Vision of the Unseen: Current Trends and Challenges in Digital Image and Video Forensics

ANDERSON ROCHA
University of Campinas, Brazil
WALTER SCHEIRER
University of Colorado at Colorado Springs, USA
TERRANCE BOULT
University of Colorado at Colorado Springs, USA
SIOME GOLDENSTEIN
University of Campinas, Brazil

Digital images are everywhere — from our cell phones to the pages of our online news sites. How we choose to use digital image processing raises a surprising host of legal and ethical questions that we must address. What are the ramifications of hiding data within an innocent image? Is this an intentional security practice when used legitimately, or intentional deception? Is tampering with an image appropriate in cases where the image might affect public behavior? Does an image represent a crime, or is it simply a representation of a scene that has never existed? Before action can even be taken on the basis of a questionable image, we must detect something about the image itself. Investigators from a diverse set of fields require the best possible tools to tackle the challenges presented by the malicious use of today's digital image processing techniques.

In this survey, we introduce the emerging field of digital image forensics, including the main topic areas of source camera identification, forgery detection, and steganalysis. In source camera identification, we seek to identify the particular model of a camera, or the exact camera, that produced an image. Forgery detection's goal is to establish the authenticity of an image, or to expose any potential tampering the image might have undergone. With steganalysis, the detection of hidden data within an image is performed, with a possible attempt to recover any detected data. Each of these components of digital image forensics is described in detail, along with a critical analysis of the state of the art, and recommendations for the direction of future research.

Categories and Subject Descriptors: K.4.0 [**Computers and Society**]: General; K.4.1 [**Computers and Society**]: Public Policy Issues; K.4.2 [**Computers and Society**]: Social Issues

General Terms: Security, Legal Aspects

Additional Key Words and Phrases: Image and Video Forensics, Forgery and Fraud Detection, Legal Aspects, Hidden Messages Detection, Image and Video Source Identification

Author's address: *Anderson Rocha* and *Siome Goldenstein* are with the *Reasoning for Complex Data Lab* (RECOD), Institute of Computing, University of Campinas, Av. Albert Einstein, 1251 – Campinas, São Paulo, Brazil, 13083-970.
Walter Scheirer and *Terrance Boult* are with the *Vision And Security Technology Lab* (VAST), University of Colorado at Colorado Springs, 1420 Austin Bluffs Parkway P.O. Box 7150, Colorado Springs, CO, USA, 80933-7150.

1. INTRODUCTION

With the advent of the Internet and low-price digital cameras, as well as powerful image editing software (such as Adobe Photoshop and Illustrator, and GNU Gimp), ordinary users have more access to the tools of digital doctoring than ever before. At the same time our understanding of the technological, ethical, and legal implications associated with image editing falls far behind. When such modifications are no longer innocent image tinkerings and start implying legal threats to a society, it becomes paramount to devise and deploy efficient and effective approaches to detect such activities [Popescu and Farid 2005a].

Digital Image and Video Forensics research aims at uncovering and analyzing the underlying facts about an image or video. Its main objectives comprise: tampering detection (cloning, healing, retouching, splicing), hidden data detection/recovery, and source identification with no prior measurement or registration of the image.

Though image manipulation is not new, its prevalence in criminal activity has surged over the past two decades, as the necessary tools have become more readily available, and easier to use. In the criminal justice arena, we most often find tampered images in connection with child pornography cases. The 1996 Child Pornography Prevention Act (CPPA) extended the existing federal criminal laws against child pornography to include certain types of "virtual porn". Notwithstanding, in 2002, the United States Supreme Court found that portions of the CPPA, being excessively broad and restrictive, violated First Amendment rights. The Court ruled that images containing an actual minor or portions of a minor are not protected, while computer generated images depicting a fictitious "computer generated" minor are constitutionally protected. However, with computer graphics, it is possible to create fake scenes visually indistinguishable from real ones [Ng and Chang 2009]. In this sense, one can apply sophisticated approaches to give more realism to the created scenes deceiving the casual eye and conveying a criminal activity. In the United States, a legal burden exists for "a strong showing of the photograph's competency and authenticity[1]" when such evidence is presented in court. In response, tampering detection and source identification are tools to satisfy this requirement.

Data hidden within digital imagery represents a new opportunity for traditional criminal activities. Most notably, the investigation of Juan Carlos Ramirez Abadia, a Columbian drug trafficker arrested in Brazil in 2008, uncovered voice and text messages hidden within images of a popular cartoon character [Herald Sun 2008; Folha de São Paulo 2008] on the suspect's computer. Similarly, a 2007 study[2] performed by Purdue University found data hiding tools on numerous computers seized in conjunction with child pornography and financial fraud cases. While a serious hinderance to a criminal investigation, data hiding is not a crime in itself; crimes can be masked by its use. Thus, an investigator's goal here is to identify and recover any hidden evidence within suspect imagery.

In our digital age, images and videos reach us at remarkable speed and frequency. Unfortunately, there are currently no established methodologies to verify their authenticity and integrity in an automatic manner. Digital image and video forensics

[1] Bergner v. State, 397 N.E.2d 1012, 1016 (Ind. Ct. App. 1979).

[2] `http://www.darkreading.com/security/encryption/showArticle.jhtml?articleID=208804788`

are still emerging research fields with important implications for ensuring the credibility of digital contents. As a consequence, on a daily basis we are faced with numerous images and videos — and it is likely that at least a few have undergone some level of manipulation. The implications of such tampering are only beginning to be understood.

In the following sections, we provide a comprehensive survey of the most relevant works with respect to this exciting new field of the *unseen* in digital imagery. Section 2 presents some brief historical remarks regarding image doctoring, including its impact on society. Section 3 is a thorough tour of the vision techniques for the forensics of the unseen, covering all the main areas of interest. Finally, in Section 4, we wrap up the survey and present some conclusions.

Throughout this survey, we emphasize approaches that we believe to be more applicable to forensics. Notwithstanding, most publications in this emerging field still lack important discussions about resilience to counter-attacks, which anticipate the existence of forensic techniques [Gloe et al. 2007]. As a result, the question of trustworthiness of digital forensics arises, for which we try to provide some positive insights.

2. HISTORICAL REMARKS

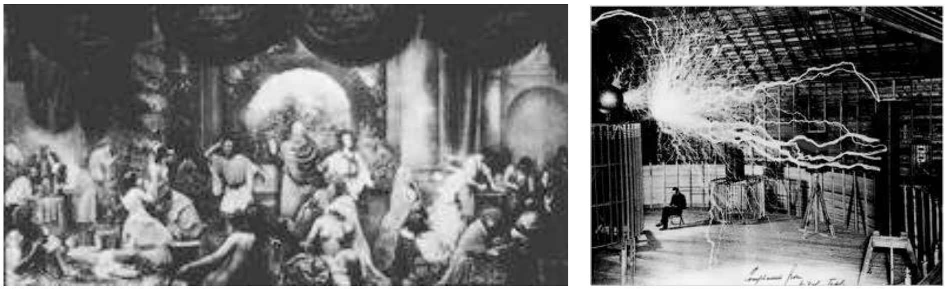
Image doctoring in order to represent a scene that never happened is as old as the art of the photograph itself. Shortly after the Frenchman Nicéphore Niepce [Coe 1990] created the first photograph in 1814[3], there were the first indications of doctored photographs. Figure 1(a) depicts one of the first examples of image forgery. The photograph, an analog composition of 30 images[4], is known as *The Two Ways of Life* and was created by Oscar G. Rejland in 1857. Figure 1(b) depicts another old example of analog montage. The image is a publicity photo taken in 1899 at Nikola Tesla's laboratory in Colorado Springs, Colorado, where the inventor worked before he established his Wardenclyffe laboratory on Long Island. The photo sports a double exposure — his pose and the sparks were recorded at different times, avoiding his electrocution [Willian J. Broad 2009].

In more recent times, we've seen an increase in questionable image processing. On March 31st, 2003, the Los Angeles Times showed on its front cover an image from photojournalist Brian Walski, in which a British soldier in Iraq stood trying to control a crowd of civilians in a passionate manner. The problem was that the moment depicted never happened (see Figure 2(a)). The photograph was a composite of two different photographs merged to create a more appealing image. The doctoring was discovered and Walski was fired.

In the 2004 presidential campaign, John Kerry's allies were surprised by a photomontage that appeared in several newspapers purporting to show Kerry and Jane Fonda standing together at a podium during a 1970s anti-war rally (see Figure 2(c)). As a matter of fact, the photograph was a fake. Kerry's picture was taken at an anti-war rally in Mineola, NY., on June 13th, 1971 by photographer Ken Light.

[3] Recent studies [Marien 2006] have pointed out that the photograph was, indeed, invented concurrently by several researchers such as Nicphore Niepce, Louis Daguerre, Fox Talbot, and Hercule Florence.

[4] Available in `http://www.bradley.edu/exhibit96/about/twoways.html`



(a) Oscar Rejland's analog composition, 1857. (b) Nikola Tesla's laboratory, 1899.

Fig. 1. Two early examples of analog compositions.

Fonda's picture was taken during a speech at Miami Beach, FL. in August, 1972 by photographer Owen Franken.

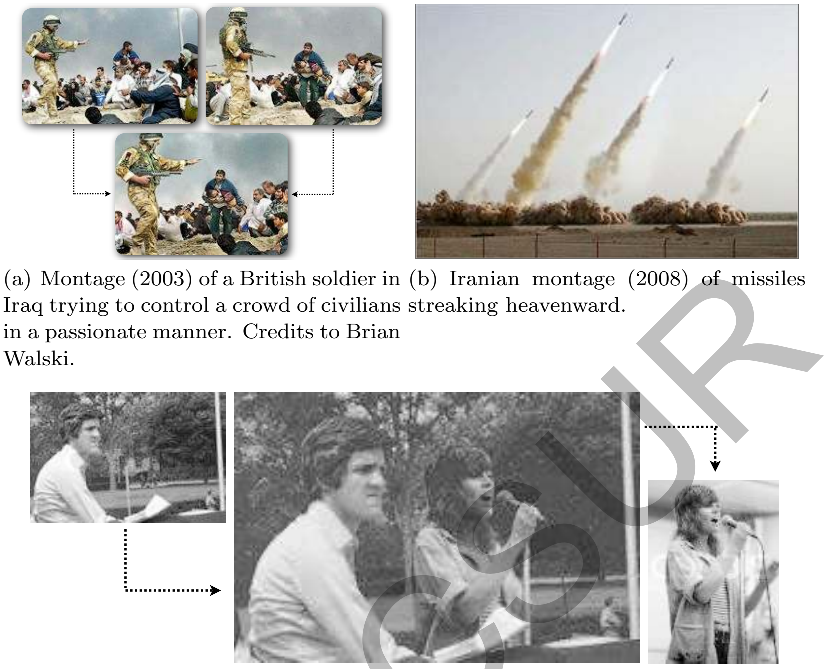
On April 5th, 2009, the Brazilian newspaper Folha de São Paulo published an article [Folha de São Paulo 2009] on how the Brazilian Chief of Staff at that point, Dilma Rousseff (a possible runner for the presidential office on the 2010 election), has actively participated in the resistance against the military regime, such as the planning and preparations of robberies and kidnappings. As part of the article, the newspaper printed an alleged image of the Repression Police Internal files of Secretary Rousseff stating that it came from the Public Archive of São Paulo, that houses the collection of documents from that period of time. A further analysis of the document pointed out that the object in question was a fabrication. The photograph in the document is the result of a splicing operation from a different grayscale image, the text is the result of a digitally-manipulated insertion, and it has not originated from a scanning procedure of a typewritten document [Goldenstein and Rocha 2009].

The scientific community has also been subject to forgeries. A particular case of scientific fraud involving doctored images in a renowned scientific publication has shed light to a problem believed to be far from the academy. In 2004, the South Korean professor Hwang Woo-Suk and colleagues published in *Science* important results regarding advances in stem cell research. Less than one year later, an investigative panel pointed out that nine out of eleven customized stem cell colonies that Hwang had claimed to have made involved doctored photographs of two other, authentic, colonies. Sadly, this is not a detached case. In at least one journal[5] [Pearson 2005], it is estimated that as many as 20% of the accepted manuscripts contain figures with improper manipulations, and roughly 1% with fraudulent manipulations [Farid 2006b; Pearson 2005].

Photo and video retouching and manipulation are also used for political purposes. On July 10th, 2008, various major daily newspapers published a photograph of four Iranian missiles streaking heavenward (see Figure 2(b)). Surprisingly, shortly after the photo's publication, a small blog provided evidence that the photograph had been doctored. The media, left in a somewhat embarrassing position, was forced to publish a plethora of retractions and apologies [Mike Nizza and Patrick Witty

[5] Journal of Cell Biology.

2008].



(a) Montage (2003) of a British soldier in Iraq trying to control a crowd of civilians in a passionate manner. Credits to Brian Walski.

(b) Iranian montage (2008) of missiles streaking heavenward.

(c) Montage (2004) of John Kerry and Jane Fonda standing together at a podium during a 1970s anti-war rally. Credits to Ken Light (left), Associated Press (middle), and Owen Franken (right).

Fig. 2. Some common press media photomontages.

It has long been said that an image worth a thousand words. Recently, a study conducted by Italian Psychologists have investigated how doctored photographs of past public events affect memory of those events. Their results indicate that doctored photographs of past public events can influence memory, attitudes and behavioral intentions [Sacchi et al. 2007]. That might be one of the reasons that several dictatorial regimes routinely wiped out of their photographic records images of people who had fallen out of favor with the system [Farid 2006a].

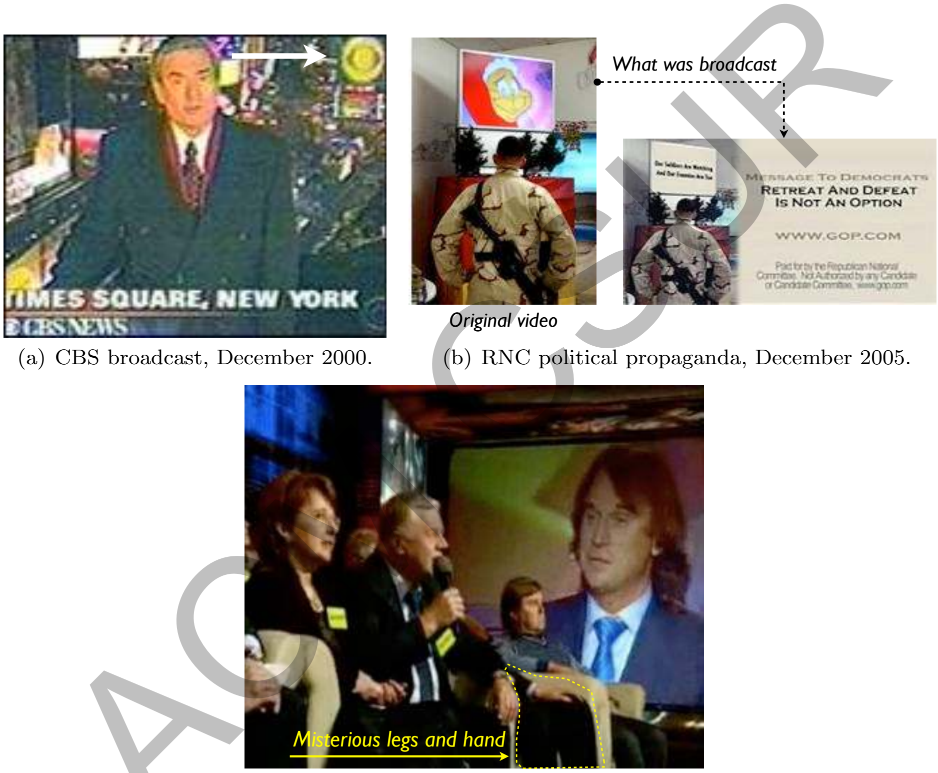
According to [Wang 2009], with the availability of sophisticated and low-cost digital video cameras and the convenience of video sharing websites such as *YouTube*, digital videos are playing a more important role in our daily-life. However, we can not take the authenticity of such videos for granted.

Although video tampering is relatively harder to perform than image tampering, it is not uncommon to find some dubious video editing cases in real life.

Video tampering can be as simple as inserting advertisements during broadcasting of sporting events or as complex as removing people digitally from a video. For instance, in December 2000, the CBS emblem on the frame depicted in Figure 3(a) of a live video broadcast was inserted so as to conceal the NBC emblem that was on display in the background.

Figure 3(b) depicts the final screen shot of a Republican National Committee political video of a U.S. soldier watching a television from December 2005. In this last shot, we read "Our soldiers are watching and our enemies are too". However, this video was digitally altered. It is an edition of another video where the soldier was watching the movie *How the Grinch Stole Christmas*.

Figure 3(c) depicts a Russian talk show in the fall of 2007. The video frames the political analyst Mikhail Delyagin making some sharp remarks about president Vladimir Putin. Later, when the program was broadcast, the analyst and his comments were digitally removed from the show. However, the technicians neglected to erase his hand and legs in one shot [Wang 2009].



(a) CBS broadcast, December 2000.

(b) RNC political propaganda, December 2005.

(c) Russian talk show in 2007. Prominent political analyst, Mikhail Delyagin, was digitally erased from the show except his hand and legs.

Fig. 3. Some video tampering examples.

3. VISION TECHNIQUES FOR THE UNSEEN

In this section, we survey many of the state-of-the-art approaches for image and video forensics, pointing out their advantages and limitations. In order to alleviate confusion with symbols that are used in different contexts by different works, we have changed nearly all of the symbols used throughout this section to resolve any ambiguity.

3.1 Image manipulation techniques

In the forensic point of view, it is paramount to distinguish simple image enhancements from image doctoring. We note that *any* image operation can be used to deceive the viewer; the distinction comes down to the intent of the person performing the image editing. Despite this burden, we generally find common operations falling into two categories.

On one extreme, we define image enhancements as operations performed in one image with the intention to improve its visibility. There is no local manipulation or pixel combination. Some image operations in this category are contrast and brightness adjustments, gamma correction, scaling, and rotation, among others. On the other extreme, image tampering operations are those used with the intention to deceive the viewer at some level. In these operations, normally one performs localized image operations such as pixel combinations and tweaks, copy/paste, and composition with other images. In between these extremes, there are some image operations that by themselves are not considered forgery operations but might be combined for such objective. Image sharpening, blurring, and compression are some of such operations.

Some common image manipulations with the intention of deceiving a viewer include:

(1) **Composition or splicing**. It consists of the composition (merging) of an image I_c using parts of one or more parts of images $I_1 \ldots I_k$. For example, with this approach, a politician in I_1 can be moved beside a person from I_2, without even knowing such person.

(2) **Retouching, healing, cloning**. These approaches consist of the alteration of parts of an image or video using parts or properties of the same image or video. Using such techniques, one can make a person 10 or 20 years younger (retouching and healing) or even change a crime scene eliminating a person in a photograph (cloning).

(3) **Content embedding or Steganography**. The goal of steganography is to convey a message by hiding it in a cover media (for the work considered here, the cover media consists of an image or video) without affecting the cover's statistical properties, which might be used for detection.

Figure 4 depicts some possible image manipulations. From the original image (top left), we clone several small parts of the same image in order to eliminate some parts of it (for example, the two people standing in front of the hills). Then we can use a process of smoothing to feather edges and make the cloning less noticeable. We can use this image as a host for another image (bottom left) and then create a composite. After the combination, we can use healing operations to adjust brightness, contrast, and illumination. This toy example was created in five minutes using the open-source software Gimp.

Sometimes the edge between image enhancing and faking is so thin that depending on the context, only the addition of text to a scene may fool the viewer. Figure 5

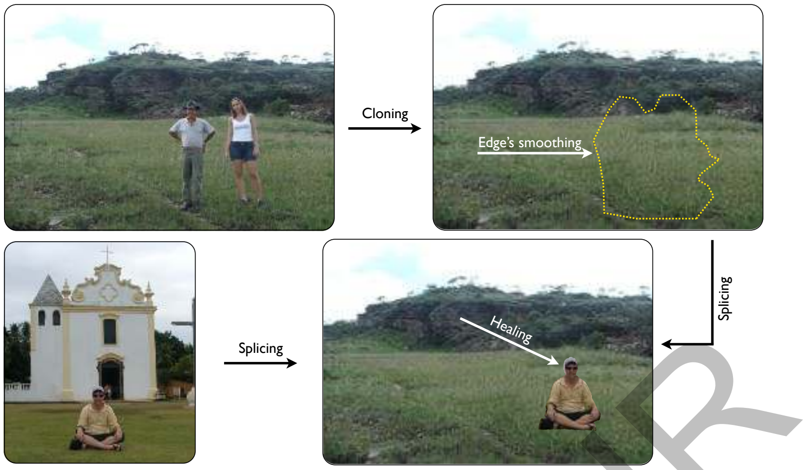


Fig. 4. Toy example of possible image manipulations, including cloning, edge smoothing, splicing, and healing.

depicts one example of two photographs presented by Colin Powell at the United Nations in 2003. The actual images are low-resolution, muddy aerial surveillance photographs of buildings and vehicles on the ground in Iraq. They were used to justify a war. Note that the text addition in this case was enough to mislead the United Nations [Errol Morris 2008]. However, many other simple image operations exist that many be considered purely innocuous, such as red eye removal, contrast enhancement, lossless compression and affine transformation. With these examples, no visual information representing objects in the image is lost, and they may be treated as conceptually invertible.

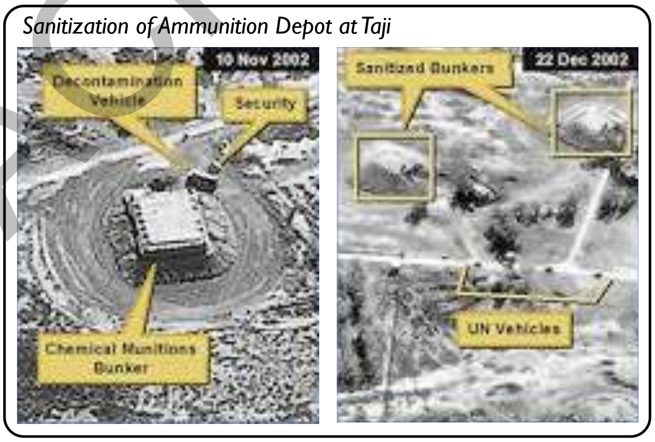


Fig. 5. Photographs presented by Colin Powell at the United Nations in 2003. (U.S. Department of State)

3.2 Important questions

In general, in digital image and video forensics, given an input digital image, for instance, one wants to answer the following important questions [Sencar and Memon 2008]:

- What imaging equipment produced this image?
  - Was this image acquired from camera vendor $C_{v,1}$ or $C_{v,2}$?
  - Was this image originally acquired with camera C as claimed?
- What is the processing history of this image?
  - Is this image an original image or has it been created from the composition (splicing) of other images?
  - Does this image represent a real moment in time or has it been tampered with to deceive the viewer?
  - Which part of this image has undergone manipulation and to what extent? What are the impacts of such modifications?
- Does this image conceal any hidden content?
  - Which algorithm or software has been used to perform the hiding?
  - Is it possible to recover the hidden content?

It is worth noting that most techniques for digital images and video are blind and passive. The approach is blind when it does not use the original content for the analysis. The approach is passive when it does not use any watermarking-based solution for the analysis.

Although digital watermarking can be used in some situations, the vast majority of digital contents do not have any digital watermarking. Any watermarking-based solution would require an implementation directly in the acquisition sensor, making its use restrictive. Furthermore, such approaches might lead to quality loss due to the markings [Ng et al. 2006; Sencar and Memon 2008].

We break up the image and video forensics approaches proposed in the literature into three categories, discussed in the next sections:

(1) Camera sensor fingerprinting or source identification;
(2) Image and video tampering detection;
(3) Image and video hidden content detection/recovery.

3.3 Source Camera Identification

With *Source Camera Identification*, we are interested in identifying the data acquisition device that generated a given image for forensics purposes [Swaminathan et al. 2009]. Source camera identification may be broken into two classes: device class identification and specific device identification. In general, source camera identification relies on the underlying characteristics of the components of digital cameras. These characteristics may take the form of image artifacts, distortions, and statistical properties of the underlying data. These characteristics are usually imperceptible to the human eye, but visible effects can also contribute clues for identification.

In general, we treat digital image acquisition as a pipeline of stages. Figure 6 illustrates the flow of data, with light initially passing through a lens and possibly

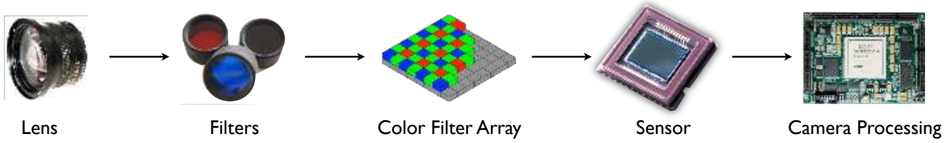


Fig. 6. The image acquisition pipeline.

through a filter (to remove infrared or ultra-violet light, for example). If the camera supports color, a Color Filter Array (CFA) is usually placed over the sensor to accommodate different color channels. Popular CFA configurations include the RGB Bayer Pattern (most common), and the CMYK subtractive color model (available on some higher end sensors). In a standard consumer grade camera, the sensor will be a silicon CCD or CMOS. The image processing will take place in logic designed by individual camera or chipset manufacturers within the camera itself. Each of these pipeline components induce anomalies in images that can be used to identify a source camera.

3.3.1 *Device Class Identification.* The goal of device class identification is to identify the model and/or manufacturer of the device that produced the image in question. For digital cameras, we consider the image acquisition pipeline, where the lens, size of the sensor, choice of CFA, and demosaicing and color processing algorithms found in the camera processing logic to provide features. It is important to note that many manufacturers use the same components, thus, the discriminatory power of some techniques may be limited. Many of the techniques that we will discuss here treat the underlying camera characteristics as features for machine learning, which separates images into particular camera classes. Thus, we can treat device class identification as a traditional classification problem. Support Vector Machines (SVM), shown in Figure 7, is a popular binary classifier for device class separation. It can also be extended for multi-class classification. In this section, we will review the relevant techniques used to identify device classes.

From the lens, radial distortions can be introduced immediately into the image acquisition pipeline. Radial distortion is commonly found with inexpensive cameras/lenses. Choi et al. [2006] introduce a method to extract aberrations from images, which are then treated as features for classification. As described in [Choi et al. 2006], radial distortion can be modeled through the second order for reasonable accuracy:

$$r_u = r_d + d_1 r_d^3 + d_2 r_d^5 \tag{1}$$

where d_1 and d_2 are the first and second degree distortion parameters, and r_u and r_d are the undistorted radius and the distorted radius. The radius is simply the radial distance $\sqrt{x^2+y^2}$ of some point (x, y) from the center of the distortion (typically the center of the image). The parameters d_1 and d_2 are treated as features for an SVM learning system. These features, however, are not used in [Choi et al. 2006] by themselves — they are combined with the 34 image features introduced in [Kharrazi et al. 2004] (described below), in a fusion approach. Thus, the utility of this approach may be seen as a supplement to other, stronger features derived

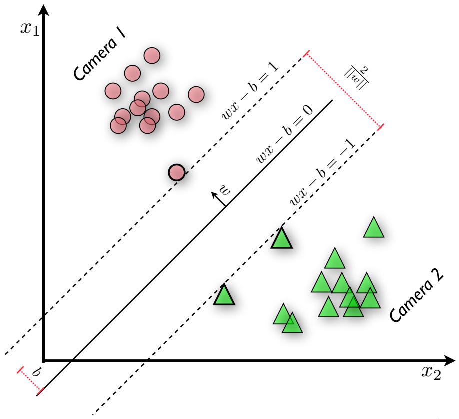


Fig. 7. An example of binary camera classification with SVM. A feature vector is constructed out of the calculated features for a given image. Training sets are built out of a collection of feature vectors for each camera class. The machine learning is used for classification of images with unknown sources.

from elsewhere in the acquisition pipeline. The average accuracy of this technique is reported to be about 91% for experiments performed on three different cameras from different manufacturers.

Image color features exist as artifacts induced by the CFA and demosaicing algorithm of a color camera, and represent a rich feature set for machine learning based classification. Kharrazi et al. [2004] define a set of image color features that are shown to be accurate for device class identification using SVMs. Average pixel values, RGB pairs correlation, neighbor distribution center of mass, RGB pairs energy ratio, and wavelet domain statistics are all used as features. Further, image quality features are also used to supplement the color features in [Kharrazi et al. 2004]. Pixel difference based measures (including mean square error, mean absolute error, and modified infinity norm), correlation based measures (including normalized cross correlation, and the Czekonowksi correlation, described below), and spectral distance based measures (including spectral phase and magnitude errors) are all used. For binary classification, Kharrazi et al. [2004] report between 90.74% and 96.08% prediction accuracy. For multi-classification considering 5 cameras, prediction accuracy between 78.71% and 95.24% is reported. These results were confirmed in [Tsai and Wu 2006].

The CFA itself as a provider of features for classification has been studied in [Celiktutan et al. 2005]. The motivation for using just the CFA and its associated demosaicing algorithm is that proprietary demosaicing algorithms leave correlations across adjacent bit planes of the images. Celiktutan et al. [2005] define a set of similarity measures $\{m_1, m_2, m_3\}$, with kNN and SVM used for classification.

The first approach is a binary similarity measure. A stencil function is first defined

$$\delta_c^n(a,b) = \begin{bmatrix} 1 \; if \;\; p_c = 0 \;\; p_n = 0 \\ 2 \; if \;\; p_c = 0 \;\; p_n = 1 \\ 3 \; if \;\; p_c = 1 \;\; p_n = 0 \\ 4 \; if \;\; p_c = 1 \;\; p_n = 1 \end{bmatrix} \tag{2}$$

where b is a bit plane (image matrix), and a indicates one of four agreement scores: $1, 2, 3,$ and 4. The subscript c defines some central pixel, and superscript n denotes one of the four possible neighbor pixels.

We sum $\delta_c^n(a,b)$ over its four neighbors (n runs over its East, West, South, and North neighbors), as well as over all the pixels (c runs over the $M \times N$ pixels). After the summations the sub- and superscripts can be omitted.

Before feature generation, the agreement scores are normalized to obtain a PDF:

$$\mathcal{A}_a^b = \delta(a,b) / \sum_a \delta(a,b). \tag{3}$$

Based on this four-bin histograms, we are able to define the binary Kullback-Leibler distance as

$$m_1 = -\sum_{n=1}^{4} \mathcal{A}_n^7 \log \frac{\mathcal{A}_n^7}{\mathcal{A}_n^8} \tag{4}$$

where $\mathcal{A}$ is the normalized agreement score.

The second approach is also a binary similarity measure, but uses a neighborhood weighting mask as opposed to a stencil function. Each binary image yields a 512-bin histogram computed using the weighted neighborhood. Each score is computed with the following function:

$$S = \sum_{i=0}^{7} p_i 2^i \tag{5}$$

The neighborhood weighting mask applied to a pixel p_i by the above function is:

1	2	4
128	256	8
64	32	16

The final binary similarity is computed based on the absolute difference between the n^{th} histogram bin in the 7^{th} bit plane and same of the 8^{th} after normalization:

$$m_2 = \sum_{n=0}^{511} |S_n^7 - S_n^8| \tag{6}$$

Quality measures, as mentioned earlier, make excellent features for classification. The Czenakowski distance is a popular feature for CFA identification because it is able to compare vectors with non-negative components — exactly what we find in color images. The third feature of [Celiktutan et al. 2005] is the Czenakowski distance defined as:

$$m_3 = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left(1 - \frac{2 \sum_{k=1}^{3} \min(I_k(i,j), \hat{I}_k(i,j))}{\sum_{k=1}^{3} (I_k(i,j) + \hat{I}_k(i,j))} \right) \tag{7}$$

Denoising is necessary for calculating this distance metric. $I_k(i,j)$ represents the $(i,j)^{th}$ pixel of the k^{th} band of a color image, with $\hat{I}_k$ being the denoised version. With these three similarity measures the authors of [Celiktutan et al. 2005] generate 108 binary similarity features and 10 image quality similarity features per image. The best reported performance for this technique (using SVM for classification) is near 100% accuracy for the two camera classification problem, 95% accuracy for the three camera classification problem, and 62.3% accuracy for a six camera classification problem.

A major weakness of the approaches described thus far is a lack of rigor in the analysis of the experimental results reported, compared with other security related vision and pattern recognition fields such as biometrics and tracking. All report raw classification results for only a handful of different cameras. Thus, it is often difficult to determine how well these techniques perform in practice. This is a common problem of this sub-field in general. By varying the SVM margin after classification, a set of marginal distances can be used to build a Receiver Operator Characteristic curve. From this curve, a more thorough understanding of the False Reject Rate (FRR) and False Accept Rate (FAR) can be gained. Also of interest is more comprehensive testing beyond limited camera classes. For a more accurate picture of the FAR, a statistically large sampling of images from cameras outside the known camera classes should be submitted to a system. None of the papers surveyed attempted this experiment. Further, the techniques introduced thus far are all shown to succeed on images with low levels of JPEG compression. How well these techniques work with high levels of compression has yet to be shown. Not all work suffers from a dearth of analysis, however.

The Expectation/Maximization algorithm [Popescu 2004] is a powerful technique for identifying demosaicing algorithms (and even more useful for forgery detection, described in Section 3.4.4), and does not rely on classification techniques directly, but can take advantage of them in extensions to the base work ([Bayram et al. 2005] , [Bayram et al. 2006]). The motivating assumption of the EM algorithm is that rows and columns of interpolated images are likely to be correlated with their neighbors. Kernels of a specified size (3×3, 4×4, and 5×5 are popular choices) provide this neighborhood information to the algorithm. The algorithm itself can be broken into two steps. In the *Expectation* (E) step, the probability of each sample belonging to a particular model is estimated. In the *Maximization* (M) step, the specific form of the correlations between samples is estimated. Both steps are iterated till convergence.

In detail, we can assume that each sample belongs to one of two models. If a sample is linearly correlated with its neighbors, it belongs to $\mathcal{M}_1$. If a sample is not correlated with its neighbors, it belongs to $\mathcal{M}_2$. The linear correlation function is defined as:

$$f(x,y) = \sum_{u,v=-k}^{k} \alpha_{u,v} f(x+u, y+v) + \mathcal{N}(x,y) \tag{8}$$

In this linear model, $f(\cdot,\cdot)$ is a color channel (R, G, or B) from a demosaiced image, k is an integer, and $\mathcal{N}(x,y)$ represents independent, identically distributed samples drawn from a Gaussian distribution with zero mean and unknown variance. $\vec{\alpha}$ is a vector of linear coefficients that express the correlations, with $\alpha_{0,0} = 0$.

The E step estimates the probability of each sample belonging to $\mathcal{M}_1$ using Bayes' rule:

$$\Pr\{f(x,y) \in \mathcal{M}_1 | f(x,y)\} = \frac{\Pr\{f(x,y)|f(x,y) \in \mathcal{M}_1\}\Pr\{f(x,y) \in \mathcal{M}_1\}}{\sum_{i=1}^{2} \Pr\{f(x,y)|f(x,y) \in \mathcal{M}_i\}\Pr\{f(x,y) \in \mathcal{M}_i\}} \tag{9}$$

$\Pr\{f(x,y) \in \mathcal{M}_1\}$ and $\Pr\{f(x,y) \in \mathcal{M}_2\}$ are prior probabilities assumed to be equal to 1/2. If we assume a sample $f(x,y)$ is generated by $\mathcal{M}_1$, the probability of this is:

$$\Pr\{f(x,y)|f(x,y) \in \mathcal{M}_1\} = \frac{1}{\sigma\sqrt{2\pi}}\left[-\frac{1}{2\sigma^2}\left(f(x,y) - \sum_{u,v=-k}^{k} \alpha_{u,v} f(x+u, y+v)\right)^2\right]. \tag{10}$$

We estimate the variance σ^2 in the M step. $\mathcal{M}_2$ is assumed to have a uniform distribution.

The M step computes an estimate of $\vec{\alpha}$ using weighted least squares (in the first round of the E step, $\vec{\alpha}$ is chosen randomly):

$$E(\vec{\alpha}) = \sum_{x,y} w(x,y)\left(f(x,y) - \sum_{u,v=-k}^{k} \alpha_{u,v} f(x+u, y+v)\right)^2 \tag{11}$$

The weights $w(x,y)$ are equivalent to $\Pr\{f(x,y) \in \mathcal{M}_1 | f(x,y)\}$. This error function is minimized via a system of linear equations before yielding its estimate. Both the steps are executed until a stable $\vec{\alpha}$ results. The final result maximizes the likelihood of observed samples.

Popescu [2004] asserts that the probability maps generated by the EM algorithm can be used to determine which demosaicing algorithm a particular camera is using. These probabilities tend to cluster — thus, an external machine learning algorithm for classification is not necessary. For a test using eight different demosaicing algorithms [Popescu 2004], the EM algorithm achieves an average classification accuracy of 97%. In the worst case presented (3 × 3 median filter vs. variable number of gradients), the algorithm achieves an accuracy of 87%. Several extensions to the EM algorithm have been proposed. Bayram et al. [2005] apply the EM algorithm to a camera identification problem, using SVM to classify the probability maps. Bayram et al. [2005] report success as high as 96.43% for the binary classification problem, and 89.28% for the multi-class problem. Bayram et al. [2006] introduce better detection of interpolation artifacts in smooth images as a feature to fuse with the standard EM results. For a three camera identification problem, Bayram et al. [2006] achieve results as high as 97.74% classification accuracy. Other variations include the use of modeling error, instead of interpolation filter coefficients [Long and Huang 2006], and the computation of error based on the assumption of CFA patterns in an image [Swaminathan et al. 2006].

3.3.2 *Specific Device Identification.* The goal of specific device identification is to identify the exact device that produced the image in question. For specific device identification, we require more detail beyond what we've discussed so far with source model identification. Features in this case may be derived from:

- hardware and component imperfections, defects, and faults

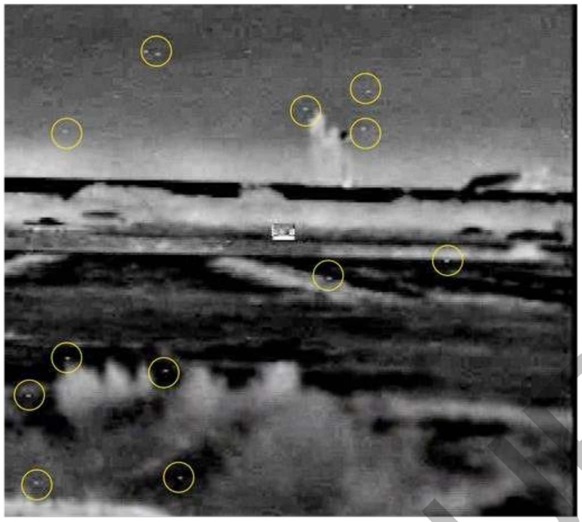


Fig. 8. Dead pixels (circled in yellow) present in an image from a thermal surveillance camera.

- effects of manufacturing process, environment, operating conditions
- aberrations produced by a lens, noisy sensor, dust on the lens

It is important to note that these artifacts may be temporal by nature, and thus, not reliable in certain circumstances.

Early work [K. Kurosawa and Saitoh 1999] in imaging sensor imperfections for specific device identification focused on detecting fixed pattern noise caused by *dark current* in digital video cameras. Dark current is the rate that electrons accumulate in each pixel due to thermal action. This thermal energy is found within inverse pin junctions of the sensor, and is independent of light falling on it. In their work, the authors just intensify the fixed pattern noise components while proposing their detection as local pixel defects. The work, as presented in [K. Kurosawa and Saitoh 1999], provides no quantitative analysis, and thus, the actual utility of dark currents cannot be assessed.

A more comprehensive use of sensor imperfections is presented in [Geradts et al. 2001], where "hot pixels," cold/dead pixels, pixel traps, and cluster defects are used for detection. Hot pixels are individual pixels on the sensor with higher than normal charge leakage. Cold or dead pixels (Figure 8) are pixels where no charge ever registers. Pixel traps are an interference with the charge transfer process and results in either a partial or whole bad line, that is either all white or all dark. While these features are compelling for identifying an individual sensor, one immediate problem with defective pixels that we can think of is that there are cameras that eliminate them by post-processing their images on-board. Geradts et al. [2001] also do not provide a quantitative analysis that we can use to assess the effectiveness of defective pixels analysis. Thus, we turn to more extensive work for reliable forensics.

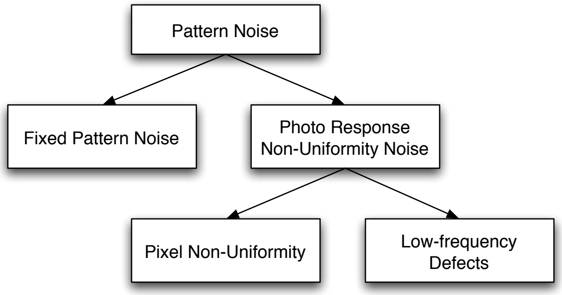


Fig. 9. Hierarchy of Pattern Noise.

Lukas et al. [2006] present a more formal quantification and analysis of sensor noise for identification, with work that is the strongest for this type of forensics. Referring to the hierarchy of sensor noise in Figure 9, we see two main types of pattern noise: fixed pattern noise and photo-response non-uniformity noise. Fixed pattern noise (FPN) is caused by the dark currents described above, and is not considered in [Lukas et al. 2006]. The authors state that the reason is FPN primarily refers to pixel-to-pixel differences when the sensor array is not exposed to light. Basically, FPN is an additive noise, which depends on exposure and temperature. It can be suppressed on-the-fly by some in-camera devices by subtracting a dark frame from the image right after its capture.

Photo-response non-uniformity noise (PRNU) is primarily caused by pixel non-uniformity noise (PNU). PNU is defined as different sensitivity various pixels have to light caused by the inconsistencies of the sensor manufacturing process. Low frequency defects are caused by light refraction on particles on or near the camera, optical surfaces, and zoom settings. Lukas et al. [2006] do not consider this type of noise, but Dirik et al. [2008] do. The temporal nature of such particle artifacts brings into question their reliability - except when dealing with short sequences of images from the same period.

To use PNU as a characteristic for sensor fingerprinting, the nature of the noise must first be isolated. An image signal μ exhibits properties of a white noise signal with an attenuated high frequency band. The attenuation is attributed to the low-pass character of the CFA algorithm (which, in this case, we are not interested in). If a large portion of the image is saturated (pixel values set to 255), it will not be possible to separate the PNU from the image signal. In a forensic scenario, we will likely not have a blank reference image that will easily allow us to gather the PNU characteristics. Thus, the first stage of the PNU camera identification algorithm is to establish a reference pattern P_c, which is an approximation to the PNU. The approximation, $\bar{I}^{(k)}$ is built from the average of K different images of a uniformly lit scene $k = 1, \cdots, K$:

$$\bar{I}^{(k)} = \frac{1}{K} \sum_{k=1}^{K} I^k \tag{12}$$

The approximation can be optimized to suppress the scene content by applying a de-noising filter λ, and averaging the noise residuals $\xi^{(k)}$ instead of the original images $I^{(k)}$:

$$\bar{\xi}^{(k)} = (\bar{I}^{(k)} - \lambda(I^{(k)}))/K \tag{13}$$

Lukas et al. [2006] report that a wavelet-based denoising filter works the best.

To determine if an image belongs to a particular known camera, a correlation ρ_c is simply calculated between the noise residual of the image in question $\xi = I - \lambda(I)$ and the reference pattern P_c (a bar above the symbol represents mean):

$$\rho_c(I) = \frac{(\xi - \bar{\xi}) \cdot (P_c - \bar{P}_c)}{\|\xi - \bar{\xi}\| \|P_c - \bar{P}_c\|} \tag{14}$$

The results of [Lukas et al. 2006] are expressed in terms of FRR and FAR (proper ROC curves are not provided, however), with very low FRR (between 5.75×10^{-11} and 1.87×10^{-3}) reported when a FAR of 10^{-3} is set for an experiment with images from nine different cameras. Excellent correlations are shown for all tests, indicating the power this technique has for digital image forensics. An enhancement to this work has been proposed by [Sutcu et al. 2007], with a technique to fuse the demosaicing characteristics of a camera described earlier with the PNU noise. Performance is enhanced by as much as 17% in that work over the base PNU classification accuracy.

One drawback of PRNU noise when used for camera and sensor fingerprinting is that its detection is sensitive to appropriate synchronization. A slight scaling or cropping operation on the image can lead to an unsuccessful detection [Goljan and Fridrich 2008]. Geometrical transformations (e.g., scaling and rotation) cause desynchronization and introduce distortions due to resampling.

In this regard, Goljan and Fridrich [2008] have extended previous camera identification technology based on sensor noise [Lukas et al. 2006] to a more general setting when the image under investigation has undergone cropping and scaling operations. Prior to perform the reference pattern comparisons, the authors deploy a brute force search to find the scaling factor of the analyzed image. Thereafter, the authors use the peak to correlation (PCE) of the normalized cross-correlation (NCC) surface between the reference patterns (the resized image under investigation and the camera reference pattern) in order to estimate the cropping parameters. This is performed till a stopping criteria is achieved. The authors report good results for reliable camera identification for images linearly scaled down by a factor of 0.5 or more and images with 90% or more cropped away.

Although Goljan and Fridrich [2008]'s work is important towards a more reliable camera identification process, it is worth noting that: (1) the quality of the response depends on the image content and subsequent JPEG compression; and (2) it is computationally intensive. In this context, there is room for research. For instance, one could think of a more efficient way of estimating the cropping and scaling parameters other than brute force and also in how to perform the comparisons of the reference patterns. The authors, themselves, propose at the very end of their work, that a hierarchical search could be employed to achieve a 4 times reduction in search time [Goljan and Fridrich 2008].

3.3.3 *Specific Device Identification from Analog Media.* What do we do if instead of a digital object, only its analog form is available? One example is forged currency, when a digital scan of a real banknote is printed on a printer [Goljan et al. 2008]. Does the sensor pattern noise fingerprint survive the digital-to-analog conversion? Goljan et al. [2008] have proposed to investigate these questions.

As discussed earlier in this paper, sensor noise-based signatures (namely PRNU) need proper synchronization to work. In [Goljan et al. 2008], the authors discuss approaches to circumvent the most influential factors in identifying the sensor from a printed picture: (1) the accuracy of the angular alignment when scanning, (2) printing quality, (3) paper quality, and (4) size of the printed picture. Geometrical transformations are the most difficult problem to address.

To tackle the geometrical transformations problem, Goljan et al. [2008] deploy the same technique used in [Goljan and Fridrich 2008] to find the cropping and scaling parameters on the scanned object of investigation. With these parameters, the authors modify the object in question and compare it to a set of reference patterns. The one signaling the highest response is considered the reference pattern of the camera that acquired the image. The reported results show that the PRNU camera signature is robust with respect to high quality printing and scanning. However, for regular paper prints identification there is still room for research.

3.3.4 *Specific Device Identification — Camcorders.* Since digital camcorders use, in essence, the same capturing imaging sensors, another valid question that arises is whether or not we can correctly identify them. According to [Chen et al. 2007], determining if two videoclips come from the same source could be an important tool for fighting motion picture piracy. Approaches limited to identifying only the camcorder model could be easily extended from previous work such as [Kharrazi et al. 2004; Popescu 2004; Popescu and Farid 2005a; 2005b; Swaminathan et al. 2006], where the authors propose to identify different traces of image processing unique to a specific camcorder.

Recently, Chen et al. [2007] have investigated the PRNU noise signatures proposed in [Lukas et al. 2006] to differentiate specific camcorders (e.g., same brand and model). According to the authors, the approach discussed in [Lukas et al. 2006] cannot be used directly because the spatial resolution of the video is usually smaller than typical still images and each frame can be subjected to different compression levels. First, the authors estimate the PRNU noise from the video clip taking advantage of its temporal resolution using a maximum likelihood estimator. Afterwards, the PRNU signature is filtered to eliminate blockiness artifacts present due to different video coding formats (lossy compression). Given two filtered PRNU signatures, they are processed using normalized cross-correlation and the peak to correlation energy coefficient is calculated over the NCC surface to establish whether or not they have a common origin.

The authors report that about 40 seconds of video is enough for a reliable decision. However, if the video is low quality, more frames need to be used. Research opportunities are available when dealing with larger compression and decreasing spatial resolution videoclips. For instance, for some "Internet quality" videos in LP resolution (264 × 352, 150Kb/sec. bit-rate) a 10-minute video sequence would be required for signature estimation.

Finally, it is worth mentioning one limitation of the method. For simplification purposes, the authors have assumed all frames in a video clip have the same variance (distortion caused by independent noise components). The authors themselves point out that some video coding schemes (e.g., DVD) use variable bit rate coding leading to almost constant picture quality, while others (e.g., DTV) use constant bit rate coding leading to variable quality. Therefore, the adaptive estimation of the variance could give better results and it is worth investigating.

3.3.5 *Counter Forensic Techniques Against Camera Identification.* Like any subfield of digital forensics, camera identification is susceptible to counter forensic techniques. Gloe et al. [2007] introduce two techniques for manipulating the image source identification of [Lukas et al. 2006]. This work makes the observation that applying the wavelet denoising filter of [Lukas et al. 2006] is not sufficient for creating a quality image. Thus, a different method, *flatfielding*, is applied to estimate the FPN and the PRNU. FPN is a signal *independent* additive noise source, while PRNU is a signal *dependent* multiplicative noise source. For the FPN estimate, a dark frame $I_{dark\_estimate}$ is created by averaging J images I_{dark} taken in the dark (with the lens cap on, for instance):

$$I_{dark\_estimate} = \frac{1}{J}\sum_{J} I_{dark} \tag{15}$$

For the PRNU estimate, K images of a homogeneously illuminated scene I_{light} with $I_{dark\_estimate}$ subtracted are required. To calculate the flatfield frame $I_{flatfield}$, these images are averaged:

$$I_{flatfield} = \frac{1}{K}\sum_{K}(I_{light} - I_{dark\_estimate}) \tag{16}$$

With an estimate of the FPN and PRNU of a camera, a nefarious individual can suppress the noise characteristics of an image from a particular camera to avoid identification in some (not all) cases. An image $\hat{I}$ with suppressed noise characteristics is simply created by noise minimization:

$$\hat{I} = \frac{I - I_{dark\_estimate}}{I_{flatfield}} \tag{17}$$

The authors of [Gloe et al. 2007] note that perfect flatfielding is, of course, not achievable, as an immense number of parameters (exposure time, shutter speed, and ISO speed) would be needed to generate $I_{dark\_estimate}$ and $I_{flatfield}$. Thus, they fix upon a single parameter set for their experiments. Results for this technique are reported for RAW and TIFF images. While powerful, flatfielding is not able to prevent identification in all images it is applied to. Figure 8 in [Gloe et al. 2007] appears to depict many instances of a correctly identified camera after the application of flatfielding, with the authors stating "correct identification of image origin was successfully prevented only for a subset of images". This subset is not identified.

Simply reducing the impact of camera identification by PRNU is not the only thing one can do with flatfielding. After the above technique has been applied, a noise pattern from a different camera can be added with inverse flatfielding. An image $\hat{I}_{forge}$ with forged noise characteristics is created from the pre-computed flatfielding information from any desired camera:

$$\hat{I}_{forge} = \hat{I} \cdot I_{flatfield\_forge} + I_{dark\_forge} \tag{18}$$

Experiments for this technique are also presented in [Gloe et al. 2007], where images from a Canon Powershot S70 are altered to appear to be from a Canon Powershot S45. While most correlation coefficients mimic the S45, some still remain characteristic of the S70. The counter forensic techniques of [Gloe et al. 2007] are indeed useful in many circumstances, but are shown to be too simplistic to fool a detection system absolutely. Further, such limited testing only hints at the potential of such techniques. As the "arms race" continues, we expect attacks against camera identification to increase in sophistication, allowing for more comprehensive parameter coverage and better noise modeling.

3.3.6 *Distinguishing Computer Graphics from Digital Photography.* As noted in Section 1, there is an important legal burden to distinguish imagery consisting of computer graphics (CG) from digital photography. Thus, in some cases where the authenticity of an image is in question, before we apply the source camera identification techniques detailed above, we can check if an image was produced by a camera in the first place. The techniques for CG detection fall into three categories [Ng and Chang 2009]: statistical wavelet features, physical models of images and camera related characteristics. In this section, we will review the most relevant work for each category.

Lyu and Farid [2005] introduced a statistical model based on first- and higher-order wavelet statistics that is able to reveal the difference between CG imagery and digital photographs that are indistinguishable to the human eye. For feature extraction, images are decomposed in frequency space into multiple scales and orientations, including vertical $Vert_i^c(x, y)$, horizontal $Horiz_i^c(x, y)$, and diagonal $Diag_i^c(x, y)$ subbands, where $c \in \{r, g, b\}$ for color imagery. For the statistical model, two sets of statistics are calculated. First, the first four order statistics (mean, variance, skewness, and kurtosis) of the subband coefficient histograms of the decompositions are computed. Second, higher-order statistics are calculated based on the errors in a linear predictor of coefficient magnitude. Using an SVM classifier with both orders of statistical features, Lyu and Farid [2005] show that this model can correctly classify 67% of photorealistic images, while only mis-classifying 1% of the photographic images for a dataset of 40,000 photographic images and 6,000 CG images. In similar work, Wang and Moulin [2006] use wavelet statistics, but improve the computational efficiency of [Lyu and Farid 2005] by a $4x$ speed enhancement.

Ng et al. [2005] propose a geometry based image model that is motivated by the physical differences between CG imagery and photographic imagery. The authors develop two levels of image discrimination: image-process authenticity and scene authenticity. Image-process authenticity is defined as images acquired by a sensor based image acquisition device, such as a camera or scanner. Scene authenticity is defined as the result of a snapshot of a physical light field. A series of features that capture information to support both definitions are computed. These features include:

(1) *Local Fractal Dimension*; to capture the texture complexity in photographs
(2) *Local Patch Vectors*; to capture the characteristics of the local edge profile
(3) *Surface Gradient*; to capture the shape of a camera response function
(4) *Principle Components of Local Quadratic Geometry*; to capture the artifacts due to the CG polygonal model
(5) *Beltrami Flow Vector*; to capture the artifacts due to the color independence assumption in CG

The above five features are considered together, producing a vector field on the image domain. The actual features used as input to an SVM classification system are rigid body moment statistics computed from the vector field. On a data set consisting of 3200 images including the categories of CG images, personal photographs, photographs collected from Google Image Search, and CG images re-imaged by a camera system, Ng et al. [2005] report a classification accuracy of 83.5%.

Camera specific characteristics, such as what we've looked at for source identification can also be used to distinguish CG images from photographs. Revisiting the work of [Lukas et al. 2006], Dehnie et al. [2006] use noise characteristics to establish the difference between different camera classes and CG images. The idea is that even though different cameras possess unique noise characteristics, statistical properties exist that correlate these characteristics across cameras to some degree. CG images do not posses these common noise characteristics. Dehnie et al. [2006] present compelling results of statistical difference for CG images generated by Maya and 3D Studio Max software, and digital photographs. The Bayer pattern also lends clues to distinguishing the origin of an image. Dirik et al. [2007] takes the idea of demosaicing features from [Bayram et al. 2005] & [Swaminathan et al. 2006], but modifies their usage by detecting the presence of CFA interpolation as opposed to estimating CFA interpolation coefficients. The presence of chromatic aberration in an image is also used as a feature. In a variety of test cases using an SVM classifier, Dirik et al. [2007] show accuracy of over 90%. Gallagher and Chen [2008] also propose a demosaicing detection approach, this time by considering the weighted linear combination of neighboring pixel values. The authors suggest that the weights directly affect the variance of the distributions from which interpolated pixels values are drawn. Very high accuracy is achieved (98.4% average accuracy) on Columbia's ADVENT[6] data set using an unnamed machine learning classifier. [Gallagher and Chen 2008] contains experiments only on JPEG imagery, and adjusts the compression parameters in many experiments; it is unclear what effect the underlying quantization is having on the correlation algorithm.

Rocha and Goldenstein [2007; Rocha and Goldenstein [2010] have showed that the *Progressive Randomization* meta-descriptor, previously introduced for Steganalysis [Rocha and Goldenstein 2006], is also suitable for distinguishing computer generated from natural images. The method captures the differences between image classes (e.g., natural and CG images) by analyzing the statistical artifacts inserted during controlled perturbation processes with increasing randomness.

Methods for distinguishing computer graphics from digital photography are also prone to counter-forensics techniques. A simple countermeasure that can be used

[6] http://www.ee.columbia.edu/ln/dvmm/downloads/PIM_PRCG_dataset/

by an attacker is the re-imaging technique introduced in [Ng et al. 2005], whereby the detector is confused by being presented with a photograph of a CG scene. Ng et al. [2005] addresses this problem by using this type of data in their SVM training set. Yu et al. [2008] presents another technique to detect re-imaging, harnessing the observation that the specularity of a recaptured photograph is modulated by the mesostructure of the photograph's surface. Thus, its spatial distribution can be used for classification. On a small data set of 400 images (200 original and 200 re-imaged), the authors validate their statistical classifiers. As with source camera identification, counter-forensics techniques here are in their infancy, and we expect to see attacks that are more sophisticated than re-imaging emerge in the near future.

3.4 Image and video tampering detection

In general, image and video tampering detection approaches [Farid 2009] rely on analyzing several properties such as: detection of cloned regions, analysis of feature variations collected from sets of original and tampered scenes, inconsistencies in the features, inconsistencies regarding the acquisition process, or even structural inconsistencies present in targeted attacks. In the following, we describe each one of such approaches and their limitations.

3.4.1 *Image cloning detection.* Cloning is one of the simplest forgeries an image can undergo. It is known as copy/move and also is present in more sophisticated operations such as healing. Often, the objective of the cloning operation is to make an object "disappear" from one scene using properties of the same scene (for example, neighboring pixels with similar properties). Cloning detection is a problem technically easy to solve using exhaustive search. However, such brute-force solutions are computationally expensive.

Fridrich et al. [2003] propose a faster and more robust approach for detecting duplicated regions in images. The authors use a sliding window over the image and calculate the discrete cosine transform (DCT) for each region[7]. Each calculated DCT window is stored row-wise in a matrix $A_{\mathcal{D}}$. The authors propose to calculate a quantized DCT in order to be more robust and perform matchings for non-exact cloned regions. The next step consists of lexicographically sorting matrix $A_{\mathcal{D}}$ and searching for similar rows. To reduce the resulting false positives, the authors proposed a post-processing step in which they only consider two rows as a clone candidate if more rows share the same condition and are close in the image space to these two rows. Popescu and Farid [2004a] propose a similar approach switching the DCT calculation to a Karhunen-Loeve Transform and reported comparable results.

As we discussed in Section 1, forgeries are present in the scientific community. Some authors may use image tampering to improve their results and make them look more attractive. Farid [2006b] have framed the detection of some scientific image manipulations as a two-stage segmentation problem. The proposed solution is suited for grayscale images such as gel DNA response maps. In the first iteration, the image is grouped, using intensity-based segmentation into regions correspond-

[7] In the Appendix A, we present a brief description of the discrete cosine transform and JPEG compression.

ing to the bands (gray pixels) and the background. In the second iteration, the background region is further grouped into two regions (black and white pixels) using the texture-based segmentation. Both segmentations are performed using normalized cuts [Shi and Malik 2000]. The authors suggest that the healing and cloning operations will result in large segmented cohesive regions in the background that are detectable using a sliding window and ad-hoc thresholds. This approach seems to work well for naive healing and cloning operations, but only a few images were tested. It would be interesting to verify if a copied band of another image still would lead to the same artifacts when spliced in the host image.

3.4.2 *Video splicing and cloning detection.* Wang and Farid [2007b] argue that the two previous approaches are too computationally inefficient to be used in videos or even for small sequences of frames and propose an alternative solution to detect duplicated regions across frames. Given a pair of frames $I(x, y, \tau_1)$ and $I(x, y, \tau_2)$, from a stationary camera, the objective is to estimate a spatial offset (Δ_x, Δ_y) corresponding to a duplicated region of one frame placed in another frame in a different spatial location. Towards this objective, the authors use phase correlation estimation [Castro and Morandi 1987]. First, the normalized cross power spectrum is defined:

$$\Psi(\omega_x, \omega_y) = \frac{F(\omega_x, \omega_y, \tau_1)F^*(\omega_x, \omega_y, \tau_2)}{||F(\omega_x, \omega_y, \tau_1)F^*(\omega_x, \omega_y, \tau_2||}, \tag{19}$$

where $F(\cdot)$ is the Fourier transform of a frame, $*$ is the complex conjugate, and $|| \cdot ||$ is the complex magnitude. Phase correlation techniques estimate spatial offsets by extracting peaks in $\psi(x, y)$, the inverse Fourier transform of $\Psi(\omega_x, \omega_y)$. A peak is expected at origin (0,0) as it is a stationary camera. Peaks at other positions denote secondary alignments that may represent a duplication but also simple camera translations (for non-stationary cameras). The spatial location of a peak corresponds to candidate spatial offsets (Δ_x, Δ_y). For each spatial offset, the authors calculate the correlation between $I(x, y, \tau_1)$ and $I(x, y, \tau_2)$ to determine if an offset corresponds to a determined duplication. Toward this objective, each frame is tiled into 16×16 overlapping (1 pixel) blocks and the correlation coefficient between each pair of corresponding blocks is computed. Blocks whose correlation is above a threshold are flagged as duplications. The authors also propose an extension for non-stationary cameras. For that, they calculate a rough measure of the camera motion and compensate by selecting subsequent non-overlapping frames. One drawback of this approach is that it assumes that the duplicated regions are rough operations (they do not undergo significant adjustments in the host frame).

Wang and Farid [2007a] present an approach for detecting traces of tampering in interlaced and de-interlaced videos. For de-interlaced videos, the authors use an expectation maximization algorithm to estimate the parameters of the underlying de-interlacing algorithm. With this model, the authors can point out the spatial/temporal correlations. Tampering in the video is likely to leave telltale artifacts that disturb the spatial/temporal correlations. For interlaced videos, the authors measure the inter-field and inter-frame motion that are often the same for an authentic video, but may be different for a doctored video. Although effective to some extent, it is worth discussing some possible limitations. The solution suitable for interlaced videos is sensitive to compression artifacts hardening the correlations estimation. In addition, a counter-attack to the de-interlacing approach consists of performing the video tampering and then generating an interlaced video (splitting the even and odd scan lines), and applying a de-interlacing algorithm on top of that to generate a new de-interlaced video whose correlations will be intact.

3.4.3 *Variations in image features.* Bayaram et al. [2006] have framed the image forgery detection problem as a feature and classification fusion problem. The authors claim that doctoring typically involves multiple steps, which often demand a sequence of elementary image processing operations such as scaling, rotation, contrast shift, and smoothing, among others. The authors develop single weak "experts" to detect each such elementary operations. Thereafter, these weak classifiers are fused. The authors have used features borrowed from the Steganalysis literature (c.f., Sec. 3.5) such as image quality metrics [Avcibas et al. 2003], binary similarity measures [Avcibas et al. 2005], and high order separable quadrature mirror filters statistics [Lyu and Farid 2004]. The main limitation with such approach is that the elementary operations by themselves do not constitute doctoring operations. Hence, this approach needs to be used wisely to point out localized operations. In this case, abrupt brightness and contrast changes in regions in the host image may point to forgeries (for example, when splicing different images). However, local intrinsic changes need to be accounted for in order to reduce the high rate of false positives. Finally, for criminal forgeries, it is likely that the forger will seek to match the target and host images in such a way to reduce these subtleties.

Ng and Chang [2004] have proposed a feature-based binary classification system using high order statistics to detect image composition. For that, the authors use bicoherence features motivated by the effectiveness of the bicoherence features for human-speech splicing detection [Nemer et al. 2001]. Bicoherence is the third order correlation of three harmonically related Fourier frequencies of a signal $\Xi(\omega)$ (normalized bispectrum). The authors report an accuracy of $\approx 71\%$ on the Columbia Splicing data set [Columbia DVMM Research Lab. 2004]. The Columbia data set, however, is composed of small composite images without any kind of post-processing. Figure 10 depicts four images from the data set. Finally, it is worth noting that the bicoherence features calculation is a computationally intensive procedure, often $O(n^4)$ where n is the number of pixels of an image.

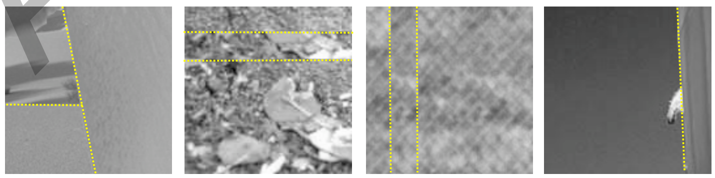


Fig. 10. Some examples from the Columbia Splicing data set. We emphasize the splicing boundaries in yellow.

Shi et al. [2007] propose a natural image model to separate spliced images from natural images. The model is represented by features extracted from a given set of test images and 2-D arrays produced by applying multi-size block discrete cosine transform (MBCT) to the given image. For each 2-D array, the authors calculate a prediction-error 2-D array, its wavelet sub-bands, and 1-D and 2-D statistical moments. In addition, the authors also calculate Markov transition probability matrices for the 2-D array differences that are taken as additional features. Although effective for simple image splicing procedures (copying and pasting) such as the ones in the Columbia Splicing data set with ≈ 92% accuracy, the approach does not seem to be effective for more sophisticated compositions that deploy adaptive edges and structural propagation [Sun et al. 2005]. This is because the transition matrices are often unable to capture the subtle edge variation upon structural propagation. In addition, such an approach is a binary-based solution; it does not point out possible forgery candidate regions.

3.4.4 *Inconsistencies in image features.* When splicing two images to create a composite, one often needs to re-sample an image onto a new sampling lattice using an interpolation technique (such as bi-cubic). Although imperceptible, the re-sampling contains specific correlations that, when detected, may represent evidence of tampering. Popescu and Farid [2005a] describe the form of these correlations, and proposes an algorithm for detecting them in an image. The authors showed that the specific form of the correlations can be determined by finding the neighborhood size, Φ, and the set of coefficients, $\vec{\beta}$, that satisfy: $\vec{M}_i = \sum_{j=-\Phi}^{\Phi} \beta_j \vec{M}_{i+j}$ in the equation

$$\left(\vec{M}_i - \sum_{j=-\Phi}^{\Phi} \beta_j \vec{M}_{i+j} \right) \cdot \vec{\mu} = 0, \tag{20}$$

where $\vec{\mu}$ is the signal, and $\vec{M}_i$ is the i^{th} row of the re-sampled matrix. The authors pointed out that, in practice, neither the samples that are correlated, nor the specific form of the correlations are known. Therefore, the authors employ an expectation maximization algorithm (EM) similar to the one in Section 3.3.1 to simultaneously estimate a set of periodic samples correlated to their neighbors and, an approximation form for these correlations. The authors assume that each sample belongs to one of two models. The first model $\mathcal{M}_1$, corresponds to those samples s_i that are correlated to their neighbors and are generated according to the following model:

$$\mathcal{M}_1 : s_i = \sum_{k=-\Phi}^{\Phi} \beta_k s_{i+k} + \mathcal{N}(i), \tag{21}$$

where $\mathcal{N}(i)$ denote independently, and identically distributed samples drawn from a Gaussian distribution with zero mean and unknown variance σ^2. In the E-step, the probability that each sample s_i belonging to model $\mathcal{M}_1$ can be estimated through Bayes rule similarly to the Equation 9, Section 3.3.2, where s_i replaces $f(x, y)$. The probability of observing a sample s_i knowing it was generated by $\mathcal{M}_1$ is calculated in the same way as in Equation 10, Section 3.3.2, where s_i again replaces $f(x, y)$. The authors claim that the generalization of their algorithm to color images is fairly straightforward. They propose to analyze each color channel independently. However, the authors do not show experiments for the performance of their algorithm under such circumstances and to what extent such an independence assumption is valid. Given that demosaiced color images present high pixel correlation, such analysis would be valuable.

It is assumed that the probability of observing samples generated by the outlier model, $\Pr\{s_i|s_i \in \mathcal{M}_2\}$, is uniformly distributed over the range of possible values of s_i. Although, it might seem a strong assumption, the authors do not go into more detail justifying the choice of the uniform distribution for this particular problem. In the M-step, the specific form of the correlations between samples is estimated minimizing a quadratic error function. It is important to note that the re-sampling itself does not constitute tampering. One could just save space by down-sampling every picture in a collection of pictures. However, when different correlations are present in one image, there is a strong indication of image composition. The authors have reported very good results for high-quality images. As the image is compressed, specially under JPEG 2000, the re-sampling correlates and hence tampering becomes harder to detect. It is worth noting that it is also possible to perform a counter attack anticipating the tampering detection and, therefore, destroying traces of re-sampling. Gloe et al. Gloe et al. [2007] present a targeted attack in which the pixel correlations are destroyed by small controlled geometric distortions. The authors superimpose a random disturbance vector to each individual pixel's position. To deal with possible jitter effects, the strength of distortion is adaptively modulated by the local image content using simple edge detectors.

3.4.5 *Lighting inconsistencies.* When creating a digital composite (for example, two people standing together), it is often difficult to match the lighting conditions from the individual photographs. Johnson and Farid [2005] present a solution that analyzes lighting inconsistencies to reveal traces of digital tampering. Standard approaches for estimating light source direction begin by making some simplifying assumptions such as: (1) the surface is Lambertian (it reflects light isotropically); (2) it has a constant reflectance value; (3) it is illuminated by a point light source infinitely far away; among others. However, to estimate the lighting direction, standard solutions require knowledge of the 3-D surface normals from, at least, four distinct points on a surface with same reflectance, which is hard to find from a single image and no objects of known geometry in the scene. The authors have used a clever solution first proposed by [Nillius and Eklundh 2001] that estimates two components of the light source direction from a single image. The authors also relax the constant reflectance assumption by assuming that the reflectance for a local surface patch is constant. This requires the technique to estimate individual light source directions for each patch along a surface. Figure 11(a) depicts an example where lighting inconsistencies can point out traces of tampering.

More recently, Johnson and Farid [2007a] extended this solution to complex lighting environments by using spherical harmonics. Under the aforementioned simplifying assumptions, an arbitrary lighting environment can be expressed as a non-negative function on the sphere, $L(\vec{A})$. $\vec{A}$ is a unit vector in Cartesian coordinates and the value of $L(\vec{A})$ is the intensity of the incident light along direction $\vec{A}$. If the object being illuminated is convex, the irradiance (light received) at any point on the surface is due only to lighting environment (no cast shadows or inter-reflections).

It is worth noting, however, that the assumptions of the authors limit the applicability of the algorithm. Real-world effects, such as interreflections, shadows, and changes in albedo can affect the estimates. In addition, sometimes there are cases where different lighting environments give rise to similar model coefficients, and therefore, the lighting differences are indistinguishable.

As a result, the irradiance, $\iota(\vec{U})$, can be parametrized by the unit length surface normal $\vec{U}$ and written as a convolution of the reflectance function on the surface, $\Lambda(\vec{A}, \vec{U})$, with the lighting environment $L(\vec{A})$:

$$\iota(\vec{U}) = \int_{\Omega} L(\vec{A})\Lambda(\vec{A}, \vec{U})d\Omega \tag{22}$$

where Ω represents the surface. For a Lambertian surface, the reflectance function is a clamped cosine:

$$\Lambda(\vec{A}, \vec{U}) = \max(\vec{A} \cdot \vec{U}, 0). \tag{23}$$

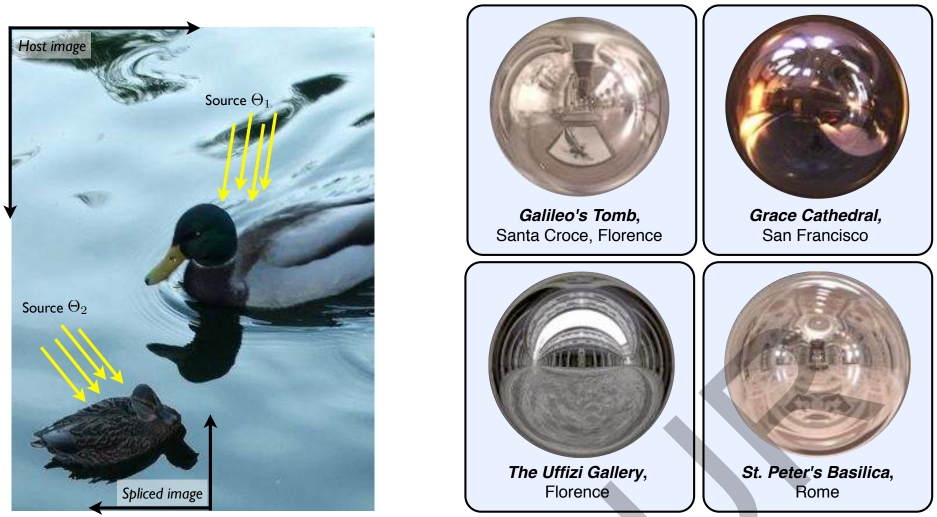
The convolution in Equation 22 can be simplified by expressing both the lighting environment and the reflectance functions in terms of spherical harmonics. The authors have validated their approach on a series of light probe images and showed good capability to estimate the lighting environment and its inconsistencies. A light probe image is an omnidirectional, high dynamic range image that records the incident illumination conditions at a particular point in space (see Figure 11(b)). Lighting environments can be captured by a variety of methods such as photographing a mirror sphere or through panoramic photographic techniques [Debevec 1998].

When analyzing the occluding contours of objects in real images, it is often the case that the range of surface normals is limited. Therefore, small amounts of noise in either the surface normals or the measured intensities can cause large variations in the estimation of the lighting environment [Johnson and Farid 2007a]. Finally, another drawback of this solution is that to inspect an image for forgery detection, the occluding contours of objects in the scene must be pointed out manually.

More recently, the work of Johnson and Farid [2007b] has also investigated lighting inconsistencies across specular highlights on the eyes to identify composites of people. The position of a specular highlight is determined by the relative positions of the light source, the reflective surface and the viewer (or camera). According to the authors, specular highlights that appear on the eye are a powerful cue as to the shape, color, and location of the light source(s). Inconsistencies in these properties of the light can be used as telltales of tampering. It is worth noting that specular highlights tend to be relatively small on the eye giving room to a more skilled forger to manipulate them to conceal traces of tampering. To do so, shape, color, and location of the highlight would have to be constructed so as to be globally consistent with the lighting in other parts of the image.

3.4.6 *Acquisition inconsistencies.* In the same way that we can use camera properties to point out the camera that captured an image, we also can use them as a digital X-ray for revealing forgeries [Chen et al. 2007].

Lin et al. [2005] present an approach that explores camera response normality and consistency functions to find tampering footprints. An image is tagged as doctored if the response functions are abnormal or inconsistent to each other. The camera response function is a mapping relationship between the pixel irradiance and the pixel value. For instance, suppose a pixel is on an edge and the scene radiance changes across the edge and is constant on both sides of the edge (Figure 12(a)).



(a) Composite example with lighting inconsistencies.

(b) Four light probes from different lighting environments. Credits to Paul Debevec and Dan Lemmon.

Fig. 11. Lighting and forgeries.

Therefore, the irradiance of the pixel on the edge should be a linear combination of those of the pixels clear off the edges (Figure 12(b)). Due to nonlinear response of the camera, the linear relationship breaks up among the read-out values of these pixels (Figure 12(c)). The authors estimate the original linear relationship when calculating the inverse camera response function [Lin et al. 2004]. Although effective in some situations, this approach has several drawbacks. Namely, (1) to estimate the camera response function, the authors must calculate an inverse camera response function which requires learning a Gaussian Mixture Model from a database with several known camera response functions (DoRF) [Lin et al. 2005]. If the analyzed image is a composite of regions from unknown cameras, the model is unable to point out an estimation for the camera response function; (2) the approach requires the user to manually select points on edges believed to be candidates for splicing; (3) the solution requires high contrast images to perform accurate edge and camera normality estimations; (4) the approach might fail if the spliced images are captured by the same camera and not synthesized along the edges of an object; (5) Finally, it is likely the solution does not work with CMOS adaptive sensors that dynamically calculate the camera response function to produce more pleasing pictures.

Chen et al. [2007] propose to use inconsistencies in the photo-response non-uniformity noise (c.f., Sec. 3.3.2) to detect traces of tampering. The method assumes that either the camera that took the image or at least some other pristine images taken by the camera are available. The algorithm starts by sliding a 128×128 block across the image and calculating the value of the test statistics, $p_{\mathcal{B}}$, for each block $\mathcal{B}$. The probability distribution function $pdf(x|H_0)$ of $p_{\mathcal{B}}$ under H_0 is estimated by correlating the PRNU noise residuals from other cameras and is modeled as a

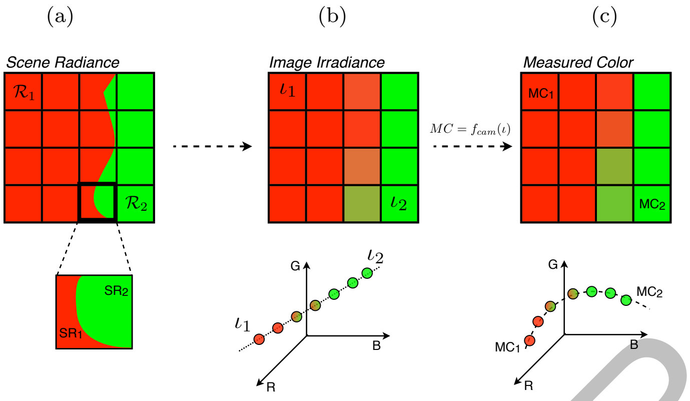


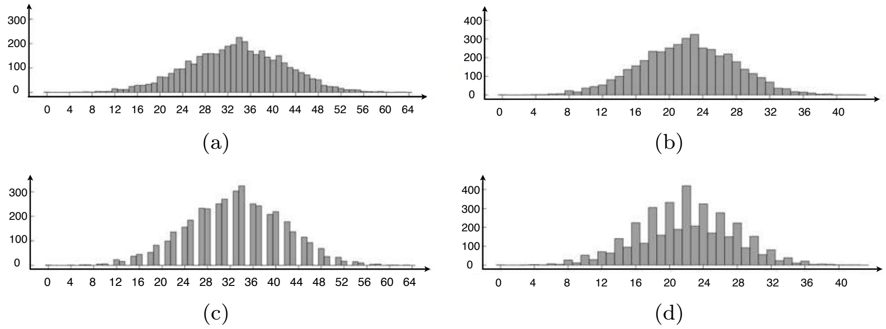
Fig. 12. Camera Response Function Estimation. (a) $\mathcal{R}_1$ and $\mathcal{R}_2$ are two regions with constant radiance. The third column images are a combination of $\mathcal{R}_1$ and $\mathcal{R}_2$. (b) The irradiances of pixels in $\mathcal{R}_1$ map to the same point ι_1, in RGB color space. The same happens for pixels in $\mathcal{R}_2$ which maps to ι_2. However, the colors of the pixels in the third column results from the linear combination of ι_1 and ι_2. (c) The camera response function f_{cam} warps the line segment in (b) into a curve during read-out.

generalized Gaussian. For each block, $pdf(x|H_1)$ is obtained from a block correlation predictor and is also modeled as a generalized Gaussian. For each block $\mathcal{B}$, the authors perform Neyman-Pearson hypothesis testing by fixing the false alarm rate Fa and decide that $\mathcal{B}$ has been tampered if $p_{\mathcal{B}} < Th$. The threshold Th is determined from the condition $Fa = \int_{Th} pdf(x|H_0)dx$.

3.4.7 *JPEG inconsistencies.* Some forgery detection approaches are devised specifically for a target. Popescu and Farid [2004b] discuss the effects of double quantization for JPEG images and presents a solution to detect such effects. Double JPEG compression introduces specific artifacts not present in single compressed images. The authors also note that evidence of double JPEG compression, however, does not necessarily prove malicious tampering. For example, it is possible for a user to simply re-save a high quality JPEG image with a lower quality[8]. Figure 3.4.7 depicts an example of the double quantization effect over a 1-d toy example signal $\mu[t]$ normally distributed in the range $[0, 127]$.

Inspired by the pioneering work of [Popescu and Farid 2004b] regarding double quantization effects and their use in forensics, He et al. [2006] propose an approach to locate doctored parts in JPEG images by examining the double quantization effect hidden among DCT coefficients. The idea is that as long as a JPEG image contains both the doctored part and the pristine part, the discrete cosine coefficient histograms of the pristine part will still have the double quantization effect (DQ), because this part of the image is the same as that of the double compressed original JPEG image. However, the histograms of a doctored part will not have the same DQ effects, if the doctored part is taken from a different image format, or different JPEG image. Some possible reasons for these observations are: (1) absence of

[8] A brief summary of JPEG compression is presented in Appendix A.



(a) (b)

(c) (d)

Fig. 13. The top row depicts histograms of single quantized signals with steps 2 (left) and 3 (right). The bottom row depicts histograms of double quantized signals with steps 3 followed by 2 (left), and 2 followed by 3 (right). Note the periodic artifacts in the histograms of double quantized signals. Credits to Alin Popescu.

the first JPEG compression in the doctored part; (2) mismatch of the DCT grid of the doctored part with that of the pristine part; or (3) composition of DCT blocks along the boundary may carry traces of the doctored and pristine parts given that it is not likely that the doctored part exactly consists of 8×8 blocks. It is worth noting, however, that this solution will not work in some circumstances. For instance, if the original image to contribute to the pristine part is not a JPEG image, the double quantization effect of the pristine part cannot be detected. In addition, the compression levels also affect the detection. Roughly speaking, the smaller the ratio of the second quantization step with respect to the first one, the harder the detection of the DQ effects. Finally, if the forger re-samples the grid of the DCT (shift the image one pixel), it is possible to destroy the traces of the double quantization and generate a complete new quantization table.

3.5 Image and video hidden content detection/recovery

Steganography is the art of secret communication. Its purpose is to hide the presence of communication — a very different goal than *Cryptography*, which aims to make communication unintelligible for those that do not possess the correct access rights [Anderson and Petitcolas 1998].

Applications of Steganography can include feature location (identification of subcomponents within a data set), captioning, time-stamping, and tamper-proofing (demonstration that original contents have not been altered). Steganography and Steganalysis have received a lot of attention around the world in the past few years [Rocha and Goldenstein 2008]. Unfortunately, not all applications are harmless; strong indications exist that Steganography has been used to spread child pornography on the Internet [Hart 2004; Morris 2004], and as an advanced communication tool for terrorists and drug-dealers [Herald Sun 2008; Folha de São Paulo 2008]. In the aftermath of the 9/11 events, some researchers have suggested that the Al Qaeda network used Steganography techniques to coordinate the World Trade Center attacks. Almost six years later, nothing was proved [Wallich 2003; Cass 2003; Kumagai 2003; Rocha and Goldenstein 2008]. Indeed, according to the *High Technology Crimes Annual Report* [USPS 2003; NHTCU 2008], Steganography threats can also appear in conjunction with dozens of other cyber-crimes such as: fraud and theft, computer cracking, online defamation, intellectual property offenses, and online harassment.

In response to such problems, the forensic analysis of such systems is paramount. We refer to *Forensic Steganalysis* as the area related to the detection and recovery of hidden messages. In this forensic scenario, we want to distinguish *non-stego* or *cover objects*, those that do not contain a hidden message, and *stego-objects*, those that contain a hidden message with the additional requirement of recovering its content as a possible proof basis for the court.

In the following sections, we present representative research with respect to the identification and recovery of hidden messages in digital multimedia. When possible, we emphasize approaches that can be used as an aid for criminal prosecution in a court of law. The fundamental goal of Steganalysis is to reliably detect the existence of hidden messages in communications and, indeed, most of the approaches in the literature have addressed only the detection problem. However, for forensics purposes, we are interested in the higher level of analysis going one step further and attempting to recover the hidden content.

We can model the detection of hidden messages in a cover medium as a classification problem. In Steganalysis, we have two extreme scenarios: (1) Eve, an eavesdropper, has only some level of suspicion that Alice and Bob are covertly communicating; and (2) Eve may have some additional information about Alice and Bob's covert communications such as the algorithm they have used, for instance. In the first case, we have a difficult forensic scenario where Eve would need to deploy a system able to detect all forms of Steganography (*Blind Steganalysis*). In the latter case, Eve might have additional information reducing her universe of possible hiding algorithms and cover media (*Targeted Steganalysis*).

In general, steganographic algorithms rely on the modification of some component of a digital object with a pseudo-random secret message [Anderson and Petitcolas 1998]. In digital images, common components used to conceal data are: (1) the least significant bits (LSBs); (2) DCT coefficients in JPEG-compressed images; and (3) areas with richness in details [Cox et al. 2008].

Figure 14 depicts a typical Steganography and Steganalysis scenario. When embedding a message in an image, one can take several steps in order to avoid message detection such as choosing an embedding key, compressing the message, and applying statistical profiling in the message and the cover media. On the other hand, in the Steganalysis scenario, we can try to point out the concealment whether performing statistical analysis on the input image, or on the image and on a set of positive and negative training examples. If we have additional information, this can be used to perform a targeted attack. In the following sections, we present some approaches used to detect such activities using either targeted or blind attacks.

3.5.1 *Targeted Steganalysis.* Some successful approaches for targeted Steganalysis proposed in the literature can estimate the embedding ratio or even reveal the secret message with the knowledge of the steganographic algorithm being very useful for forensics.

Basic LSB embedding can be reliably detected using the histogram attack as proposed by [Westfeld and Pfitzmann 1999]. Any possible LSB embedding procedure

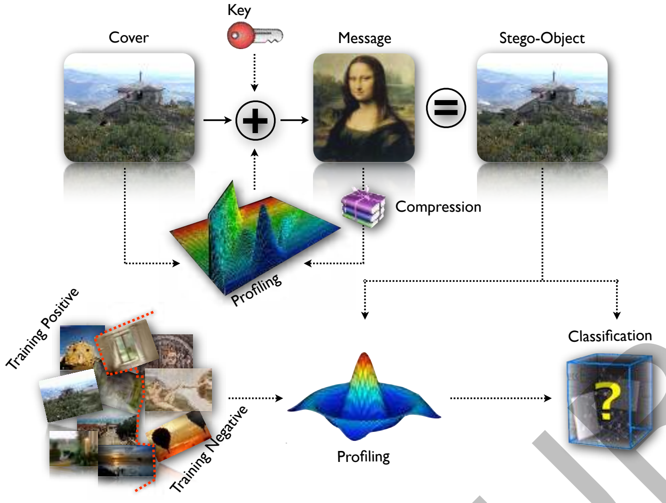


Fig. 14. Typical Steganography and Steganalysis scenario.

will change the contents of a selected number of pixels and therefore will change the pixel value statistics in a local neighborhood.

A K-bit color channel can represent 2^K possible values. If we split these values into 2^{K-1} pairs that only differ in the LSBs, we are considering all possible patterns of neighboring bits for the LSBs. Each of these pairs are called *pair of value* (PoV) in the sequence [Westfeld and Pfitzmann 1999].

When we use all the available LSB fields to hide a message in an image, the distribution of odd and even values of a PoV will be the same as the 0/1 distribution of the message bits. The idea of the statistical analysis is to compare the theoretically expected frequency distribution of the PoVs with the real observed ones [Westfeld and Pfitzmann 1999]. However, we do not have the original image and thus the expected frequency. In the original image, the theoretically expected frequency is the arithmetical mean of the two frequencies in a PoV. As we know, the embedding function only affects the LSBs, so it does not affect the PoV's distribution after an embedding. Therefore the arithmetical mean remains the same in each PoV, and we can derive the expected frequency through the arithmetic mean between the two frequencies in each PoV.

As presented in [Provos and Honeyman 2001; Westfeld and Pfitzmann 1999], we can apply the χ^2 (chi squared-test) T over these PoVs to detect hidden messages

$$T = \sum_{i=1}^{k} \frac{(freq_i^{obs} - freq_i^{exp})^2}{freq_i^{exp}}, \tag{24}$$

where k is the number of analyzed PoVs, $freq_i^{obs}$ and $freq_i^{exp}$ are the observed frequencies and the expected frequencies respectively. A small value of T points out that the data follows the expected distribution and we can conclude that the image was tweaked. We can measure the statistical significance of T by calculating the PR-value, which is the probability that a chi-square distributed random variable with $k-1$ degrees of freedom would attain a value larger than or equal to T:

$$\Pr(T) = \frac{1}{2^{\frac{k-1}{2}}\Gamma(\frac{k-1}{2})}\int_{T}^{\infty} e^{\frac{-x}{2}} x^{\frac{k-1}{2}-1} dx. \tag{25}$$

If the image does not have a hidden message, T is large and $\Pr(T)$ is small. In practice, we calculate a threshold value T_{th} so that $\Pr(T_{th}) = \eta$ where η is the chosen significance level. The main limitation with this approach is that it only detects sequential embeddings. For random embeddings, we could apply this approach window-wise. However, in this case it is effective only for large embeddings such as the ones that modify, at least, 50% of the available LSBs. For small embeddings, there is a simple counter-attack that breaks down this detection technique. For that, it is possible to learn the basic statistics about the image and to keep such statistics when embedding the message. For instance, for each bit modified to one, another one is flipped to zero. Indeed, as we shall show later, Outguess[9] is one approach that uses such tricks when performing embeddings in digital images.

Fridrich et al. [2001] present RS analysis. It consists of the analysis of the LSB lossless embedding capacity in color and gray-scale images. The lossless capacity reflects the fact that the LSB plane — even though it looks random — is related to the other bit planes [Fridrich et al. 2001]. Modifications in the LSB plane can lead to statistically detectable artifacts in the other bit planes of the image. The authors have reported good results (detection for message-sizes as small as $\approx 2-5\%$ on a limited set of images for the Steganography tools: Steganos, S-Tools, Hide4PGP, among others[10].

A similar approach was devised by [Dumitrescu et al. 2002] and is known as *sample pair analysis*. Such an approach relies on the formation of some subsets of pixels whose cardinalities change with LSB embedding, and such changes can be precisely quantified under the assumption that the embedded bits form a random walk on the image. Consider the partitioning of an input image in vectorized form U into pairs of pixels (p_u, p_v). Let $\mathcal{P}$ be the set of all pairs. Let us partition $\mathcal{P}$ into three disjoint sets X, Y, and Z, where

$$\begin{aligned}
X &= \{(p_u, p_v) \in \mathcal{P} \mid (p_v \text{ is even and } p_u < p_v) \text{ or } (p_v \text{ is odd and } p_u > p_v)\ \} \\
Y &= \{(p_u, p_v) \in \mathcal{P} \mid (p_v \text{ is even and } p_u > p_v) \text{ or } (p_v \text{ is odd and } p_u < p_v)\ \} \\
Z &= \{(p_u, p_v) \in \mathcal{P} \mid (p_u = p_v)\}
\end{aligned} \tag{26}$$

Furthermore, let us partition the subset Y into two subsets, W, and V, where $V = Y \setminus W$, and

$$Y = \{(p_u, p_v) \in \mathcal{P} \mid (p_u = 2k, p_v = 2k+1) \text{ or } (p_u = 2k+1, p_v = 2k)\} \tag{27}$$

The sets X, W, V, and Z are called primary sets and $\mathcal{P} = X \cup W \cup V \cup Z$. When one embeds content in an image, the LSB values are altered and therefore the cardinalities of the sets will change accordingly. As we show in Figure 15, we have four possible cases $\pi \in \{00, 01, 10, 11\}$. Let p be the relative amount of modified pixels in one image due to embedding. Hence, the probability of a state change is

[9] `http://www.outguess.org/`

[10] `http://members.tripod.com/steganography/stego/software.html`

given by

$$\begin{aligned}\rho(00,\mathcal{P}) &= (1-p/2)^2\\ \rho(01,\mathcal{P}) &= \rho(10,\mathcal{P}) = p/2(1-p/2)^2\\ \rho(11,\mathcal{P}) &= (p/2)^2. \end{aligned} \tag{28}$$

and the cardinalities after the changes are

$$\begin{aligned}|X'| &= |X|(1-p/2)+|V|p/2\\ |V'| &= |V|(1-p/2)+|X|p/2\\ |W'| &= |W|(1-p+p^2/2)+|Z|p(1-p/2) \end{aligned} \tag{29}$$

It follows that

$$|X'|-|V'| = (|X|-|V|)(1-p). \tag{30}$$

The authors have empirically noted that, on average, for natural images (no hidden content) $|X| = |Y|$. Therefore,

$$|X'|-|V'| = |W|(1-p). \tag{31}$$

Observe in Figure 15 that the embedding process does not alter $W \cup Z$. Hence, we define $\gamma = |W|+|Z| = |W'|+|Z'|$ yielding

$$|W'| = (|X'|-|V'|)(1-p)^2 + \gamma p(1-p/2). \tag{32}$$

Given that $|X'|+|V'|+|W'|+|Z'| = |\mathcal{P}|$, we have the estimation of the embedded content size

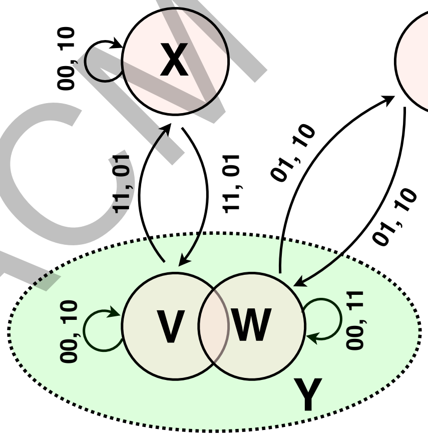


Fig. 15. Transitions between primary sets under LSB changing.

$$0.5\gamma p^2 + (2|X'|-|\mathcal{P}|)p + |Y'|-|X'| = 0. \tag{33}$$

This approach has been tested in [Cox et al. 2008] over three data sets summing up to 5,000 images. The data sets comprise raw, compressed, and also scanned images.

The approach is able to detect messages as small as 5% of the available space for normal LSB embedding with no statistical profiling.

Ker [2007b] has studied the statistical properties of the analysis of pairs and also proposes an extension using weighted least squares. Recently, Bohme [2008] presented an extension for JPEG covers. Several other approaches have been designed to detect targeted Steganalysis specifically in the JPEG domain [Fridrich 2004; Pevny and Fridrich 2005; Fu et al. 2006].

Shi et al. [2003] have analyzed the gradient energy flipping rate during the embedding process. The hypothesis is that the gradient energy varies consistently when the image is altered to conceal data.

For most of the above techniques, the authors do not discuss possible counter-attacks to their solutions. For instance, the sample pairs solution [Dumitrescu et al. 2002] and the RS analysis [Fridrich et al. 2001] rely on the analysis of groups of modified and non-modified pixels. What happens if someone knows these detection solutions and compensates for the group distribution for each modified pixel? Do the solutions still work after such kind of statistical profiling?

3.5.2 *Blind Steganalysis.* Most of the blind- and semi-blind detection approaches rely on supervised learning techniques. The classifiers used in existing blind and semi-blind Steganalysis refer to virtually all categories of classical classification such as regression, multi-variate regression, one class, two class, and hyper-geometric classifications, among others.

Both in blind and semi-blind scenarios, the classifier is a mapping that depends on one or more parameters that are determined through training and based on the desired tradeoff between both type of errors (false accept and false reject) that the classifier can make. Therefore, Steganalysis begins with the appropriate choice of features to represent both the stego and non-stego objects.

In the semi-blind scenario, we select a set of stego algorithms and train a classifier in the hope that when analyzing an object concealing a message embedded with an unknown algorithm, the detector will be able to generalize. On the other hand, in the complete blind scenario, we only train a set of cover objects based on features we believe will be altered during the concealment of data. In this case, we train one-class classifiers and use the trained model to detect outliers.

Some of the most common features used in the literature to feed classifiers are based on wavelet image decompositions, image quality metrics, controlled perturbations, moment functions, and histogram characteristic functions.

Lyu and Farid [2002a; 2002b] introduce a detection approach based on probability distribution functions of image sub-bands coefficients. This work has become a basis for several others. The motivation is that natural images have regularities that can be detected by high-order statistics through quadrature mirror filter (QMF) decompositions [Vaidyanathan 1987].

The QMF decomposition divides the image into multiple scales and orientations. We denote the vertical, horizontal, and diagonal sub-bands in a given scale $\{i = 1 \ldots n\}$ as $Vert_i(x,y)$, $Horiz_i(x,y)$, $Diag_i(x,y)$, respectively. Figure 16 depicts one image decomposition with three scales. Lyu and Farid [2002a; 2002b] propose to detect hidden messages using two sets of statistics collected throughout the multiple scales and orientations. The first set of statistics comprises *mean*, *variance*, *skewness*, and *kurtosis*. These statistics are unlikely to capture the strong

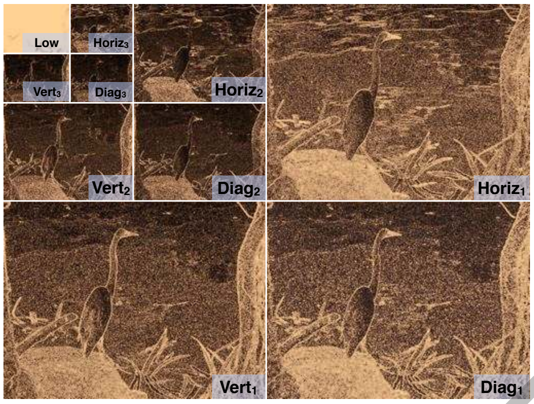


Fig. 16. Image sub-bands QMF decomposition.

correlations that exist across space, orientation, scale and color. Therefore, the authors calculate a second set of statistics based on the errors in a linear predictor of coefficient magnitude. For the sake of illustration, consider a vertical sub-band of a gray image at scale i, $Vert_i(x, y)$. A linear predictor for the magnitude of these coefficients in a subset of all possible spatial, orientation, and scale neighbors is given by

$$\begin{aligned} |Vert_i(x,y)| &= w_1|Vert_i(x-1,y)| + w_2|Vert_i(x+1,y)| + w_3|Vert_i(x,y-1)| \\ &+ w_4|Vert_i(x,y+1)| + w_5\left|Vert_{i+1}\left(\frac{x}{2},\frac{y}{2}\right)\right| + w_6|Diag_i(x,y)| \\ &+ w_7\left|Diag_{i+1}\left(\frac{x}{2},\frac{y}{2}\right)\right|, \end{aligned} \tag{34}$$

where $|\cdot|$ represents absolute value and w_j are the weights. We can represent this linear relationship in matrix form as $\vec{\mathcal{V}} = Q\vec{\mathcal{W}}$, where the column vector $\vec{\mathcal{W}} = (w_1, \ \ldots, w_7)^T$, the vector $\vec{\mathcal{V}}$ contains the coefficient magnitudes of $\mathcal{V}_i(x, y)$ strung out into a column vector, and the columns of the matrix Q contain the neighboring coefficient magnitudes as in Equation 34 also strung out into column vectors. The coefficients are determined through the minimization of the quadratic error function

$$\epsilon(\vec{\mathcal{W}}) = [\vec{\mathcal{V}} - Q\vec{\mathcal{W}}]^2. \tag{35}$$

This error is minimized through differentiation with respect to $\vec{\mathcal{W}}$. Setting the result equal to zero, and solving for $\vec{\mathcal{W}}$, we have

$$\vec{\mathcal{W}} = (Q^{\mathrm{T}}Q)^{-1}Q^{\mathrm{T}}\vec{\mathcal{V}}. \tag{36}$$

Finally, the log error in the linear predictor is given by

$$\vec{\epsilon} = \log_2 \vec{\mathcal{V}} - \log_2 (Q\vec{\mathcal{W}}). \tag{37}$$

It is from this error that the additional mean, variance, skewness, and kurtosis statistics are collected. This process is repeated for each sub-band, and scale. From this set of statistics, the authors train the detector with images with and without hidden messages.

Lyu and Farid [2004; 2006] have extended this set of features to color images and proposes a one-class classifier with hyper-spheres representing cover objects. Outliers of this model are tagged as stego objects. A similar procedure using Parzen-Windows was devised by [Rodriguez et al. 2007] to detect anomalies in stego systems.

Rocha and Goldenstein [2006; Rocha and Goldenstein [2010] have presented the *Progressive Randomization* meta-descriptor for Steganalysis. The principle is that it captures the difference between image classes (e.g., with and without hidden messages) by analyzing the statistical artifacts inserted during controlled perturbation processes with increasing randomness.

Avcibas et al. [2003] have presented a detection scheme based on image quality metrics (IQMs). The motivation is that the embedding can be understood as an addition of noise to the image therefore degrading its quality. They have used multivariate regression analysis for detection. Avcibas et al. [2005] have introduced an approach that explores binary similarity measures within image bit planes. The basic idea is that the correlation between the bit planes as well as the binary texture characteristics within the bit planes differ between a stego image and a cover image.

Histogram characteristic functions and statistics of empirical co-occurrence matrices also have been presented with relative success [Shi et al. 2005; Chen et al. 2006; Xuan et al. 2006; Xuan et al. 2005; Fridrich 2004].

Despite of all the advances, one major drawback of the previous approaches is that most of them are only able to point out whether or not a given image contains a hidden message. Currently, with classifier-based blind or semi-blind approaches it is extremely difficult or even impossible to identify portions of the image where a message is hidden and perform message extraction or even only point out possible tools used in the embedding process. A second drawback in this body of work is the lack of counter-analysis techniques to assess the viability of the existing research. Outguess[11] [Provos 2001] and F5 [Westfeld 2001] are two early examples of such works.

Outguess is a steganographic algorithm that relies on data specific handlers that extract redundant bits and write them back after modification. For JPEG images, Outguess preserves statistics based on frequency counts. As a result, statistical tests based on simple frequency counts are unable to detect the presence of steganographic content [Provos 2001]. Outguess uses a generic iterator object to select which bits in the data should be modified. In addition, F5 was proposed with the goal of providing high steganographic capacity without sacrificing security. Instead of LSB flipping (traditional embedding approaches), the embedding operation in F5 preserves the shape of the DCT histogram. The embedding is performed according to a pseudo-random path determined from a user pass-phrase. Later on, Fridrich et al. [2002] provided a targeted attack that detects messages embedded with the F5 algorithm throughout a process called calibration. With this approach, we estimate the original cover-object from the suspected stego-object. In the case of JPEG images, for instance, this is possible because the quantized DCT coefficients are robust to small distortions (the ones performed by some steganographic algorithms) [Cox et al. 2008]. The approach of [Fridrich et al. 2002] is no longer as

[11] http://www.outguess.org/

effective if we improve F5 with some sort of statistical profiling preserving not only the DCT histogram shape but also compensating for the modified coefficients.

Much more work of this sort is essential, given that this scenario looks like an *arms race* in which Steganographers and Steganalyzers compete to produce better approaches in a technological escalation.

In the Stegi@Work section, we present a common framework that allows us to combine most of the state of the art solutions in a compact and efficient way toward the objective of recovering the hidden content.

Some other flaws related to the classifier-based blind or semi-blind approaches are

- The choice of proper features to train the classifier upon is a key step. There is no systematic rule for feature selection. It is mostly a heuristic, trial and error method [Chandramouli and Subbalakshmi 2004].
- Some classifiers have several parameters that have to be chosen (type of kernels, learning rate, training conditions) making the process a hard task [Chandramouli and Subbalakshmi 2004].
- To our knowledge, a standard reference set has yet to emerge in the Steganalysis field to allow fair comparison across different approaches. One step in that direction is the work of [Rocha et al. 2008] which presents two controlled data sets to test hidden message detection approaches and the work of [Ker 2007a] which presents a new benchmark for binary steganalysis methods.

3.5.3 *Stegi@Work.* What is needed for today's forensics applications is a scalable framework that is able to process a large volume of images (the sheer volume of images on sites such as Flickr and Picasa is testament to this). As we have repeatedly seen throughout this paper, individual techniques for forensic analysis have been developed for specific tools, image characteristics, and imaging hardware, with results presented in the limited capacity of each individual work's focus. If a high capacity framework for digital image and video forensics was available, the forensic tools presented in this paper could be deployed in a common way, allowing the application of many tools against a candidate image, with the fusion of results giving a high-confidence answer as to whether an image contains steganographic content, is a forgery, or has been produced by a particular imaging system. In our own work in the "Vision of the Unseen," we have focused on the development of a cross-platform distributed framework specifically for Steganalysis, embodying the above ideas, that we call *Stegi@Work*. In this section, we will summarize the overall architecture and capabilities of the Stegi@Work framework as an example of what a distributed forensics framework should encompass.

Stegi@Work, at the highest architectural level (details in Figure 17), consists of three entities. A requester client issues jobs for the system to process. Each job consists of a file that does or does not contain steganographic content. This file is transmitted to the Stegi server, which in turn, dispatches the job's processing to the worker clients. Much like other distributed computing frameworks such as *Seti@home*[12] and *Folding@home*[13], worker clients can be ordinary workstations on

[12] http://setiathome.berkeley.edu/

[13] http://folding.stanford.edu/

a network with CPU cycles to spare. The Stegi server collects the results for each job, and performs fusion over the set of results, to come to a final conclusion about the status of the file in question. Each network entity may be connected via a LAN, or logically separated by firewalls in a WAN, facilitating the use of worker clients or requestor clients on a secure or classified network, while maintaining presence on an insecure network, such as the Internet. The Stegi server exists as the common point of contact for both.

The specifics of job communication (details in Figure 18), include the specific definitions for each job packet transmitted between network entities. Between the requester client and the Stegi server, both job request and job results packets are exchanged. In a job request, the file in question is transmitted to the server, along with optional tool selection and response requests. If these are not specified, the server can choose them automatically based on the type of the submitted file, as well as a defined site policy. The server receives a detailed report packet from each worker client, including the results of all of the tools applied against a file, as well as additional details about the job, such as execution time. Additional status packets are transmitted between all network entities, including server status to a worker client, notifying it that a job (with the file and appropriate tools) is ready, worker client status to the server, indicating the current state of a job, and server status to a worker client indicating what should be known about a job that is in the system.

The Stegi@Work architecture provides tool support for each worker client in the form of a wrapper API around the tool for each native platform. This API defines process handling, process status, and control signaling, allowing the Stegi server full control over each process on each worker client. The current system as implemented supports wrappers written in C/C++, Java, and Matlab, thus supporting a wide range of tools on multiple platforms. Network communication between each native tool on the worker client and the Stegi@Work system is defined via a set of XML messages. We have created wrappers for the popular analysis tools stegdetect[14] and Digital Invisible Ink Toolkit[15], as well as a custom tool supporting signature-based detection, as well as the statistical χ^2 test.

In order for high portability, allowing for many worker clients, the Stegi@Work framework has been implemented in Java, with tool support, as mentioned above, in a variety of different languages. This is accomplished through the use of Java Native Interface[16] (JNI), with Win32 and Linux calls currently supported. The Stegi@Work server is built on top of JBOSS[17], with an Enterprise Java Beans[18] (EJB) 3.0 object model for all network entities. GUI level dialogues are available for system control at each entity throughout the framework.

The actual use cases for a system like Stegi@Work extend beyond large-scale forensics for intelligence or law enforcement purposes. Corporate espionage remains a serious threat to business, with loss estimates as high as $200 billion[19]. An

[14] http://www.outguess.org/detection.php

[15] http://diit.sourceforge.net/

[16] http://swik.net/JNI+Tutorial

[17] http://www.jboss.org/

[18] http://www.conceptgo.com/gsejb/index.html

[19] http://news.bbc.co.uk/2/hi/technology/5313772.stm

enterprise can deploy requestor clients at the outgoing SMTP servers to scan each message attachment for steganographic content. If such content is detected, the system can quarantine the message, issue alerts, or simply attempt to destroy [Johnson and Jajodia 1998; Petitcolas et al. 1998] any detected content automatically, and send the message back on its way. This last option is desirable in cases where false positives are more likely, and thus, a problem for legitimate network users. Likewise, a government agency may choose to deploy the system in the same manner to prevent the theft of very sensitive data.

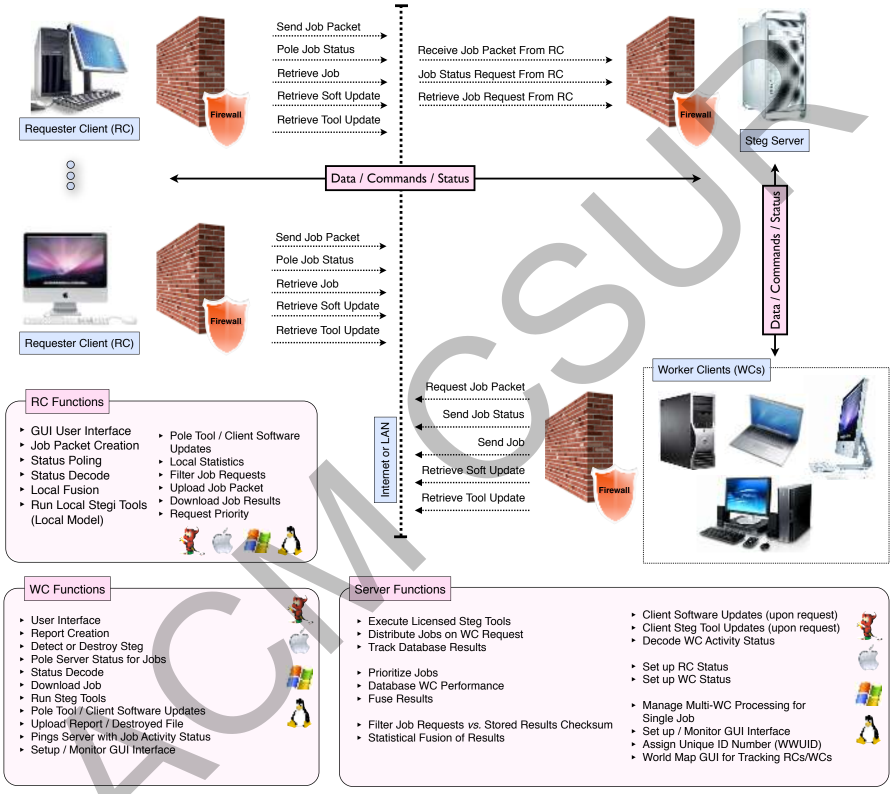


Fig. 17. Stegi@Work overall architecture.

4. CONCLUSIONS

A remarkable demand for image- and video- based forensics has emerged in recent years in response to a growing need for investigative tools for a diverse set of needs. From the law enforcement community's perspective, image based analysis is crucial for the investigation of many crimes, most notably child pornography. Yet, crime that utilizes images is not limited to just pornography, with entities as diverse as Colombian drug cartels taking advantage of steganography to mask their activities. From the intelligence community's perspective, the ability to scan large amounts of

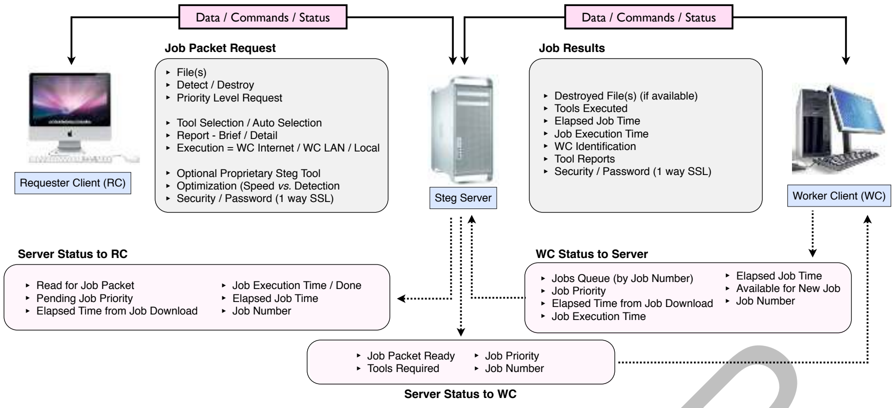


Fig. 18. Stegi@Work communications architecture.

secret and public data for tampering and hidden content is of interest for strategic national security. As the case of the Iranian missiles has shown, state based actors are just as willing to abuse image processing as common criminals.

But the obvious crimes are not necessarily the most damaging. The digital world presents its denizens with a staggering number of images of dubious authenticity. Disinformation via the media has been prevalent throughout the last century, with doctored images routinely being used for political propaganda. But now, with the near universal accessibility of digital publishing, disinformation has spread to commercial advertising, news media, and the work of malicious pranksters. Is it at all possible to determine whether an image is authentic or not? If we cannot determine the authenticity, what are we to believe about the information the image represents?

Digital Image and Video Forensics research is an important emerging field in computing that acts as a countermeasure to the intentional misuse of digital image editing tools. As we have seen in this survey, the main objectives of digital image and video forensics include tampering detection (cloning, healing, retouching, splicing), hidden messages detection/recovery, and source identification with no prior measurement or registration of the image (the availability of the original reference image or video). We have taken a look at many individual algorithms and techniques designed for very specific detection goals. However, the specific nature of the entire body of digital image and video forensics work is its main limitation at this point in time. How is an investigator able to choose the correct method for an image at hand? Moreover, the sheer magnitude of images that proliferate throughout the Internet poses a serious challenge for large-scale hidden content detection or authenticity verification.

In response to this challenge, we make several recommendations for researchers working in this field. First, work on decision level and temporal fusion serves as an excellent basis for operational systems. Combining information from many algorithms and techniques yields more accurate results — especially when we do not know precisely what we are looking for. Second, the need for large distributed (or clustered) systems for parallel evaluation fills an important role for national and corporate security. Our Stegi@Work system is an example of this. Third, the evaluation of existing and new algorithms must be improved. The analysis of detection results in nearly all papers surveyed lacks the rigor found in other areas of digital image processing and computer vision, making the assessment of their utility difficult. More troubling, in our survey, only a few papers on counter-forensics for image based forensics were found, leading us to question the robustness of much of the work presented here to a clever manipulator. Finally, for forgery detection and steganalysis, more powerful algorithms are needed to detect specifics about manipulations found in images, not just that an image has been tampered with. Despite these shortcoming, the advancement of the state of the art will continue to improve our *Vision of the Unseen*.

5. ACKNOWLEDGMENTS

This work was supported by FAPESP (Award Number 2008/08681-9, 05/58103-3, and 07/52015-0), CNPq (Award Number 309254/2007-8 and 551007/2007-9), and United States Air Force STTR (Award Number FA9550-05-C-0172).

A. DISCRETE COSINE TRANSFORM AND JPEG COMPRESSION

The discrete cosine transform (DCT) algorithm is one of the main components of the JPEG compression technique [Gonzalez and Woods 2007]. The JPEG standard specifies two compression schemes: a lossless predictive scheme and a lossy scheme based on the discrete cosine transform (DCT). For the lossy part, the most used technique is based on a subset of the DCT-based modes of operation. In this appendix, we present a summary of the baseline method. In general, JPEG compression works as follow

(1) **Split** the image into 8×8 blocks.

(2) **Transform each block via DCT**. This outputs a multi-dimensional array of 64 coefficients. For this intent, the pixels in the image samples grouped in the 8×8 blocks are shifted from unsigned to signed integers (i.e., from $[0, 255] \rightarrow [-128, 127]$). Thereafter, the DCT of the blocks is computed. Let $f(x, y)$ denote an 8×8 image block, then its DCT takes the form

$$\mathcal{D}(\omega_x, \omega_y) = \frac{1}{4} c(\omega_x) c(\omega_y) \sum_{x=0}^{7} \sum_{y=0}^{7} f(x, y) \cos \frac{(2x+1)\omega_x \pi}{16} \cos \frac{(2y+1)\omega_y \pi}{16}, \quad (38)$$

where $\omega_x, \omega_y = 0, \ldots, 7$, and $c(\omega) = \frac{1}{\sqrt{2}}$, for $\omega = 0$, and $c(\omega) = 1$ otherwise.

(3) **Use a lossy quantizer** to round each of the resulting coefficients. This is essentially the compression stage and it is where data is lost. Small unimportant coefficients are rounded to 0 while larger ones lose some of their precision. Quantization is a point-wise operation defined as a division by a quantization step followed by rounding to the nearest integer.

$$\mathcal{D}_{quant}(\omega_x, \omega_y) = \left\lfloor \frac{\mathcal{D}(\omega_x, \omega_y)}{s(\omega_x, \omega_y)} + \frac{1}{2} \right\rfloor, \omega_x, \omega_y = 0, \ldots, 7, \quad (39)$$

where $s(\omega_x, \omega_y)$ is a frequency-dependent quantization step and it is related to the JPEG compression quality. For more details on how to find $s(\omega_x, \omega_y)$ and the quantization tables used in JPEG compression, please refer to [Gonzalez and Woods 2007].

(4) **Use a lossless quantizer**. At this stage, the array of streamlined coefficients is further compressed using lossless entropy compression. The most frequently used procedure is Huffman coding, while arithmetic coding is also supported.

(5) **Decompressing**. To decompress, use the entropy decoding, de-quantization, and the inverse DCT procedures, in this order.

REFERENCES

Anderson, R. and Petitcolas, F. 1998. On the limits of steganography. *Journal of Selected Areas in Communications 16,* 4 (May), 474–481.

Avcibas, I., Kharrazi, M., Memon, N., and Sankur, B. 2005. Image steganalysis with binary similarity measures. *Journal on Applied Signal Processing 2005*, 2749–2757.

Avcibas, I., Memon, N., and Sankur, B. 2003. Steganalysis using image quality metrics. *IEEE Transactions On Image Processing 12*, 221–229.

Bayaram, S., Avcibas, I., Sankur, B., and Memon, N. 2006. Image manipulation detection. *Journal of Electronic Imaging 15,* 4 (October), 1–17.

Bayram, S., Sencar, H., and Memon, N. 2005. Source camera identification based on cfa interpolation. In *Intl. Conf. on Image Processing*. IEEE, Genova, Italy.

Bayram, S., Sencar, H., and Memon, N. 2006. Improvements on source camera-model identiciation based on cfa interpolation. In *WG 11.9 Int. Conf. on Digital Forensics*. IFIP, Orlando, USA.

Bohme, R. 2008. Weighted stego-image steganalysis for jpeg covers. In *Intl. Workshop in Information Hiding*. Springer, Santa Barbara, USA.

Cass, S. 2003. Listening in. *IEEE Spectrum 40,* 4 (April), 32–37.

Castro, E. D. and Morandi, C. 1987. Registration of translated and rotated images using finite fourier transforms. *IEEE Transactions on Pattern Analysis and Machine Intelligence 9*, 700–703.

Celiktutan, O., Avcibas, I., Sankur, B., and Memon, N. 2005. Source cell-phone identification. In *Intl. Conf. on Advanced Computing and Communication*. Computer Society of India, Tamil Nadu, India.

Chandramouli, R. and Subbalakshmi, K. P. 2004. Current trends in steganalysis: a critical survey. In *Intl. Conf. on Control, Automation, Robotics and Vision*. IEEE, Kunming, China, 964–967.

Chen, M., Fridrich, J., Goljan, M., and Lukas, J. 2007. Source digital camcorder identification using sensor photo-response non-uniformity. In *SPIE Photonics West*. SPIE, San Jose, USA, 1G–1H.

Chen, M., Fridrich, J., Lukas, J., and Goljan, M. 2007. Imaging sensor noise as digital x-ray for revealing forgeries. In *Intl. Workshop in Information Hiding*. Springer, Saint Malo, France.

Chen, X., Wang, Y., Tan, T., and Guo, L. 2006. Blind image steganalysis based on statistical analysis of empirical matrix. In *Intl. Conf. on Pattern Recognition*. IAPR, Hong Kong, China, 1107?–1110.

Choi, K. S., Lam, E., and Wong, K. 2006. Automatic source camera identification using the intrinsic lens radial distortion. *Optics Express 14,* 24 (November), 11551–11565.

Coe, B. 1990. *The Birth of Photography: The Story of the Formative Years, 1800-1900*, 1 ed. Book Sales, –.

Cox, I. J., Miller, M. L., Bloom, J. A., Fridrich, J., and Kalker, T. 2008. *Digital Watermarking and Steganography*, 2 ed. Morgan Kauffman, Burlington, USA.

Debevec, P. 1998. Rendering synthetic objects into real scenes: bridging traditional and imgebased graphics with global illumination and high dynamic range photograph. In *ACM Siggraph*. ACM Press, Orlando, US, 189–198.

Dehnie, S., Sencar, T., and Memon, N. 2006. Identification of computer generated and digital camera images for digital image forensics,. In *Intl. Conf. on Image Processing*. IEEE, Atlanta, USA.

Dirik, A., Sencar, H., and Memon, N. 2008. Digital single lens reflex camera identification from traces of sensor dust. *IEEE Trans. On Inf. Forensics and Security 3,* 3, 539–552.

Dirik, E., Bayram, S., Sencar, T., and Memon, N. 2007. New features to identify computer generated images. In *Intl. Conf. on Image Processing*. IEEE, San Antonio, Texas.

Dumitrescu, S., Wu, X., and Memon, N. 2002. On steganalysis of random LSB embedding in continuous-tone images. In *Intl. Conf. on Image Processing*. Vol. 3. IEEE, Rochester, USA, 641–644.

Farid, H. 2006a. Digital doctoring: How to tell the real from the fake. *Significance 3,* 4, 162–166.

Farid, H. 2006b. Exposing digital forgeries in scientific images. In *Multimedia and Security Workshop*. ACM, Geneva, Switzerland.

Farid, H. 2009. Image forgery detection. *IEEE Signal Processing Magazine 26,* 2 (March), 16–25.

Fridrich, J. 2004. Feature-based steganalysis for jpeg images and its implications for future design of steganographic schemes. In *Intl. Workshop in Information Hiding*. Springer, Toronto, Canada, 67–81.

Fridrich, J., Goljan, M., and Du, R. 2001. Detecting LSB steganography in color and grayscale images. *IEEE Multimedia 8*, 22–28.

Fridrich, J., Hogea, D., and Goljan, M. 2002. Steganalysis of jpeg images: Breaking the f5 algorithm. In *Intl. Workshop in Information Hiding*. Springer-Verlag, Noordwijkerhout, Germany, 310–323.

Fridrich, J., Soukal, D., and Lukas, J. 2003. Detection of copy-move forgery in digital images. In *Digital Forensic Research Workshop*. DFRWS, Cleveland, USA.

Fu, D., Shi, Y., Zou, D., and Xuan, G. 2006. JPEG steganalysis using empirical transition matrix in block dct domain. In *Intl. Workshop on Multimedia and Signal Processing*. IEEE, Victoria, Canada, 310–313.

Gallagher, A. and Chen, T. 2008. Image authentication by detecting traces of demosaicing. In *Intl. CVPR Workshop on Vision of the Unseen*. IEEE, Anchorage, Alaska.

Geradts, Z., Bijhold, J., Kieft, M., Kurusawa, K., Kuroki, K., and Saitoh, N. 2001. Methods for identification of images acquired with digital cameras. In *Enabling Technologies for Law Enforcement and Security*. Vol. 4232. SPIE, –.

Gloe, T., Kirchner, M., Winkler, A., and Bohme, R. 2007. Can we trust digital image forensics? In *ACM Multimedia*. ACM, Augsburg, Germany, 78–86.

Goldenstein, S. and Rocha, A. 2009. High-profile forensic analysis of images. In *Intl. Conf. on Imaging for Crime Detection and Prevention*. IET, London, UK, 1–6.

Goljan, M. and Fridrich, J. 2008. Camera identification from scaled and cropped images. In *Proc. of the SPIE Electronic Imaging, Forensics, Security, Steganography, and Watermarking of Multimedia Contents*. SPIE, San Jose, USA, OE–1–OE–13.

Goljan, M., Fridrich, J., and Lukas, J. 2008. Camera identification from printed images. In *Proc. of the SPIE Electronic Imaging, Forensics, Security, Steganography, and Watermarking of Multimedia Contents*. SPIE, San Jose, USA, OI–1–OI–12.

Gonzalez, R. and Woods, R. 2007. *Digital Image Processing*, 3 ed. Prentice-Hall, Upper Saddle River, USA.

Hart, S. V. 2004. Forensic examination of digital evidence: a guide for law enforcement. Tech. Rep. NCJ 199408, National Institute of Justice NIJ-US, Washington DC, USA. September.

He, J., Lin, Z., Wang, L., and Tang, X. 2006. Detecting doctored jpeg images via dct coefficient analysis. In *European Conf. on Computer Vision*. Springer, Graz, Austria, 423–435.

Johnson, M. K. and Farid, H. 2005. Exposing digital forgeries by detecting inconsistencies in lighting. In *ACM Multimedia and Security Workshop*. ACM, New York, USA.

Johnson, M. K. and Farid, H. 2007a. Exposing digital forgeries in complex lighting environments. *IEEE Transactions on Information Forensics and Security 2,* 3, 450–461.

Johnson, M. K. and Farid, H. 2007b. Exposing digital forgeries through specular highlights on the eye. In *Intl. Workshop in Information Hiding*. Springer, Saint Malo, France.

Johnson, N. F. and Jajodia, S. 1998. Steganalysis of images created using current steganography software. In *Intl. Workshop in Information Hiding*. Springer-Verlag, Portland, Oregon.

K. Kurosawa, K. K. and Saitoh, N. 1999. Ccd fingerprint method. In *Intl. Conf. on Image Processing*. IEEE, Kobe, Japan.

Ker, A. 2007a. The ultimate steganalysis benchmark? In *Intl. Conf. on Multimedia & Security*. ACM, Dallas, USA, 141–148.

Ker, A. D. 2007b. Optimally weighted least-squares steganalysis. In *Steganography and Watermarking of Multimedia Contents*. SPIE, San Jose, USA.

Kharrazi, M., Sencar, H., and Memon, N. 2004. Blind source camera identification. In *Intl. Conf. on Image Processing*. IEEE, Singapore.

Kumagai, J. 2003. Mission impossible? *IEEE Spectrum 40,* 4 (April), 26–31.

Lin, S., Gu, J., Yamazaki, S., and Shum, H. Y. 2004. Radimetric calibration from a single image. In *Intl. Conf. on Computer Vision and Pattern Recognition*. IEEE, Washington, USA, 938–945.

Lin, Z., Wang, R., Tang, X., and Shum, H.-Y. 2005. Detecting doctored images using camera response normality and consistency. In *Intl. Conf. on Computer Vision and Pattern Recognition*. IEEE, New York, USA.

Long, Y. and Huang, Y. 2006. Image based source camera identification using demosaicing. In *Intl. Workshop on Multimedia Signal Processing*. IEEE, Victoria, Canada.

Lukas, J., Fridrich, J., and Goljan, M. 2006. Digital camera identification from sensor pattern noise. *IEEE Trans. On Inf. Forensics and Security 1,* 2, 205–214.

Lyu, S. and Farid, H. 2002a. Detecting hidden messages using higher-order statistics and support vector machines. In *Proceedings of the Fifth Intl. Workshop on Information Hiding*. Springer-Verlag, Noordwijkerhout, The Netherlands, 340–354.

Lyu, S. and Farid, H. 2002b. Detecting hidden messages using higher-order statistics and support vector machines. In *Intl. Workshop in Information Hiding*. Springer, Dresden, Germany, 340–354.

Lyu, S. and Farid, H. 2004. Steganalysis using color wavelet statistics and one-class support vector machines. In *Symposium on Electronic Imaging*. SPIE, San Jose, USA.

Lyu, S. and Farid, H. 2005. How realistic is photorealistic? *IEEE Transactions on Signal Processing 53,* 2 (March), 845–850.

Lyu, S. and Farid, H. 2006. Steganalysis using higher-order image statistics. *IEEE Transactions on Information Forensics and Security 1*, 111–119.

Marien, M. W. 2006. *Photography: A Cultural History*, 2 ed. Prentice Hall, –.

Columbia DVMM Research Lab. 2004. Columbia image splicing detection evaluation data set. Available at `http://www.ee.columbia.edu/ln/dvmm/downloads/AuthSplicedDataSet/AuthSplicedDataSet.htm`.

Errol Morris. The New York Times, July 11th, 2008. Photography as a weapon. Available at `http://morris.blogs.nytimes.com/2008/08/11/photography-as-a-weapon/index.html`.

Folha de São Paulo. April 5th, 2009. . Available at http://www1.folha.uol.com.br/fsp/brasil/fc0504200906.htm (In Portuguese).

Folha de São Paulo. March 10th, 2008. Para agência dos EUA, Abadía traficou no Brasil. Available at http://www1.folha.uol.com.br/fsp/cotidian/ff1003200801.htm (In Portuguese).

Herald Sun. March 11th, 2008. Hello Kitty was drug lord's messenger. Available at `http://www.news.com.au/heraldsun/story/0,21985,23354813-5005961,00.html`.

Mike Nizza and Patrick Witty. The New York Times, July 10th, 2008. In an iranian image, a missile too many. Available at `http://thelede.blogs.nytimes.com/2008/07/10/in-an-iranian-image-a-missile-too-many`.

USPS. 2003. USPS – US Postal Inspection Service. Available at `www.usps.com/postalinspectors/ar01intr.pdf`.

Willian J. Broad. The New York Times, May 4^{th}, 2009. A battle to preserve a visionary?s bold failure. Available at http://www.nytimes.com/2009/05/05/science/05tesla.html.

Morris, S. 2004. The future of netcrime now: Part 1 - threats and challenges. Tech. Rep. 62/04, Home Office Crime and Policing Group, Washington DC, USA.

Nemer, E., Goubran, R., and Mahmoud, S. 2001. Robust voice activity detection using higher-order statistics in the LPC residual domain. *IEEE Transactions on Speech and Audio Processing 9,* 3 (March), 217–231.

Ng, T.-T. and Chang, S.-F. 2004. Blind detection of photomontage using higher order statistics. In *Intl. Symposium on Circuits and Systems*. IEEE, Vancouver, Canada, 688–691.

Ng, T.-T. and Chang, S.-F. 2009. Identifying and prefiltering images. *IEEE Signal Processing Magazine 26,* 2 (March), 49–58.

Ng, T.-T., Chang, S.-F., Lin, C.-Y., and Sun, Q. 2006. *Multimedia Security Technologies for Digital Rights Management*. Academic Press, Burlington, USA, Chapter Passive-blind image Forensics, 1–30.

Ng, T.-T., Chang, S.-F., and Tsui, M.-P. 2005. Physics-motivated features for distinguishing photographic images and computer graphics. In *ACM Multimedia*. ACM, Singapore, 239–248.

NHTCU. 2008. National high tech crime unit. www.nhtcu.org.

Nillius, P. and Eklundh, J.-O. 2001. Automatic estimation of the projected light source direction. In *Intl. Conf. on Computer Vision and Pattern Recognition*. IEEE, Hawaii, US, 1076–1082.

Pearson, H. 2005. Image manipulation: Csi: Cell biology. *Nature 434*, 952–953.

Petitcolas, F. A. P., Anderson, R. J., and Kuhn, M. G. 1998. Attacks on copyright marking systems. In *Intl. Workshop in Information Hiding*. Springer-Verlag, Portland, Oregon, 219–239.

Pevny, T. and Fridrich, J. 2005. Toward multi-class blind steganalyzer for jpeg images. In *Intl. Workshop on Digital Watermarking*. Springer, Siena, Italy, 39–53.

Popescu, A. C. 2004. Statistical tools for digital image forensics. Ph.D. thesis, Department of Computer Science - Dartmouth College, Hanover, USA.

Popescu, A. C. and Farid, H. 2004a. Exposing digital forgeries by detecting duplicated image regions. Tech. Rep. TR 2004-515, Department of Computer Science - Dartmouth College, Hanover, USA.

Popescu, A. C. and Farid, H. 2004b. Statistical tools for digital forensics. In *Intl. Workshop in Information Hiding*. Springer, Toronto, Canada.

Popescu, A. C. and Farid, H. 2005a. Exposing digital forgeries by detecting traces of re-sampling. *IEEE Transactions on Signal Processing 53,* 2, 758–767.

Popescu, A. C. and Farid, H. 2005b. Exposing digital forgeries in color filter array interpolated images. *IEEE Transactions on Signal Processing 53,* 10, 3948–3959.

Provos, N. 2001. Defending against statistical steganalysis. In *Usenix Security Symposium*. Vol. 10. Usenix, Washington, USA, 24–36.

Provos, N. and Honeyman, P. 2001. Detecting steganographic content on the internet. Ann Arbor, USA CITI 01-11, Department of Computer Science - University of Michigan. August.

Rocha, A. and Goldenstein, S. 2006. Progressive Randomization for Steganalysis. In *Intl. Workshop on Multimedia and Signal Processing*. IEEE, Victoria, Canada, 314–319.

Rocha, A. and Goldenstein, S. 2007. PR: More than meets the eye. In *Intl. Conf. on Computer Vision*. IEEE, Rio de Janeiro, Brazil, 1–8.

Rocha, A. and Goldenstein, S. 2008. Steganography and Steganalysis in Digital Multimedia: Hype or Hallelujah? *Journal of Theoretical and Applied Computing (RITA) 15,* 1, 83–110.

Rocha, A. and Goldenstein, S. 2010. Progressive Randomization: Seeing the Unseen. *Computer Vision and Image Understanding*. To appear, 2010.

Rocha, A., Scheirer, W., Goldenstein, S., and Boult, T. E. 2008. The Unseen Challenge Data Sets. In *Intl. CVPR Workshop on Vision of the Unseen*. IEEE, Anchorage, USA, 1–8.

Rodriguez, B., Peterson, G., and Agaian, S. 2007. Steganography anomaly detection using simple one-class classification. In *Mobile Multimedia/Image Processing for Military and Security Applications*. SPIE, Orlando, USA, 65790E.1–65790E.9.

Sacchi, D. L. M., Agnoli, F., and Loftus, E. F. 2007. Changing history: Doctored photographs affect memory for past public events. *Applied Cognitive Psychology 21,* 8 (August), 249–273.

Sencar, T. and Memon, N. 2008. *Statistical Science and Interdisciplinary Research.* World Scientific Press, Mountain View, USA, Chapter Overview of State-of-the-art in Digital Image Forensics, –.

Shi, J. and Malik, J. 2000. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22,* 8 (August), 888–905.

Shi, L., Fen, S. A., and Xian, Y. Y. 2003. A LSB steganography detection algorithm. In *Personal, Indoor and Mobile Radio Communications.* Vol. 3. IEEE, Beijing, China, 2780–2783.

Shi, Y. Q., Chen, C., and Chen, W. 2007. A natural image model approach to splicing detection. In *ACM Multimedia and Security Workshop.* ACM, Dallas, USA, 51–62.

Shi, Y. Q., Xuan, G., Zou, D., Gao, J., Yang, C., Zhang, Z., Chai, P., Chen, W., and Chen, C. 2005. Image steganalysis based on moments of characteristic functions and wavelet decomposition, prediction, error-image, and neural network. In *Intl. Conf. on Multimedia and Expo.* IEEE, Amsterdam, The Netherlands, 268–272.

Sun, J., Yuan, L., Jia, J., and Shum, H.-Y. 2005. Image completion with structure propagation. *ACM Transactions on Graphics 24,* 3, 861–868.

Sutcu, Y., Bayaram, S., Sencar, H., and Memon, N. 2007. Improvements on sensor noise based source camera identification. In *Intl. Conf. on Multimedia and Expo.* IEEE, Beijing, China.

Swaminathan, A., Wu, M., and Liu, K. R. 2006. Non-instrusive forensics analysis of visual sensors using output images. In *Intl Conf. on Image Processing.* IEEE, Atlanta, USA.

Swaminathan, A., Wu, M., and Liu, K. R. 2009. Component forensics. *IEEE Signal Processing Magazine 26,* 2 (March), 38–48.

Tsai, M. and Wu, G. 2006. Using image features to identify camera sources. In *Intl. Conf. on Acoustics, Speech, and Signal Processing.* IEEE, Toulouse, France.

Vaidyanathan, P. P. 1987. Quadrature mirror filter banks, m-band extensions and perfect reconstruction techniques. *IEEE Signal Processing Magazine 4,* 3, 4–20.

Wallich, P. 2003. Getting the message. *IEEE Spectrum 40,* 4 (April), 38–40.

Wang, W. 2009. Digital video forensics. Ph.D. thesis, Department of Computer Science - Dartmouth College, Hanover, USA.

Wang, W. and Farid, H. 2007a. Exposing digital forgeries in interlaced and deinterlaced video. *IEEE Transactions on Information Forensics and Security 2*, 438–449.

Wang, W. and Farid, H. 2007b. Exposing digital forgeries in video by detecting duplication. In *ACM Multimedia and Security Workshop.* ACM, Dallas, USA.

Wang, Y. and Moulin, P. 2006. On discrimination between photorealistic and photographic images. In *Intl. Conf. on Acoustics, Speech, and Signal Processing.* IEEE, Toulouse, France.

Westfeld, A. 2001. F5 — a steganographic algorithm high capacity despite better steganalysis. In *Intl. Workshop in Information Hiding.* Vol. 2137. Springer-Verlag, Pittsburgh, US, 289–302.

Westfeld, A. and Pfitzmann, A. 1999. Attacks on steganographic systems. In *Intl. Workshop in Information Hiding.* Springer, Dresden, Germany, 61–76.

Xuan, G., Gao, J., Shi, Y., and Zou, D. 2005. Image steganalysis based on statistical moments of wavelet subband histograms in dft domain. In *Intl. Workshop on Multimedia Signal Processing.* IEEE, Shanghai, China, 1–4.

Xuan, G., Shi, Y., Huang, C., Fu, D., Zhu, X., Chai, P., and Gao, J. 2006. Steganalysis using high-dimensional features derived from co-occurrence matrix and class-wise non-principal components analysis (CNPCA). In *Intl. Workshop on Digital Watermarking.* IEEE, Jeju Island, South Korea, 49–60.

Yu, H., Ng, T.-T., and Sun, Q. 2008. Recaptured photo detection using specularity distribution. In *Intl. Conf. on Image Processing.* IEEE, San Diego, California.