cfa interpolation," in *Proc. of WG 11.9 Int. Conf. on Digital Forensics*, 2006.

[19] G. Healey and R. Kondepudy, "Radiometric ccd camera calibration and noise estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 3, pp. 267–276, 1994.

[20] M. K. K. K. K. K. Z. J. Geradts, J. Bijhold and N. Saitoh, "Methods for identification of images acquired with digital cameras," in *Proc. of SPIE*, 2001.

[21] K. Kurosawa, K. Kuroki, and N. Saitoh, "Ccd fingerprint method-identification of a video camera from videotaped images," in *ICIP*, 1999.

[22] J. F. J. Lukas and M. Goljan, "Digital camera identification from sensor pattern noise," *IEEE Trans. Inf. Forensics and Security*, vol. 1 no. 2, pp. 205–214, 2006.

[23] M. Goljan and J. Fridrich, "Camera identification from scaled and cropped images," in *SPIE Conference on Security, Forensics, Steganography, and Watermarking of Multimedia Contents*, E. J. Delp and P. W. Wong, Eds., vol. 6819, 2008.

[24] H. T. S. S. Dehnie and N. Memon, "On discrimination between photorealistic and photographic images," in *Proc. of IEEE ICIP*, 2006.

[25] G. T. C. C. J. P. A. N. Khanna, A. K. Mikkilineni and E. J. Delp, "Forensic classification of imaging sensor types," in *Proc. of SPIE*, 2007.

[26] J. F. M. Chen and M. Goljan, "Digital imaging sensor identification (further study)," in *Proc. of SPIE*, 2007.

[27] E. Y. L. K. S. Choi and K. K. Y. Wong, "Source camera identification using footprints from lens aberration," in *Proc. of SPIE*, 2006.

[28] M. S. K. Lanh Tran Van, Sabu Emmanuel, "Identifying source cell phone using chromatic aberration," in *IEEE International Conference on Multimedia and Expo*, 2007.

[29] H. T. S. M. Kharrazi and N. Memon, "Blind source camera identification," in *Proc. of IEEE ICIP*, 2004.

[30] B. S. O. Celiktutan, I. Avcibas and N. Memon, "Source cell-phone identification," in *Proc. of ADCOM*, 2005.

[31] I. B.Sankur, O. Celiktutan, "Blind identification of cell phone cameras," in *SPIE*, 2007.

[32] Y. Wang and P. Moulin, "On discrimination between photorealistic and photographic images," in *Proc. of IEEE ICASSP*, 2006.

[33] H. T. S. E. Dirik, S. Bayram and N. Memon, "New features to identify computer generated images," in *Proc. of IEEE ICIP*, 2007.

[34] S. Lyu and H. Farid, "How realistic is photorealistic?" *IEEE Trans. On Signal Processing*, vol. vol. 53, no. 2, pp. pp. 845–850, 2005.

[35] H. Gou, A. Swaminathan, and M. Wu, "Robust scanner identification based on noise features," in *Proc. SPIE*, vol. 6505, 65050S, 2007.

[36] C. McKay, A. Swaminathan, H. Gou, and M. Wu, "Image acquisition forensics: Forensic analysis to identify imaging source," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Apr. 2008, pp. 1657–1660.

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

[38] J. F. J. Lukas and M. Goljan, "Detecting digital image forgeries using sensor pattern noise," in *Proc. of SPIE*, 2006.

[39] M. K. Johnson and H. Farid, "Exposing digital forgeries through chromatic aberration," in *Proc. of ACM Multimedia Security Workshop*, 2006.

[40] M.-P. T. Tian-Tsong Ng, Shih-Fu Chang, "Camera response function estimation from a single-channel image using differential invariants," Columbia University, Tech. Rep. 216-2006-2,, 2006.

[41] X. T. Z. Lin, R.Wang and H. Y. Shum, "Detecting doctored images using camera response normality and consistency analysis," in *Proc. of CVPR*, 2005.

[42] Y.-F. Hsu and S.-F. Chang, "Detecting image splicing using geometry invariants and camera characteristics consistency," in *ICME*, 2006.

[43] J. Fridrich, M. Goljan, and R. Du, "Steganalysis based on jpeg compatibility," A. G. Tescher, B. Vasudev, V. M. Bove, and Jr., Eds., vol. 4518, no. 1. SPIE, 2001, pp. 275–280. [Online]. Available: http://link.aip.org/link/?PSI/4518/275/1

[44] L. W. J. He, Z. Lin and X. Tang, "Detecting doctored jpeg images via dct coefficient analysis," in *Proc. of ECCV*, 2006.

[45] T. T. Ng, "Statistical and geometric methods for passive-blind image forensics," Ph.D. dissertation, New York, NY, USA, 2007, adviser-Chang, Shih-Fu.

[46] A. C. Popescu and H. Farid, "Exposing digital forgeries by detecting traces of resampling," *IEEE Trans. Signal Processing*, vol. vol. 53, no. 2, pp. 758–767, 2005.

[47] A. Gallagher, "Detection of linear and cubic interpolation in jpeg compressed images," may. 2005, pp. 65 – 72.

[48] B. Mahdian and S. Saic, "Detection of copy-move forgery using a method based on blur moment invariants," *Forensic Science International*, vol. 171, no. 2-3, pp. 180–189, 2007.

[49] W. Luo, J. Huang, and G. Qiu, "Robust detection of region-duplication forgery in digital image," in *Proc. of ICPR*, Washington, D.C., USA, 2006.

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

[51] 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*, E. J. Delp and P. W. Wong, Eds., vol. 6505, 2007.

[52] S. F. C. T. Ng and Q. Sun, "Blind detection of photomontage using higher order statistics," in *Proc. of ISCAS*, 2004.

[53] T. Ng and S. F. Chang, "A model for image splicing," in *Proc. of ICIP*, 2004.

[54] W. Chen, Y. Q. Shi, and W. Su, "Image splicing detection using 2-d phase congruency and statistical moments of characteristic function," E. J. D. III and P. W. Wong, Eds., vol. 6505, no. 1. SPIE, 2007, p. 65050R. [Online]. Available: http://link.aip.org/link/?PSI/6505/65050R/1

[55] J. Fridrich, D. Soukal, and J. Lukás, "Detection of copy-move forgery in digital images," in *Proc. of DFRWS*, 2003.

[56] A. Popescu and H. Farid, "Exposing digital forgeries by detecting duplicated image regions," Dartmouth College, Computer Science, Tech. Rep. TR2004-515, 2004.

[57] H. Farid, "Blind inverse gamma correction," *IEEE Transactions on Image Processing*, vol. 10, no. 10, pp. 1428–1433, 2001. [Online]. Available: www.cs.dartmouth.edu/farid/publications/ip01.html

[58] A. Swaminathan, M. Wu, and K. J. R. Liu, "Image tampering identification using blind deconvolution," in *IEEE International Conference on Image Processing*, 2006, pp. 2309–2312.

[59] H. Farid, "Detecting digital forgeries using bispectral analysis," AI Lab, Massachusetts Institute of Technology, Tech. Rep. AIM-1657,, 1999. [Online]. Available: www.cs.dartmouth.edu/farid/publications/tr99.html

[60] M. K. Johnson and H. Farid, "Exposing digital forgeries by detecting inconsistencies in lighting," in *Proc. of ACM Multimedia Security Workshop*, 2005.

[61] ——, "Exposing digital forgeries in complex lighting environments," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 2, pp. 450–461, 2007. [Online]. Available: www.cs.dartmouth.edu/farid/publications/tifs07a.html

[62] ——, "Exposing digital forgeries through specular highlights on the eye," in *International Workshop on Information Hiding*, 2007, pp. 311–325. [Online]. Available: www.cs.dartmouth.edu/farid/publications/ih07.html

[63] ——, "Detecting photographic composites of people," in *International Workshop on Digital Watermarking*, 2007, pp. 19–33. [Online]. Available: www.cs.dartmouth.edu/farid/publications/iwdw07.html

[64] T. Gloe, M. Kirchner, A. Winkler, and R. Bohme, "Can we trust digital image forensics?" in *International Conference on Multimedia*, 2007, pp. 78–86.

[65] M. Kirchner and R. Bohme, "Tamper hiding: Defeating image forensics," in *International Workshop on Information Hiding*, T. Furon, F. Cayre, G. Doerr, and P. Bas, Eds., 2007, pp. 326–341.

[66] ——, "Hiding traces of resampling in digital images," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 4, pp. 582–592, 2008.

[67] M. Goljan, J. Fridrich, and M. Chen, "Sensor noise camera identification: Countering counter-forensics," in *SPIE Conference on Media Forensics and Security*, 2010.

[68] A. K. Jain, *Fundamentals of digital image processing.* Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1989.

[69] L. A. Gionatan Torricelli, Fabrizio Argenti, "Modelling and assessment of signal-dependent noise for image de-noising," in *EUSIPCO*, 2002.

[70] C.-T. Li, "Unsupervise classification of digital images enhanced sensor pattern noise," in *IEEE International Symposium on Circuits and Systems (ISCAS'10)*, 2010.

[71] M. Mihcak, I. Kozintsev, and K. Ramchandran, "Spatially adaptive statistical modeling of wavelet image coefficients and its application to denoising," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing- Proceedings*, vol. 6, 1999, pp. 3253–3256.

[72] C.-T. Li, "Source camera identification using enhanced sensor pattern noise," in *Proceedings of IEEE International Conference on Image Processing*, 2009, pp. 7–11.

[73] Y. Sutcu, S. Bayram, H. Sencar, and N. Memon, "Improvements on sensor noise based source camera identification," in *2007 IEEE International Conference on Multimedia and Expo*, 2007, pp. 24–27.

[74] P. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining.* Addison Wesley, May 2005.

[75] H. Farid, "A survey of image forgery detection," *IEEE Signal Processing Magazine*, vol. 2, no. 26, pp. 16–25, 2009. [Online]. Available: www.cs.dartmouth.edu/farid/publications/spm09.html

[76] I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo, and G. Serra, "Geometric tampering estimation by means of a sift-based forensic analysis," in *Proc. of IEEE ICASSP*, Dallas, USA, 2010.

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

[78] S. Bayram, I. Avcibas, B. Sankur, and N. Memon, "Image manipulation detection with binary similarity measures," in *Proc. of European Signal Processing Conference (ESPC)*, 2005.

[79] H. Farid and S. Lyu, "Higher-order wavelet statistics and their application to digital forensics," in *Proc. of IEEE CVPR Workshop on Statistical Analysis in Computer Vision*, Madison, WI, USA, 2003.

[80] G. Li, Q. Wu, D. Tu, and S. J. Sun, "A sorted neighborhood approach for detecting duplicated regions in image forgeries based on DWT and SVD," in *Proc. of IEEE ICME*, Beijing, China, 2007.

[81] B. Dybala, B. Jennings, and D. Letscher, "Detecting filtered cloning in digital images," in *MM&Sec '07: Proceedings of the 9th workshop on Multimedia & security.* New York, NY, USA: ACM, 2007, pp. 43–50.

[82] S. Bayram, H. Taha Sencar, and N. Memon, "An efficient and robust method for detecting copy-move forgery," in *Proc. of IEEE ICASSP*, Washington, DC, USA, 2009.

[83] A. C. Popescu and H. Farid, "Exposing digital forgeries by detecting traces of resampling," *IEEE Transactions on Signal Processing*, vol. 53, no. 2, pp. 758–767, 2005.

[84] S. Bayram, H. T. Sencar, and N. Memon, "A survey of copy-move forgery detection techniques," in *IEEE Western New York Image Processing Workshop*, 2008.

[85] S.-J. Ryu, M.-J. Lee, and H.-K. Lee, "Detection of copy-rotate-move forgery using zernike moments," in *International Workshop on Information Hiding*, 2010.

[86] H.-J. Lin, C.-W. Wang, and Y.-T. Kao, "Fast copy-move forgery detection," *WSEAS Trans. Sig. Proc.*, vol. 5, no. 5, pp. 188–197, 2009.

[87] S. Bravo-Solorio and A. K. Nandi, "Passive method for detecting duplicated regions affected by reflection, rotation and scaling," in *European Signal Processing Conference*, 2009.

[88] V. Christlein, C. Riess, and E. Angelopoulou, "A study on features for the detection of copy-move forgeries," in *Information Security Solutions Europe*, 2010.

[89] X. Shuai, C. Zhang, and P. Hao, "Fingerprint indexing based on composite set of reduced sift features," in *Proc. of ICPR*, 2008.

[90] H. Su, A. Bouridane, and M. Gueham, "Local image features for shoeprint image retrieval," in *Proc. of BMVC*, 2007.

[91] H. Huang, W. Guo, and Y. Zhang, "Detection of copy-move forgery in digital images using sift algorithm," in *Proc. of IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, 2008.

[92] X. Pan and S. Lyu, "Detecting image region duplication using SIFT features," in *Proc. of IEEE ICASSP*, Dallas, USA, 2010.

[93] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.

[94] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, "A comparison of affine region detectors," *International Journal of Computer Vision*, vol. 65, no. 1/2, pp. 43–72, 2005.

[95] T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning*. Springer, 2003.

[96] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

[97] M. Fischler and R. 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.

[98] T.-T. Ng, S.-F. Chang, J. Hsu, and M. Pepeljugoski, "Columbia photographic images and photorealistic computer graphics dataset," ADVENT, Columbia University, Tech. Rep., 2004.

[99] J. Leung, J. Dudas, G. H. Chapman, Z. Koren, and I. Koren, "Characterization of pixel defect development during digital imager lifetime," M. M. Blouke and E. Bodegom, Eds., vol. 6816, no. 1. SPIE, 2008, p. 68160A. [Online]. Available: http://link.aip.org/link/?PSI/6816/68160A/1