

# Non-Intrusive Component Forensics of Visual Sensors Using Output Images

Ashwin Swaminathan, Min Wu, and K. J. Ray Liu

**Abstract**—Visual sensors have experienced a tremendous amount of growth and are becoming increasingly popular every year. Such rapid technology development and widespread use has led to a number of new problems related to protecting intellectual property rights, handling patent infringements, authenticating acquisition source, and identifying content manipulations. This paper introduces *non-intrusive component forensics* as a new methodology for forensic analysis. Non-intrusive component forensics aims at identifying the algorithms and parameters employed inside the various processing modules of a digital device, using only the sample data collected from device outputs without breaking the device apart. In this paper, we propose a novel methodology for non-intrusive forensic analysis of visual sensors, and develop techniques to estimate the algorithms and parameters employed by such important camera components as color filter array and color interpolation modules. The estimated interpolation coefficients provide useful features to build an efficient camera identifier to determine the brand/make from which an image was captured. The results obtained from such component analysis are also used to study the similarities between the technologies employed by different camera models to identify potential infringement/licensing and to facilitate studies on technology evolution.

**Index Terms:** Visual sensors, component forensics, non-intrusive image forensics, digital forensic signal processing, infringement/licensing forensics, evolutionary forensics, camera identification.

## I. INTRODUCTION

Visual sensors have experienced tremendous growth in recent decades. The resolution and quality of electronic imaging have been steadily improving, and digital cameras are becoming ubiquitous. Shipment of digital cameras alone has grown from \$46.4 million in 2003 to \$62 million in 2004, forming an approximately \$15 billion market worldwide [1]. Digital images taken by various imaging devices have been used in a growing number of applications, from military and reconnaissance to medical diagnosis and consumer photography. Consequently, a series of new forensic issues arise amidst such rapid advancement and widespread adoption of imaging technologies. For example, one can readily ask what kinds of hardware and software components as well as their parameters have been employed inside the devices? Given a digital image, which imaging sensor or which brand of sensors was used to acquire the image? What kinds of legitimate processing and undesired alteration have been applied to an image since it leaves the device?

There are various ways to address the questions at hand. The most challenging yet powerful approach is to answer them using the clues obtained from the output images without

having access to the devices – that is, by a non-intrusive approach. In this paper, we propose to develop a new forensic methodology called *non-intrusive component forensics*, which aims at identifying the components inside a visual device solely from its output data by inferring what algorithms/processing are employed and estimating their parameter settings. Furthermore, building upon component forensics, we extend these ideas to address a number of larger forensic issues in discovering technology infringement, protecting intellectual property rights, and identifying acquisition devices.

For centuries, intellectual property protection has played a crucial role in fostering innovation, as it has been known for “adding the fuel of interest to the fire of genius” since the time of Abraham Lincoln. Fierce competition in the electronic imaging industry has led to an increasing number of infringement cases filed in U.S. courts. The remunerations awarded to successful prosecution have also grown tremendously, sometimes in billions of dollars. For example, the Ampex Corporation has more than 600 patents related to digital cameras and recently received more than \$275-million compensation from lawsuits and settlements involving patent infringement cases with many digital camera vendors using one of its patents [2]. According to the U.S. patent law [3], *infringement of a patent* consists of the unauthorized making, using, offering for sale or selling any patented invention during the term of its validity. Patent infringement is usually difficult to detect, and even harder to prove in the court of law. The burden of proof often lies on patent holders, who are expected to provide solid evidence to substantiate their accusations. A common way to perform infringement analysis is to examine the design and implementation of a product and to look for similarities with what have been claimed in existing patents, through some type of reverse engineering. However, this approach could be very cumbersome and ineffective, for example, it may involve going over VHDL design codes of an IC chip in charge of core information processing tasks, which is a daunting task even to the most experienced expert in the field. Such analysis is often limited to the *implementation of the idea* rather than the *idea* itself, and thus could potentially lead to misleading conclusions [4] [5]. Component forensics is an important technology to detect patent infringement and protect intellectual property rights, by obtaining evidence about the algorithms employed in various components of the digital device.

Component forensics also serves as a foundation to establish the trustworthiness of imaging devices. With the fast development of tools to manipulate multimedia data, the integrity of both content and acquisition device has become particularly important when images are used as critical evidence in journalism, reconnaissance, and law enforcement applications. For example, information about hardware/software modules

and their parameters in a camera can help building camera identification systems. Such systems would provide useful *acquisition forensic* information to law enforcement and intelligence agencies about which camera or which brand of camera is used to acquire an image. Additionally, component forensics helps establish a solid model on the characteristics of image that is obtained directly from a camera. This in turn will facilitate *tampering forensics* to determine if there is any additional editing and processing applied to an image since it leaves the camera.

We can classify component forensics into three main categories, based on the nature of the available evidences:

- 1) **Intrusive Forensics:** A forensic analyst has access to the device in question and can disassemble it to carefully examine every part, including analyzing any available intermediate signals and states to identify the algorithms employed in each of its processing blocks.
- 2) **Semi Non-Intrusive Forensics:** An analyst has access to the device as a black box. He/she can design appropriate inputs to be fed into the device so as to collect forensic evidence about the processing techniques and parameters of the individual components inside.
- 3) **Non-Intrusive Forensics:** An analyst does not have access to the device in question. He/she is provided with some sample data produced by the device, and studies them to gather forensic evidence.

In this paper, we illustrate the proposed non-intrusive component forensic methodology for visual sensors, while the suggested techniques can be extended to other types of acquisition models, and sensing technologies, with appropriate modifications<sup>1</sup>. We focus on important camera components such as the color filter array (CFA) and the color interpolation algorithms. Our proposed techniques aim to determine the parameters of CFA and the interpolation algorithms using only sample output images obtained over diverse and uncontrolled input conditions. The features and acquisition models that we develop can be used to construct an efficient *camera identifier* that determines the brand/type of camera used to take the image. Further, our forensic algorithms can quantitatively help ascertain the similarities and differences among the corresponding camera components of different cameras. For devices from different vendors, the digital forensic knowledge obtained from such analysis can provide clues and evidence on technology infringement or licensing, which we shall refer to as *infringement/licensing forensics*, and will assist the enforcement of intellectual rights protection and foster technology innovation. For devices of the same brand but different models released at different years and/or at different price tiers, our analysis forms a basis of *evolutionary forensics*, as it can provide clues on technology evolution.

This paper is organized as follows. After reviewing related works in Section II, we present in Section III the image capturing process in digital cameras and our problem for-

mulation. In Section IV, we present methods to identify the CFA pattern and the color interpolation algorithm. We then illustrate proofs of concept with synthetic data in Section V-A and present results with a real data set of 19 cameras in Section V-B. The estimated model parameters are used to construct a camera identifier and to study the similarities (and differences) among the cameras in Section VI. Section VII generalizes the proposed methods to extend to other devices. The final conclusions are drawn in Section VIII.

## II. RELATED PRIOR WORKS ON FORENSIC ANALYSIS

While a growing amount of signal processing research in the recent years has been devoted to the security and protection of multimedia information (e.g. through encryption, hashing, and data embedding), forensic research on digital visual devices is still in its infancy. Related prior art on non-intrusive image forensics fall into the following two main categories.

In the forgery detection literature, there have been works that consider a tampered image has undergone a series of processing operations. Based on this observation, several methods were proposed to explore the salient features associated with each of these tampering operations, such as resampling [6], luminance and brightness change, copy-paste operations [7], and inconsistent noise patterns [8] or lighting [9]. For image compression such as JPEG that involves quantization in the discrete cosine transform (DCT) domain, statistical analysis based on binning techniques have been used to estimate the quantization matrices [10] [11]. Higher order statistics such as the bispectrum have been proposed to identify contrast changes, gamma correction [12], and other nonlinear point operations [13] on images; and wavelet based features have been used to detect image tampering [14], and identify photorealism [15]. Physics motivated features have been introduced to distinguish photographic images and computer graphics [16]. However, most of these techniques mentioned above are primarily targeted at finding the processing steps that occurs after the image has been captured by the camera, and are not for finding the algorithms and parameters used in various components inside the digital camera.

A second group of prior art on non-intrusive image forensics concerns camera identification. Camera pixel defects [17], pattern noise associated with the non-uniformity of dark currents on camera CCDs [18], and fixed pixel noise [19] inherent to an image sensor have been recently used as unique camera identifiers. While useful in some forensic tasks when a suspicious camera is available in the testing, this approach does not provide information about the internal components and cannot be used for identifying common features tied to the same camera models and makes. Another recent approach employs statistics from visually similar images taken with different cameras to train classifiers for identifying the camera source [20] [21]. Features such as average pixel values, RGB pairs correlation, and neighbor center of mass are used in [20]. In [21], the authors employ the Expectation Maximization algorithm introduced in [6] [22] to detect traces of color interpolation and resampling. The coefficient weights from the image and the peak magnitude and locations of the spectral

<sup>1</sup>For example, our recent work examined a modified imaging model to include digital scanners that handle color differently from cameras and have obtained good forensics results [38].

representation of the coefficients are used as features to build the camera brand classifier. Although good results were reported in distinguishing pictures taken in controlled scenarios with three different cameras, its ability to differentiate under diverse training sets, non-intrusive testing conditions, and in the presence of compression noise needs further investigation. Further, the classification results do not provide knowledge on the techniques employed in various processing modules inside the camera. As shall be seen from our results later in the paper, by acquiring information about the CFA pattern and the interpolation algorithms used in a camera, our proposed forensic methodology can support more accurate identification for a large number of cameras.

### III. CAMERA CAPTURING MODEL AND PROBLEM FORMULATION

In this section, we discuss the image capturing model and present our problem formulation. As illustrated by the image capturing model in Fig. 1(a), the light from a scene pass through a lens and optical filters, and is finally recorded by an array of charge coupled device (CCD) detectors. Few consumer-level color cameras directly acquire full-resolution information for all three primary colors (usually red, green, and blue)<sup>2</sup>. This is not only because of the high cost in producing a full-resolution sensor for all three colors, but also due to the substantial difficulty involved in perfectly matching the corresponding pixels and aligning the three color planes together. For these reasons, most digital cameras use a color filter array (CFA) to sample real-world scenes.

A color filter array consists of an array of color sensors, each of which captures the corresponding color of the real-world scene at an appropriate pixel location. Some examples of CFA patterns are shown in Fig. 1(b). The Bayer pattern (shown in left corner of Fig. 1(b)) is one of the most popular CFA patterns. It uses a rectangular lattice for the red and blue components of light and a diagonal lattice for the green color. The sensors are aligned on a square grid with the green color repeated twice compared to the corresponding red and blue sensors. The higher rate of sampling for the green color component enables to better capture the luminance component of light and thus provides better picture quality [23]. After CFA sampling, the remaining pixels are interpolated using the sampled data. Color interpolation (or demosaicking) is an important step to produce an output image with full resolution for all three color components [22] [24]. After interpolation, the three images corresponding to the red, green and the blue components go through a post-processing stage. In this stage, different types of operations such as white balancing, color correction, color matrixing, gamma correction, bit-depth reduction, and compression may be performed to enhance the overall picture quality and to reduce storage space.

To facilitate discussions, let  $S$  be the real-world scene to be captured by the camera and let  $p$  be the CFA pattern matrix.  $S(i, j, c)$  can be represented as a 3-D array of pixel values

of size  $H \times W \times C$ , where  $H$  and  $W$  represent the height and the width of the image, respectively, and  $C = 3$  denotes the number of color components (red, green, and blue). The CFA sampling converts the real-world scene  $S$  into a three dimensional matrix  $S_p$  of the form

$$S_p(i, j, c) = \begin{cases} S(i, j, c) & \text{if } p(i, j) = c, \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

After the data obtained from the CFA is recorded, the intermediate pixel values (corresponding to the points where  $S_p(i, j, c) = 0$  in Eqn. (1)) are interpolated using its neighboring pixel values to obtain  $S_p^{(I)}$ .

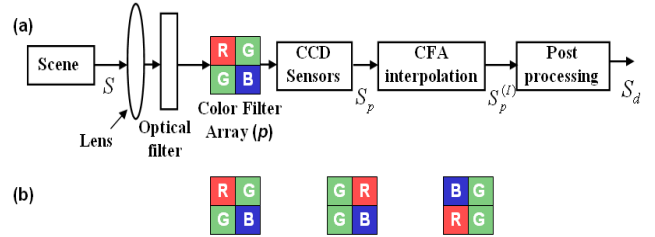


Fig. 1. (a) Image Capturing Model in Digital Cameras; (b) Sample Color Filter Arrays.

The performance of color interpolation directly affects the quality of the image captured by a camera [23]–[25]. There have been several algorithms for color interpolation. These algorithms can be broadly classified into two categories, namely, non-adaptive and adaptive algorithms. Non-adaptive algorithms apply the same principle for interpolation for all pixels in a group. Some typical examples of non-adaptive algorithms include the nearest neighbor, bilinear, bicubic, and smooth hue interpolations [24]. Traditionally, the bilinear and bicubic interpolation algorithms are popular due to their simplicity and ease in hardware implementation. However, these methods are known to produce visual artifacts in the final image, causing significant blurring along edge regions due to averaging across edges. More computationally intensive adaptive algorithms employing edge directed interpolation, such as the gradient based [26] and the adaptive color plane interpolation [27], have been proposed to reduce the blurring artifacts. The details of several popular interpolation methods are reviewed in Appendix I. The CFA interpolated image goes through post-processing stage to produce the final output image.

The problem of *component forensics* deals with a methodology and a systematic procedure to find the algorithms and parameters employed in the various components inside the device. In this work, we consider the problem of non-intrusive forensic analysis where we use sample images obtained from a digital camera under diverse and uncontrolled scene settings to determine the algorithms (and its parameters) employed in internal processing blocks. In particular, given an output image  $S_d$ , we focus on finding the color filter array pattern and the color interpolation algorithms, and show that the forensic analysis results of these components can be used as a first step in *reverse engineering* the making of a digital camera. In the

<sup>2</sup>New digital cameras employing Foveon X3 sensor, such as Sigma SD9 and Polaroid x530, capture all the three colors at each pixel location [39].

subsequent sections, we describe our proposed methodology and algorithms, and demonstrate them with detailed simulation results and case studies.

#### IV. FORENSIC ANALYSIS AND PARAMETER ESTIMATION OF CAMERA COMPONENTS

In this section, we develop a robust and non-intrusive algorithm to jointly estimate the CFA pattern and the interpolation coefficients by using only the output images from cameras. Our algorithm estimates the color interpolation coefficients in each local region through texture classification and linear approximation, and finds the CFA pattern that minimizes the interpolation errors [28].

More specifically, we establish a search space of CFA patterns based on common practice in digital camera design. We observe that most commercial cameras use a RGB type of CFA with a fixed periodicity of  $2 \times 2$  that can be represented as

$$\begin{array}{cc|c} C_1 & C_2 & \dots \\ C_3 & C_4 & \dots \\ \vdots & \vdots & \ddots \end{array}$$

where  $C_i \in \{R, G, B\}$  is the color of the corresponding sensor at a particular pixel location. In typical digital cameras, each of the three types of color sensors (R, G, and B) appears at least once in a  $2 \times 2$  cell, resulting in a total of 36 possible patterns in the search space, denoted by  $P$ . For every CFA pattern  $p$  in the search space  $P$ , we estimate the interpolation coefficients in different types of texture regions of the image by fitting linear filtering models. These coefficients are then used to re-estimate the output image  $\hat{S}_d^{(p)}$ , and find the interpolation error ( $\hat{S}_d^{(p)} - S_d$ ). We now present the details of the proposed algorithm.

##### A. Texture Classification and Linear Approximation

We approximate the color interpolation to be linear in chosen regions of the image [29]. More specifically, we divide the image into three kinds of regions based on the gradient features in a local neighborhood. Defining  $I_{i,j} = S_d(i, j, p(i, j))$ , the horizontal and vertical gradients at the location  $(i, j)$  can be found using

$$H_{i,j} = |I_{i,j-2} + I_{i,j+2} - 2I_{i,j}|, \quad (2)$$

$$V_{i,j} = |I_{i-2,j} + I_{i+2,j} - 2I_{i,j}|. \quad (3)$$

The image pixel at location  $(i, j)$  is classified into one of the three categories:

- *Region  $\mathcal{R}_1$*  contains those parts of the image with a significant horizontal gradient for which  $(H_{i,j} - V_{i,j}) > T$ , where  $T$  is a suitably chosen threshold;
- *Region  $\mathcal{R}_2$*  contains those parts of the image with a significant vertical gradient and is defined by the set of points for which  $(V_{i,j} - H_{i,j}) > T$ ; and
- *Region  $\mathcal{R}_3$*  includes the remaining parts of the image which are mostly smooth.

Using the final camera output  $S_d$  and the assumed sample pattern  $p$ , we identify the set of locations in each color of  $S_d$  that are acquired directly from the sensor array. We

approximate the remaining pixels to be interpolated with a set of linear equations in terms of the colors of the pixels captured directly. In this process, we obtain nine sets of linear equations corresponding to the three types of regions  $\mathcal{R}_m (m = 1, 2, 3)$  and three color channels (R, G, B) of the image.

Let the set of  $N_e$  equations with  $N_u$  unknowns for a particular region and color channel be represented as  $\mathbf{A}\mathbf{x} = \mathbf{b}$ , where  $\mathbf{A}$  of dimension  $N_e \times N_u$  and  $\mathbf{b}$  of dimension  $N_e \times 1$  specify the values of the pixels captured directly and those interpolated, respectively, and  $\mathbf{x}$  of dimension  $N_u \times 1$  stands for the interpolation coefficients to be estimated. To cope with possible noisy pixel values in  $\mathbf{A}$  and  $\mathbf{b}$  due to other in-camera operations following interpolation (such as JPEG compression), we employ singular value decomposition [30] to estimate the interpolation coefficients. Let  $\mathbf{A}_0$  and  $\mathbf{b}_0$  represent the ideal values of  $\mathbf{A}$  and  $\mathbf{b}$  in the absence of noise, and the errors in  $\mathbf{A}$  and  $\mathbf{b}$  be denoted by  $\mathbf{E}$  and  $\mathbf{r}$ , respectively, so that

$$\mathbf{A} = \mathbf{A}_0 - \mathbf{E}, \quad \mathbf{b} = \mathbf{b}_0 - \mathbf{r},$$

The values of  $\mathbf{x}$  are found by solving the minimization problem

$$\min_{\mathbf{E}, \mathbf{r}} \|\mathbf{E} \quad \mathbf{r}\|_F,$$

subject to the constraint that  $\mathbf{A}_0\mathbf{x} - \mathbf{b}_0 = 0$ , equivalently written as

$$[\mathbf{A} + \mathbf{E}, \quad \mathbf{b} + \mathbf{r}] \begin{bmatrix} \mathbf{x} \\ -1 \end{bmatrix} = 0. \quad (4)$$

Here  $\|\cdot\|_F$  denotes the *Forbenius* norm of the matrix so that

$$\|\mathbf{E} \quad \mathbf{r}\|_F = \left( \sum_{m=1}^{N_e} \sum_{n=1}^{N_u} |e(m, n)|^2 + \sum_{m=1}^{N_e} |r(m)|^2 \right)^{1/2}. \quad (5)$$

The solution to the minimization problem can be written as

$$\begin{bmatrix} \mathbf{x} \\ -1 \end{bmatrix} = -\frac{1}{v_{N_u+1, N_u+1}} \mathbf{v}_{N_u+1}, \quad (6)$$

where  $\mathbf{v}_{N_u+1}$  represents the  $(N_u + 1)^{th}$  right singular vector of the combined matrix  $[\mathbf{A} \quad \mathbf{b}]$ .

##### B. Finding the Interpolation Error and the CFA Sampling Pattern

Once we find the interpolation coefficients in each region, we use them to re-interpolate the sampled CFA output in the corresponding regions  $\mathcal{R}_m$ , to obtain an estimate of the final output image  $\hat{S}_d^{(p)}$ . Here, the superscript  $p$  is used to denote that the output estimate is based on the choice of the CFA pattern  $p$ . The difference between the estimated final output and the actual camera output image is the pixel-wise difference,  $e^{(p)} = \hat{S}_d^{(p)} - S_d$ . The interpolation error matrix  $e^{(p)}$  of dimension  $H \times W \times C$  is obtained for all candidate search patterns  $p \in P$ . Denoting the interpolation error in the red color component as  $e^{(p)}(., ., 1)$  and so on, the final error is computed by a weighted sum of the errors of the three color channels:

$$\begin{aligned} \varepsilon(p) = & w_R \|e^{(p)}(., ., 1)\|_F^2 + w_G \|e^{(p)}(., ., 2)\|_F^2 \\ & + w_B \|e^{(p)}(., ., 3)\|_F^2, \end{aligned} \quad (7)$$

The pattern  $\hat{p} = \arg \min_{p \in P} \varepsilon(p)$  that gives the lowest overall absolute value of the weighted error is chosen as the estimated pattern. The constants  $w_R, w_G$ , and  $w_B$  denote the corresponding weights used for the three color components (red, green, and blue), and their values are based on the relative significance of the magnitude of errors in the three colors. In our experiments, we choose  $w_R = w_B = 1$  and  $w_G = 2$  to give more importance to the error in the green channel as it provides more information about the luminance values of the pixel [23]. The interpolation coefficients corresponding to the estimated CFA pattern  $\hat{p}$  for all three types of regions and the three color channels are also obtained in this process. These coefficients can then be directly used to obtain the parameters of the components in the imaging model, as will be shown later in Section V-B. They can be processed to obtain further forensic evidence, and will be demonstrated by several case studies in Section VI.

### C. Reducing the Search Space for CFA Patterns

The search space for the CFA patterns can be reduced using a hierarchical approach. As an example, we synthetically generate a  $512 \times 512$  image, sample it on the Bayer pattern, and interpolate using the bicubic method. In Fig. 2, we show the detection statistics  $d_s(p)$  given by

$$d_s(p) = \frac{\varepsilon(p)}{H \times W \times (w_R + w_G + w_B)}, \quad (8)$$

and sorted in ascending order for the 36 different CFA patterns. In this case, the Bayer pattern gave the lowest interpolation error and was correctly identified. A closer look on the results in Fig. 2 reveals that the detection statistics form three separate clusters, with some values close to 0, some around 0.3–0.4, and others close to 0.7. Similar trend is also observed for real camera data and other synthetically generated images sampled on different CFA patterns and interpolated with the six representative interpolation techniques reviewed in Appendix I. This observation forms the basis for the proposed heuristic to reduce the search space of the CFA patterns.

Fig. 3 shows sample patterns from these three clusters. *Cluster 1* includes all  $2 \times 2$  patterns that have the same color along diagonal directions (either along the main diagonal or off-diagonal), chosen among the three colors (red, green, or blue). The remaining two spots can be filled in two different ways, giving a total of 12 such patterns in the first cluster. *Cluster 2* and *cluster 3* consists of patterns that have the same color along the horizontally (or vertically) adjacent blocks of the  $2 \times 2$  grid. *Cluster 2* has either red or blue color repeated to produce a total of 16 possible patterns. The remaining eight patterns with green appearing twice form *cluster 3*. In this example, the Bayer pattern is the actual color filter array and the patterns from first cluster give lower errors compared to the other clusters. The patterns from the third cluster gives the highest error values because the error in the green color channel is penalized more with the weight assignment  $w_G = 2$  and  $w_R = w_B = 1$  in Eqn. (7).

Such clustering of patterns into three groups helps us develop a heuristic to reduce the search space. We first divide

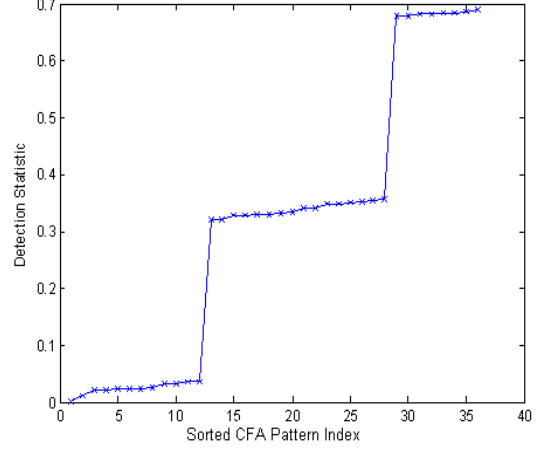


Fig. 2. Sorted detection statistics in terms of normalized overall error for different candidate search patterns

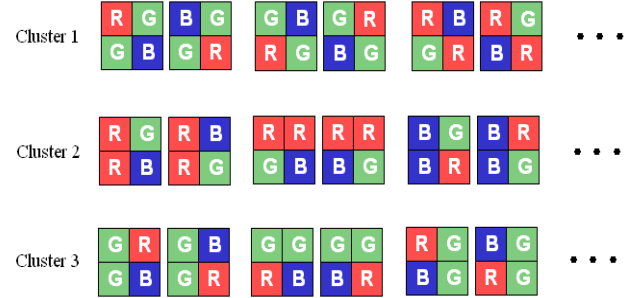


Fig. 3. Sample CFA patterns from the three clusters

the 36 patterns into three groups and choose one representative pattern from each of the three classes. The interpolation error is then estimated for these representative patterns to find the category that the actual CFA pattern is most likely to belong. Finally, a full search is performed on the chosen category to find the pattern with the lowest interpolation error. The number of searches required to find the optimal solution can be reduced to around 10. If additional information about the patterns are available, it may be used to further reduce the search space. For instance, a forensic analyst may choose to test only on those CFA patterns that have 2 green color components if he/she has such prior knowledge about the visual sensor.

### D. Evaluating Confidence in Component Parameter Estimation

In addition to identifying the parameters of the internal building blocks of the camera, it is also important to know the confidence level on the estimation result. A higher confidence value in estimation would increase the trustworthiness of the decision made by a forensic analyst.

We propose an entropy based metric to quantify the confidence level on the estimation result. Given a test image, we estimate its interpolation coefficients and provide it as an input to a  $c$ -class SVM classifier, trained on the coefficients of the  $c$  candidate interpolation methods. The probability that a

given test sample comes from the  $i^{th}$  class,  $q_i$ , is estimated from the soft decision values using the probabilistic SVM framework [31], and the test data point is classified into class  $k$  if  $q_k$  is larger than the other probabilities. Some details of the probabilistic SVMs are included in Appendix II for readers' reference. The confidence score  $\eta$  on the decision is then defined as

$$\eta = 2\Upsilon \left( 1 - \frac{\sum_{i=1}^c q_i \log_2 \left( \frac{1}{q_i} \right)}{\log_2 c} \right). \quad (9)$$

where  $\Upsilon(y) = z$  is defined as the inverse binary entropy function such that  $y = -z \log_2(z) - (1-z) \log_2(1-z)$  for  $0 \leq z \leq \frac{1}{2}$ . The argument to the  $\Upsilon$  function measures the entropy difference between the distribution  $\{q_i\}$  and a discrete uniform distribution, and the final value of  $\eta$  is normalized to the range of  $[0, 1]$  to represent a probability.

To verify that the proposed metric  $\eta$  can reflect the confidence level, we examine two extreme cases. When  $\mathbf{q} = [1, 0, 0, \dots, 0]$ , the decision of choosing the first class is made with a very high confidence and  $\eta = 1$ . And when  $\mathbf{q} = [\frac{1}{c} + \epsilon, \frac{1-\epsilon}{c}, \frac{1-\epsilon}{c}, \dots, \frac{1-\epsilon}{c}]$  where  $\epsilon$  is a small positive real number, there is an almost equal probability that the given data sample comes from any of the  $c$  classes. In this case, the decision is made with a very low confidence and  $\eta$  also approaches zero. For other values of  $\mathbf{q}$  between these two extreme cases, the value of  $\eta$  would lie in the interval  $[0, 1]$ , with a higher value indicating more confidence in the decision.

## V. EXPERIMENTAL RESULTS

### A. Simulation Results with Synthetic Data

We use synthetic data constructed from 20 representative images to study the performance of the proposed techniques. The original images are first downsampled to remove the effect of previously applied filtering and interpolation operations. They are then sampled on the three different CFA patterns as shown in Fig. 1(b). Each of the sampled images are interpolated using one of the six interpolation methods reviewed in Appendix I, namely, (1) Bilinear, (2) Bicubic, (3) Smooth Hue, (4) Median Filter, (5) Gradient based, and (6) Adaptive Color Plane. Thus, our total dataset contains  $20 \times 3 \times 6 = 360$  images, each of size  $512 \times 512$ .

1) *Simulation Results under no Post-processing:* We test the proposed CFA pattern and color interpolation identification algorithms on this synthetic data set. In the noiseless case with no post-processing, we observe no errors in estimating the CFA pattern. We use a  $7 \times 7$  neighborhood to estimate the interpolation coefficients for the three color components in the three types of texture regions, and pass it to a classifier to identify the interpolation algorithm. A support vector machine (SVM) [32] [33] classifier with a third-degree polynomial kernel is used to identify the interpolation method. We randomly choose 8 images (out of the 20) from each of the six interpolation techniques as ground truth for training and the remaining 12 images for testing. We repeat the experiment 500 times with a different set of images each time. The classifier is 100% accurate in identifying the correct color interpolation algorithm without any errors.

2) *Simulation Results with Post-processing:* As mentioned earlier, post-processing such as color correction and compression are commonly done in nearly all commercial cameras. Therefore, to derive useful forensic evidence from output images, it is very important that the proposed methods be robust to the common post-processing operations done in cameras.

Such post processing operations as color correction and white-balancing are typically multiplicative, where the final image is obtained by multiplying the color interpolated image by an appropriately chosen constant. The multiplicative factor depends on the camera type, lighting conditions, and chosen camera software options. These would not affect our solution to the linear equation sets since the same factor would be applied to both sides of the equations. In this work, we primarily focus on JPEG compression and additive noise, and study the performance under these distortions. Other post-processing operations such as gamma correction can be estimated from the final output images [12] and can be undone before computing the interpolation coefficients. For the results presented in this sub-section, we directly obtain the coefficients from the output images and do not perform inverse gamma correction based on the estimated values of gamma. Later in Section V-B, we show that the estimation results are robust to gamma correction distortions.

(i) **Performance Results Under JPEG compression:** JPEG compression is an important post-processing operation that is done in cameras. The noise introduced by compression could potentially result in errors in estimating the color interpolation coefficients and the CFA pattern. We test the proposed CFA pattern identification algorithm with the synthetic data obtained under different JPEG quality factors  $\{20, 30, \dots, 80, 90, 99\}$ . We observe that in all cases, the estimator gives very good results and the correct CFA pattern is always identified.

Next, we study the accuracy in identifying the color interpolation when the synthetically generated images are JPEG compressed. Here, we consider two possible scenarios. In the first case, a forensic analyst does not have access to the camera(s) and therefore does not have control over the input(s) to the device. He/she makes a judgement based on the forensic evidence obtained from the images submitted for trial. In this scenario, the pictures obtained with different interpolation methods would correspond to different scenes, which we shall call as the *multiple-scene* case. The performance of the proposed color interpolation identification methodology for the multiple-scene case at different JPEG quality factors is shown in Fig. 4(a). Here we use a total of 12 images (two images for each of the six interpolation methods) for training, and test with the remaining 8 images under each interpolation ( $8 \times 6 = 48$  in total). The experiment is repeated 500 times by choosing a random training set each time. We observe that the average percentage of images for which the interpolation technique is correctly identified is around 95 – 100% for moderate to high JPEG quality factors (80 – 100) and the average performance reduces to 80 – 85% for quality factors from 50 – 80.

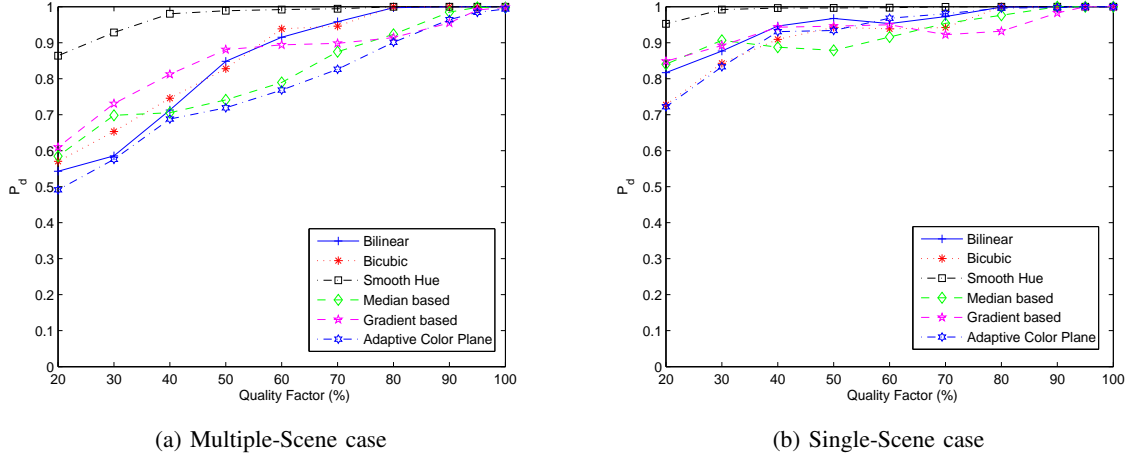


Fig. 4. Fraction of images for which the color interpolation technique is correctly identified under different JPEG compression quality factors. Testing results for synthetic data.

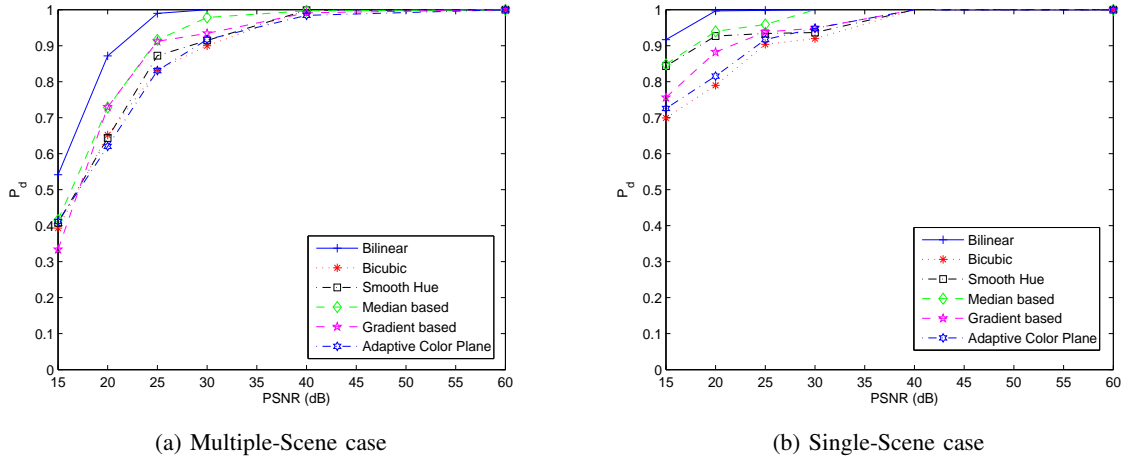


Fig. 5. Fraction of images for which the color interpolation technique is correctly identified under different noise PSNR's. Testing results for synthetic data.

Alternatively, if a forensic analyst has access to the camera, he/she can do controlled testing by choosing the input to the cameras so as to reduce the impact of the input's variation on the forensic analysis. In this scenario, the analyst may consider taking similar images with all the cameras under study, in order to improve the estimation accuracy, and to increase the confidence level on his/her final judgement. We call this situation the *single-scene* case. The single-scene case corresponds to the *semi non-intrusive* forensic analysis discussed earlier in Section I. The performance of the proposed color interpolation technique for this case for different JPEG quality factors is shown in Fig. 4(b). Here we use 8 images under the six interpolation techniques for training (48 in total) and the 72 remaining images for testing. We observe that for most JPEG quality factors, the average percentage of images for which the color interpolation technique is correctly identified is around 96% and thus the forensic decision can be made with a higher confidence compared to the multiple-scene case. The accuracy can be further improved using larger percentage of images while training. This suggests that with an increasing number of well-designed inputs to the system,

the detection performance can be much enhanced.

**(ii) Performance Results Under Additive Noise:** Additive noise can be used to model other kinds of random post-processing operations that may occur during the scene capture process. In order to study the noise resilience of a forensics system, we test the proposed CFA pattern identification algorithm with the images obtained under different noise levels with peak-signal to noise ratios (PSNR) of 15, 20, 30, and 40 dB, respectively. In almost all cases, we correctly identified the actual CFA pattern. We obtained only one error in estimation, when the PSNR was extremely low at 15dB and the adaptive color plane interpolation method was used. However, in this case, the correct pattern came in the top three choices.

We then study the identification performance of the color interpolation method under additive noise. The performance for synthetic data, averaged over 500 iterations, for the multiple-scene and the single-scene case are shown in Fig. 5(a) and Fig. 5(b), respectively. We observe that there is around 90% accuracy for the multiple-scene case and it increases to around 95% for the single-scene scenario.

TABLE I  
CAMERA MODELS USED IN SIMULATION, ORDER BY CAMERA MAKE

No.	Camera Model	No.	Camera Model
1	Canon Powershot A75	11	Olympus C3100Z/C3020Z
2	Canon Powershot S400	12	Olympus C765UZ
3	Canon Powershot S410	13	Minolta DiMAGE S304
4	Canon Powershot S1 IS	14	Minolta DiMAGE F100
5	Canon Powershot G6	15	Casio QV 2000UX
6	Canon EOS Digital Rebel	16	FujiFilm Finepix S3000
7	Nikon E4300	17	FujiFilm Finepix A500
8	Nikon E5400	18	Kodak CX6330
9	Sony Cyber-shot	19	Epson PhotoPC 650
10	Sony DSC P72		



Fig. 6. Super CCD sensor pattern

### B. Results on Camera Data

For our study, we consider 19 different cameras as shown in Table I. Around 40 different images are taken from each camera. To make it closer to the real-world scenario, the images are captured under completely random conditions—different sceneries, different lighting situations, and compressed under JPEG quality factors as specified by default values used in each camera. The default camera settings (prescribed image size, color correction, auto white balancing, JPEG compression, etc) are used in data acquisition. From each of these camera images, we randomly choose five different non-overlapping  $512 \times 512$  blocks per image and use it for subsequent analysis. Thus, our database consists of a total of 3800 different  $512 \times 512$  pictures with 200 samples for each of the 19 cameras.

Note that all the cameras in our database use RGB type of CFA pattern with red, green, and blue sensors. The search space for CFA in our experiments focusses on such CFA type, since it has been widely employed in digital camera design and most cameras currently in the market use a variation of this pattern. There are a few exceptions in CFA designs, for example, some models use CMYG type of CFA that captures the cyan, magenta, yellow, and green components of light [24]. The proposed algorithms can be extended to identify CMYG type CFA patterns by including appropriate patterns in the search space.

Among RGB type CFA patterns, several layouts of the three types of color filters have been used in practice. The  $2 \times 2$  square arrangement is the most popular and most digital cameras utilize a shifted variation of the Bayer pattern to capture the real world scene. Recently introduced super CCD cameras [34] have sensors placed as shown in Fig. 6. To test the performance of the proposed algorithms to such cameras, we include images from Fujifilm Finepix A500 (camera no. 17) that uses super CCD [34] in our database.

As an initial step, we try to estimate the CFA pattern from the output images using the algorithm described in Section IV. We observe with a high confidence that all the cameras except

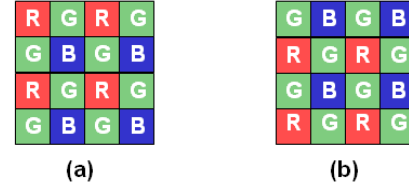


Fig. 7. Sample CFA patterns for (a) Canon EOS Digital Rebel, and (b) Fujifilm Finepix S3000.

0	0.007	0	0.027	0	0.014	0	0	0.006	0	0.009	0	0.009	0	0.007	0
0.008	0	-0.062	0	-0.070	0	0.003	0	0.005	0	-0.070	0	-0.069	0	0.007	0
0	-0.041	0	0.435	0	-0.032	0	0	0	-0.023	0	0.213	0	-0.019	0	0
-0.004	0	0.218	1.000	0.204	0	0.001	0	0.019	0	0.414	1.000	0.399	0	0.029	0
0	-0.034	0	0.441	0	-0.038	0	0	0	-0.014	0	0.214	0	-0.020	0	0
0.005	0	-0.075	0	-0.064	0	0.002	0	0.002	0	-0.074	0	-0.064	0	0.006	0
0	0.013	0	0.030	0	0.012	0	0	0	0.008	0	0.011	0	0.005	0	0
(a)															
0	-0.005	0	0.008	0	0.001	0	0	0	0.000	0	0.004	0	0.000	0	0
0.018	0	-0.079	0	-0.050	0	0.006	0	0	0	-0.035	0	-0.035	0	0	0
0	-0.021	0	0.315	0	-0.023	0	0	0	-0.035	0	0.316	0	-0.035	0	0
0.003	0	0.325	1.000	0.301	0	0.023	0	0.004	0	0.316	1.000	0.316	0	0.004	0
0	-0.023	0	0.301	0	-0.015	0	0	0	-0.035	0	0.316	0	-0.035	0	0
-0.006	0	-0.061	0	-0.051	0	0.010	0	0	0	-0.035	0	-0.035	0	0	0
0	0.002	0	0.018	0	-0.001	0	0	0	0.000	0	0.004	0	0.000	0	0
(c)															
(d)															

Fig. 8. Interpolation coefficients for the green channel for one sample image taken with the Canon Powershot A75 camera for (a) Region  $\mathcal{R}_1$  with significant horizontal gradient, (b) Region  $\mathcal{R}_2$  with significant vertical gradient, (c) Smooth region  $\mathcal{R}_3$ , (d) Coefficients of bicubic interpolation.

Fujifilm Finepix A500 (camera no. 17) use shifted versions of the Bayer color filter array as their CFA pattern. For instance, we apply our CFA estimation algorithm on JPEG images from Canon EOS Digital Rebel (camera no. 6) and the Fujifilm Finepix S3000 (camera no. 16); and the estimated  $2 \times 2$  CFA that minimized the fitting errors is shown in Fig. 7(a) and (b), respectively. The estimation results perfectly match these cameras' ground-truth data obtained by reading the headers of the raw images produced by the two cameras.

When testing the images from Fujifilm Finepix A500 (camera no. 17) with the same 36 square patterns in the CFA pattern search space, we notice that the best  $2 \times 2$  pattern in the search space is still a shifted version of the Bayer pattern. However, in this case, we observe that the minimum value of the error  $\varepsilon$  (as given by Eqn. (7)) is larger than the ones obtained from other square CFA cameras. Therefore, the overall confidence in decision making is lower for this super CCD camera compared to the other cameras in the database. Further, we also find that the CFA pattern estimation results are not consistent across different images taken with the same camera, i.e., different images from Fujifilm Finepix A500 give different shifted versions of the Bayer pattern as the estimated CFA. Such inconsistencies in the results along with lower confidence in parameter estimation could provide forensic evidence to indicate that the camera does not employ a square CFA pattern. One possible approach to identify super CCD is to enlarge the CFA search space to include these patterns. We plan to further investigate this aspect in our future work to gather forensic evidence to distinguish super CCD cameras and square CFA cameras.

Next, we try to estimate the color interpolation coefficients in different image regions using the algorithm presented in Section IV-B. In our simulations, we find the coefficients of

a  $7 \times 7$  filter in each type of region, and color channel, thus giving a total of  $7 \times 7 \times 3 \times 3 = 441$  coefficients per image. Sample coefficients obtained using the Canon Powershot A75 camera for the three types of regions in the green image are shown in Fig. 8. For region  $\mathcal{R}_1$  that corresponds to areas having significant horizontal gradient, we observe that the value of the coefficients in the vertical direction are significantly higher than those in the horizontal directions. This indicates that the interpolation is done along the edge, which in this case is oriented along the horizontal direction. Similar corresponding inferences can be made from coefficients in region  $\mathcal{R}_2$  of significant vertical gradient. Compared to these two regions, the coefficients in region  $\mathcal{R}_3$  have almost equal values in all four directions, and do not have any directional properties. Moreover, careful observation of the coefficients in region  $\mathcal{R}_3$  reveals their close resemblance to the bicubic interpolation coefficients shown in Fig. 8(d). This suggests that it is very much likely that the Canon Powershot A75 camera uses bicubic interpolation for smooth regions of the image. Similar results obtained for other camera models indicate with  $\eta = 96\%$  confidence that all cameras use the bicubic interpolation for handling smooth regions. This is consistent with common knowledge in image processing practice that bicubic interpolation is good for regions with slowly changing intensity values. [35].

## VI. CASE STUDIES AND APPLICATIONS OF NON-INTRUSIVE FORENSIC ANALYSIS

In this section, we present case studies to illustrate the applications of the proposed non-intrusive forensic analysis methodology for camera identification (acquisition forensics), and for providing clues to identify infringement/licensing.

### A. Identifying Camera Brand from Output Images

The color interpolation coefficients estimated from the image can be used as features to identify the camera brand utilized to capture the digital image. As shown in Section V-B, most cameras employ similar kinds of interpolation techniques for smooth regions. Therefore, we use the coefficients obtained from the significant horizontal edge regions  $\mathcal{R}_1$  and vertical edge regions  $\mathcal{R}_2$  as features to construct a camera brand identifier.

To obtain more reliable forensic evidence from the input image for camera identification, we first pre-process the image by edge detection, to locate five significant  $512 \times 512$  blocks with the highest absolute sum of gradient values. The interpolation coefficients corresponding to the regions  $\mathcal{R}_1$  and  $\mathcal{R}_2$ , from all three color channels, estimated from these  $512 \times 512$  blocks are used as features for identification.

We use a classification based framework to identify the camera brand. For each camera in the database, we collect 40 different images and obtain 200 different  $512 \times 512$  image blocks by locating the top five regions with higher gradient values. These 200 image blocks collected from each of the 19 cameras are grouped so that all images from the same brand form one class. An 9-camera brand SVM classifier

with a polynomial kernel function [33] is constructed with a randomly chosen 50% of the images from each class for training. The remaining images are used in testing and the process is repeated 500 times by randomly choosing a training set each time. Table II shows the average confusion matrix, where the  $(i, j)^{th}$  element gives the percentage of images from camera brand- $i$  that are classified to belong to camera brand- $j$ . The main diagonal elements represent the classification accuracy and achieve a high average value of 90% for nine camera brands.

The above results have demonstrated the effectiveness of the color interpolation component as features to differentiate different camera brands. The robustness of estimating these features under JPEG and additive noise has been shown earlier in Section V-A.2. Here we further examine the robustness against such nonlinear point operations as gamma correction. As a common practice in digital camera design, most cameras perform gamma correction with a  $\gamma = 1/2.2$  to match the luminance of the digital image with that of the display monitor. In order to test the goodness of the proposed algorithms for gamma correction, we first do inverse gamma correction with  $\gamma = 2.2$  on the original camera images<sup>3</sup>. The interpolation coefficients are then estimated from these gamma corrected images and used in camera brand identification. In this case, the confusion matrices are similar to the ones in Table II, and average identification accuracy was estimated to be 89%. This suggests that the camera identification results are invariant to gamma correction in digital cameras.

Some exploration on the problem of camera brand identification were made recently in [20] [21]. In these works, the authors test their algorithms with pictures taken under controlled conditions with the same scene captured with multiple cameras (corresponding to the *single-scene* case discussed earlier in Section V-A) and the best performance reported is 84% on only three brands [21]. Further, the performance of the null-based features employed in [21] is considerably dependent on the image content, and sensitive to other in-camera processing such as compression. Compared to these prior approaches, the interpolation coefficients derived in our work by exploring the spatial filtering relations are less dependent on input scenes and are robust against various common in-camera processing. The formulation of minimizing noise norm helps resist noise and other in-camera processing. As a result, the features coming from the proposed component forensics methodologies are able to achieve a higher classification accuracy over a much larger database with 19 camera models from nine different brands. Further, the proposed component forensic techniques are more generic with the goal of identifying the algorithms and parameters employed in various components in digital cameras, and are not restricted to camera brand identification.

<sup>3</sup>In a general scenario, the value of  $\gamma$  can be estimated from the output images and the corresponding inverse could be applied before estimating the interpolation coefficients.

TABLE II  
CONFUSION MATRIX FOR IDENTIFYING DIFFERENT CAMERA BRANDS (\* DENOTES VALUES SMALLER THAN 4%)

	Canon	Nikon	Sony	Olympus	Minolta	Casio	Fuji	Kodak	Epson
Canon	96%	*	*	*	*	*	*	*	*
Nikon	*	83%	5%	*	*	*	*	*	*
Sony	*	*	90%	*	*	*	*	*	*
Olympus	*	*	*	93%	*	*	*	*	*
Minolta	8%	*	*	*	81%	*	*	*	*
Casio	*	*	*	6%	*	89%	*	*	*
Fuji	*	*	*	*	7%	*	87%	*	*
Kodak	*	*	*	*	*	*	*	89%	*
Epson	*	*	*	*	*	*	*	*	100%

### B. Identifying Camera Model from Output Images

Our results in the previous subsection demonstrate the robustness in non-intrusively identifying the camera brand using the color interpolation coefficients as features. In this subsection, we extend our studies to answer further forensic questions to find the exact camera model used to capture the given digital image, and examine the performance in identifying the camera model.

We use 200 images from each of the 19 cameras in our experiments. A randomly chosen 125 images are used for training and the remaining are for testing with a 19-camera model SVM classifier. The results of the simulation are shown in Table III in terms of the average confusion matrix. The  $(i, j)^{th}$  element in the confusion matrix gives the fraction of images from camera model  $i$  classified as camera model  $j$ . In order to highlight the significant values of the table, we show only those set of values that are greater than or equal to a chosen threshold  $\lambda = 1/N_c$ , where  $N_c$  is the number of cameras ( $\lambda = 1/19$  in our experiments). The average classification accuracy is 86% for 19 camera models.

The classification results reveal some similarity among different camera models in handling interpolation, as there are some off-diagonal elements that have a non-zero value greater than the threshold of  $1/19$ . For example, among the Canon Powershot S410 (camera no. 3) images, 20% were classified as belonging to Canon Powershot S400 (camera no. 2). A similar trend is also observed for images from Canon Powershot G6 (camera no. 5) and Canon EOS Digital REBEL (camera no. 6). These results indicate that the color interpolation coefficients are quite similar among the Canon models and hence it is likely that they are using similar kinds of interpolation methods.

### C. Similarities in Camera Color Interpolation Algorithms

Motivated by the results in the previous subsection, we further analyze the similarity between the camera models in this subsection, and propose metrics to quantitatively evaluate the closeness among interpolation coefficients from several cameras.

1) *Studying Similarities in Cameras using Leave-One-Out:* We perform additional experiments to identify the camera models with similar color interpolation by a *leave-one-out* procedure. More specifically, we train the classifier by omitting the data from one of the camera models and test it with these coefficients, to find the nearest neighbor in the color

interpolation coefficient space. For instance, when we train the SVM using all the 200 images from 18 cameras except Canon Powershot S410 (Camera no. 3), and then test it using the 200 images from Canon Powershot S410, we observe that 66% of the Canon Powershot S410 images are classified as Canon Powershot S400. Furthermore, out of the remaining images, 28% of the pictures are classified among one of the remaining Canon models. The reverse trend is also observed when we train with all the images except Canon Powershot S400 (camera no. 2) and use these images for testing. In this case, we notice that around 45% of the Canon Powershot S400 pictures are classified as Canon Powershot S410, 19% are categorized as Canon Powershot A75, and 15% of the remaining guessed as some other Canon model. This result suggests that there is a considerable amount of similarity in the kind of interpolation algorithms used by various Canon models.

Similar trend is also observed for the two Sony cameras in our database. We note that around 66% of the Sony Cybershot model are classified as Sony DSC P72 model when the former was not used in training. These results indicate the similarities in the kind of interpolation algorithm among various models of the same brand. Interestingly, we also observe similarity between Minolta DiMage S304 and Nikon E4300. Around 53% of the Minolta DiMage S304 pictures are designated as Nikon E4300 camera model. This indicates a closeness between the interpolation coefficients in the feature space.

2) *Quantifying Similarity in Color Interpolation with a Divergence Score:* From our preliminary analysis in V-B, we observe that the majority of the cameras use similar kinds of interpolation techniques in handling smooth regions. As a next step, we perform studies to investigate and obtain forensic evidence on the type of interpolation used by a camera in the non-smooth regions.

We extend our interpolation coefficient estimation model in Section IV-B to explicitly target at non-smooth regions in the image. To do so, we divide the image into *eight* regions depending on the relative gradient estimates in eight directions (namely north, east, west, south, north-east, north-west, south-east and south-west). The gradient values in this case are obtained as in the threshold-based variable number of gradients (VNG) algorithm [36]. In this case, the gradient in the north

TABLE III

CONFUSION MATRIX FOR ALL CAMERAS. THE MATRIX IS DIVIDED BASED ON DIFFERENT CAMERA MAKES. THE VALUES BELOW THE THRESHOLD  $\lambda = \frac{1}{19}$  ARE DENOTED BY \*. THE CAMERA INDEX NUMBERS ARE ACCORDING TO TABLE I.

		01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
Canon	01	0.92	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
	02	*	0.84	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
	03	*	0.20	0.80	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
	04	*	0.16	*	0.56	*	*	*	*	0.16	*	*	*	*	0.08	*	*	*	*	*
	05	*	*	0.08	*	0.88	*	*	*	*	*	*	*	*	*	*	*	*	*	*
	06	*	*	*	0.08	*	0.88	*	*	*	*	*	*	*	*	*	*	*	*	*
Nikon	07	*	*	*	*	0.16	*	0.60	*	*	0.08	*	*	*	0.08	*	*	*	*	*
	08	*	*	*	*	*	*	*	0.72	*	*	*	*	*	0.08	*	*	0.16	*	*
Sony	09	*	*	*	*	*	*	*	*	1.00	*	*	*	*	*	*	*	*	*	*
	10	*	*	*	*	*	*	*	*	*	1.00	*	*	*	*	*	*	*	*	*
Olympus	11	*	*	*	*	*	*	*	*	*	*	1.00	*	*	*	*	*	*	*	*
	12	*	*	*	*	*	*	*	*	*	*	*	1.00	*	*	*	*	*	*	*
Minolta	13	*	*	*	*	*	*	*	*	*	*	*	*	0.96	*	*	*	*	*	*
	14	0.08	*	*	*	*	*	*	*	*	0.08	*	*	0.80	*	*	*	*	*	*
Casio	15	*	0.16	*	*	*	*	*	*	*	*	*	*	*	0.84	*	*	*	*	*
	16	*	*	*	*	*	*	*	*	*	*	*	*	*	*	0.96	*	*	*	*
FujiFilm	17	*	*	0.08	*	*	*	*	*	*	*	*	*	0.20	*	*	0.68	*	*	*
	18	*	*	*	*	*	*	*	*	*	*	*	*	*	0.08	*	*	0.88	*	*
Epson	19	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	1.00	*

TABLE IV

DIVERGENCE SCORES FOR DIFFERENT CAMERA MODELS. THE VALUES BELOW 0.04 ARE SHADED, AND THE \* INDICATES ZERO SIMILARITIES BETWEEN THE SAME CAMERA MODELS BY DEFINITION. THE CAMERA INDEX NUMBERS ARE ACCORDING TO TABLE I.

		01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
Canon	01	*	0.06	0.06	0.17	0.14	0.35	0.36	0.68	0.22	0.76	0.25	0.31	0.18	0.09	0.31	0.18	0.54	0.83	0.25
	02	0.06	*	0.05	0.07	0.05	0.19	0.19	0.46	0.11	0.55	0.12	0.14	0.11	0.03	0.15	0.17	0.35	0.57	0.16
	03	0.06	0.05	*	0.15	0.06	0.18	0.22	0.47	0.22	0.51	0.23	0.27	0.20	0.10	0.29	0.33	0.31	0.57	0.27
	04	0.17	0.07	0.15	*	0.10	0.22	0.23	0.50	0.14	0.71	0.04	0.08	0.10	0.07	0.10	0.19	0.49	0.58	0.23
	05	0.14	0.05	0.06	0.10	*	0.07	0.14	0.36	0.14	0.46	0.19	0.14	0.16	0.09	0.16	0.32	0.25	0.39	0.26
	06	0.35	0.19	0.18	0.22	0.07	*	0.15	0.36	0.18	0.42	0.32	0.21	0.30	0.22	0.23	0.54	0.23	0.35	0.37
Nikon	07	0.36	0.19	0.22	0.23	0.14	0.15	*	0.21	0.19	0.21	0.24	0.16	0.12	0.18	0.17	0.47	0.12	0.19	0.34
	08	0.68	0.46	0.47	0.50	0.36	0.36	0.21	*	0.53	0.16	0.47	0.39	0.39	0.48	0.41	0.89	0.31	0.10	0.92
Sony	09	0.22	0.11	0.22	0.14	0.14	0.18	0.19	0.53	*	0.09	0.17	0.08	0.11	0.07	0.07	0.13	0.43	0.61	0.14
	10	0.76	0.55	0.51	0.71	0.46	0.42	0.21	0.16	0.09	*	0.66	0.61	0.52	0.56	0.59	1.02	0.18	0.23	0.82
Olympus	11	0.25	0.12	0.23	0.04	0.19	0.32	0.24	0.47	0.17	0.66	*	0.11	0.10	0.08	0.11	0.19	0.56	0.61	0.25
	12	0.31	0.14	0.27	0.08	0.14	0.21	0.16	0.39	0.08	0.61	0.11	*	0.08	0.12	0.01	0.20	0.42	0.62	0.26
Minolta	13	0.18	0.11	0.20	0.10	0.16	0.30	0.12	0.39	0.11	0.52	0.10	0.08	*	0.06	0.08	0.17	0.39	0.45	0.25
	14	0.09	0.03	0.10	0.07	0.09	0.22	0.18	0.48	0.07	0.56	0.08	0.12	0.06	*	0.11	0.11	0.42	0.61	0.13
Casio	15	0.31	0.15	0.29	0.10	0.16	0.23	0.17	0.41	0.07	0.59	0.11	0.01	0.08	0.11	*	0.18	0.44	0.45	0.24
	16	0.18	0.17	0.33	0.19	0.32	0.54	0.47	0.89	0.13	1.02	0.19	0.20	0.17	0.11	0.18	*	0.82	1.05	0.17
FujiFilm	17	0.54	0.35	0.31	0.49	0.25	0.23	0.12	0.31	0.43	0.18	0.56	0.42	0.39	0.42	0.44	0.82	*	0.23	0.51
	18	0.83	0.57	0.57	0.58	0.39	0.35	0.19	0.10	0.61	0.23	0.61	0.42	0.45	0.61	0.45	1.05	0.23	*	0.98
Epson	19	0.25	0.16	0.27	0.23	0.26	0.37	0.34	0.92	0.14	0.82	0.25	0.26	0.25	0.13	0.24	0.17	0.51	0.98	*

direction  $J_N$  is obtained using

$$\begin{aligned}
J_N(x, y) = & |Y(x-1, y) - Y(x+1, y)| \\
& + |Y(x-2, y) - Y(x, y)| \\
& + 0.5 \times |Y(x-1, y-1) - Y(x+1, y-1)| \\
& + 0.5 \times |Y(x-1, y+1) - Y(x+1, y+1)| \\
& + 0.5 \times |Y(x-2, y-1) - Y(x, y-1)| \\
& + 0.5 \times |Y(x-2, y+1) - Y(x, y+1)|. \quad (10)
\end{aligned}$$

where the image  $Y$  is defined as in Eqn. (20) in Appendix I. Similar expressions for gradients in the remaining seven directions are used to find the local gradient values [36]. Once these gradients are obtained, they are compared to a threshold to divide the image into eight texture regions. The interpolation coefficients are obtained in each region by solving a set of linear equations as given by Eqn. (6).

We use a classification based methodology to study the similarities in interpolation algorithms used by different cameras. To construct classifiers, we start with 100 representative

images, downsample them (by a factor of 2) and then re-interpolate with each of the six different interpolation methods as discussed in Section V-A. With a total of 600 images synthetically generated in this way, we run the color interpolation estimator to find the coefficients for each image. The estimated coefficients are then used to train a 6-class SVM classifier, where each class represents one interpolation method. After training the SVM classifier, we use it to test the images taken by the 19 cameras. For each of the 2000 images taken by every camera in the 19-camera dataset, we estimate the CFA parameters (eight sets of coefficients each with a dimension of  $5 \times 5$ ), feed them as input to the above classifier and record the classification results. Probabilistic SVM framework is used in classification and the soft decision values are recorded for each image [31] (refer to Appendix II for more details). If the two camera models employ different interpolation methods (not necessarily the same as the six typical methods in the classifier), then the classification results are likely to be quite different, and their differences can be

quantified by an appropriate distance between the classification results.

More specifically, for each image in the database, the interpolation coefficients are found and fed into the  $N$ -class classifier, where  $N$  denotes the number of possible choices of the interpolation algorithms studied ( $N = 6$  in our experiments). Let the output of the classifier be denoted as a probability vector  $\underline{g} = [g_1, g_2, \dots, g_N]$ , where  $g_i$  gives the probability that the input image employs the interpolation algorithm- $k$ . Similar probability vectors are obtained for every image in the database and the average performance is computed for each camera model. Let the average classification results for camera model- $i$  be represented by the vector  $\underline{\pi}_i = [\pi_{i1}, \pi_{i2}, \dots, \pi_{iN}]$ , where  $\pi_{ik}$  is the average probability for an image from camera model- $i$  to be classified as using the interpolation algorithm- $k$ . The  $\pi_{ik}$ 's are estimated using soft decision values obtained using the probabilistic SVM framework. The similarities of the interpolation algorithms used by any two cameras (with indices  $i$  and  $j$ ) can now be measured in terms of a *divergence* score- $\varphi_{ij}$  defined as symmetric *Kullback-Leibler* (KL) distance between the two probability distributions  $\underline{\pi}_i$  and  $\underline{\pi}_j$ :

$$\varphi_{ij} = D(\underline{\pi}_i || \underline{\pi}_j) + D(\underline{\pi}_j || \underline{\pi}_i), \quad (11)$$

$$\text{where } D(\underline{\pi}_i || \underline{\pi}_j) = \sum_{k=1}^N \pi_{ik} \log_2 \left( \frac{\pi_{ik}}{\pi_{jk}} \right). \quad (12)$$

The symmetric KL distance is obtained in each of the eight types of regions separately by training with synthetic data and testing with the camera images using the appropriately chosen coefficients as features. The overall divergence score is obtained by taking the mean of the individual divergence scores in eight regions and three color components. A low value of overall divergence score indicates that the two cameras are similar and are likely to use very similar kind of interpolation methods.

The divergence scores of the 19 different camera models are shown in Table IV. Here, the  $(i, j)^{th}$  element in the matrix represents the average symmetric KL distance between the interpolation coefficients of camera model- $i$  and camera model- $j$ . Divergence scores below a threshold of 0.06 have been highlighted for reference. We observe from the table that most cameras from the same brand seem to use similar kinds of interpolation algorithms. This is especially evident for the some models of Canon and Minolta used in our analysis.

The divergence score between the two Canon models, S400 and S410, are very low, suggesting that both of these models are likely to use similar techniques for color interpolation. We also observe similarities between the two Minolta models, DiMage S301 and DiMage F100, and between the two Sony models, Cybershot and DSC P72. The metric is very close to zero in all these cases, thus indicating that cameras from the same manufacturer have similar interpolation. Interestingly, we also observe some similarity between several cameras from different manufactures. As shown in Table IV, the divergence score between Nikon model E4300 (camera no. 7) and the Minolta DiMage S304 (camera no. 13) is low, which suggests

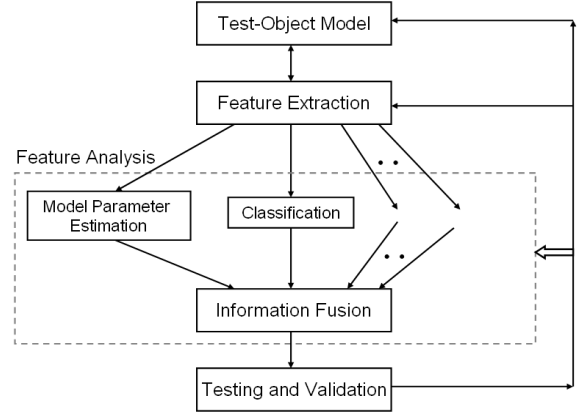


Fig. 9. The Proposed Forensic Analysis Methodology

a resemblance in the type of interpolation used by these two cameras.

The work that we have presented so far quantifies the similarity of camera models based on the estimated color interpolation coefficients. The parameters of the other stages in the scene capture model, such as white balancing and JPEG compression, may be further used to study similarities among different camera models and brands. In such cases, the forensic information collected from various components may also be fused together to provide quantitative evidence to identify and analyze technology infringement/licensing of cameras.

## VII. GENERAL COMPONENT FORENSICS METHODOLOGY

In this section, we extend the proposed non-intrusive forensic analysis to a methodology applicable to a broad range of devices. Let  $O_1, O_2, \dots, O_{N_o}$  be the sample outputs obtained from the test device that we model as a black box, and  $C_1, C_2, \dots, C_{N_c}$  be the individual components of the black box. Component forensics provides a set of methods to help identify the algorithm and parameters used by each of the processing blocks  $C_y$ . A general forensic analysis framework is composed of the following processing steps as shown in Fig. 9.

- 1) *Modelling of the Test Device*: As the first step of forensic analysis, a model is constructed for the object under study. This modeling helps in breaking down the test device into a set of individual processing components  $C_1, C_2, \dots, C_{N_c}$  and systematically studying the effect of each of these blocks on the final outputs obtained with the test object.
- 2) *Feature Extraction*: The forensic analyst identifies a set of features  $F_1, F_2, \dots, F_{N_f}$  that has a good potential to help identify the algorithms used in  $y^{th}$  device component  $C_y$ . These features are based on the final output data, and are chosen to uniquely represent each of the algorithms used. For the case of digital cameras, we have used in this paper the estimated color interpolation coefficients as features for further analysis. Parameters of other components, such as white balancing constants

and gamma correction values, can also be used as features.

- 3) *Feature Analysis and Information Fusion:* We analyze the features extraction from the previous stage to obtain forensic evidence to meet specific applications' needs. The appropriate analysis technique depends on the component under study, the application scenario, and the type of evidence desired. The results obtained from each of the analysis techniques can be combined to provide useful evidence about the inner working of the device components.
- 4) *Testing and Validation Process:* The validation stage uses test data with known ground truth to quantify the accuracy and performance of the forensic analysis system. It reflects the degree of success of each of the above processing stages and their combinations. Representative synthetic data obtained using the model of the test object can help provide ground truth to validate the forensic analysis systems and provide confidence levels on estimation. The results of this stage can also facilitate a further refinement of the other stages in the framework.

The methods and techniques adopted in each stage may vary depending on the device, the nature of the device components, and the application scenario. Regarding feature extraction, in some situations, the features by themselves (without further processing) can prove to be useful forensic evidence, and they can be used to estimate the parameters of the model. For instance, the color interpolation coefficients were directly estimated from the camera output, and used to study the nature of interpolation in different regions of the image in Section V-B. Evidence collected from such analysis can be used to study the similarities and differences in the techniques employed in the device components across several models and answer questions related to infringement/licensing and evolution of digital devices. In some other application scenarios, the component parameters might be an intermediate step and further processing would be required to answer specific forensic questions. For example, we have used the estimated color interpolation coefficients as features to build a robust camera identifier to determine the camera model (and make) that was used to capture a given digital image as seen in Sections VI-A and VI-B.

## VIII. CONCLUSIONS AND FUTURE WORK

In this paper, we consider the problem of component forensics and propose a set of forensic signal processing techniques to identify the algorithms and parameters employed in individual processing modules in digital cameras. The proposed methodology is non-intrusive and uses only the sample data obtained from the digital camera to find the camera's color array pattern and the color interpolation methods. We show through detailed simulations that the proposed algorithms are robust to various kinds of postprocessing that may occur in the camera. These techniques are then used to gather forensic evidence on real world data-sets captured with 19 camera models of nine different brands under diverse situations. The proposed forensic methodology is used to build a robust

camera classifier to non-intrusively find the camera brand and model employed to capture a given image for problems involving image source authentication. Our results indicate that we can efficiently identify the correct camera brand with an overall average accuracy of 90% for nine brands. Our analysis also suggests that there is a considerable degree of similarity within the cameras of the same brand (e.g. Canon models) and some level of resemblance among cameras from different manufacturers. Measures for similarity are defined and elaborate case-studies are presented to elucidate the similarities and differences among several digital cameras. We believe that such forensic evidence would be great source of information for technology infringement cases, intellectual property rights management, and for studying evolution for digital media.

In our future work, we plan to investigate other important components inside digital cameras, such as white balancing. For many cameras in the market that do not provide raw sensor output, the estimation of white balancing algorithm and parameters will facilitate non-intrusive estimations of the raw data acquired directly by the sensor prior to the corrective operations. Comparing the information about the raw sensor data and the white balanced results will provide valuable information on the distinct characteristics of the sensor. This will allow us to push the component forensic capability deeper into the core of the imaging device.

## APPENDIX I: SOME POPULAR COLOR INTERPOLATION ALGORITHMS

There have been numerous algorithms employed in practice for Color Filter Array interpolation. In this appendix, we briefly review some of the popular methods. For a detailed survey, the readers are referred to [24]. The color interpolation methods can be broadly classified into two main categories, namely, adaptive and non-adaptive methods, depending on their adaptability to the image content. While non-adaptive methods use the same pattern for all pixels in image, adaptive methods such as gradient based algorithms use the pixel values of the local neighborhood to find the best set of coefficients to minimize the overall interpolation error.

*Bilinear and Bicubic* methods are examples of non-adaptive interpolation schemes. In these algorithms, the pixel values are interpolated according to the following equations [22]:

$$R(x, y) = \sum_{u, v \in -N_g}^{N_g} h_r(u, v) \tilde{R}(x - u, y - v), \quad (13)$$

$$G(x, y) = \sum_{u, v \in -N_g}^{N_g} h_g(u, v) \tilde{G}(x - u, y - v), \quad (14)$$

$$B(x, y) = \sum_{u, v \in -N_g}^{N_g} h_b(u, v) \tilde{B}(x - u, y - v), \quad (15)$$

where  $\tilde{R}$ ,  $\tilde{G}$  and  $\tilde{B}$  are the original values obtained from the sensor and captured by the appropriate sensors;  $h_r$ ,  $h_g$ , and  $h_b$  denote the 2-D filters of dimension  $N_g \times N_g$  used in interpolation. For the bilinear case, these filters are given by

$$h_r = h_b = \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}, \text{ and } h_g = \frac{1}{4} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

The corresponding filters for the bicubic case are given by

$$h_r = h_b = \frac{1}{256} \begin{bmatrix} 1 & 0 & -9 & -16 & -9 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -9 & 0 & 81 & 144 & 81 & 0 & -9 \\ -16 & 0 & 144 & 256 & 144 & 0 & -16 \\ -9 & 0 & 81 & 144 & 81 & 0 & -9 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -9 & -16 & -9 & 0 & 1 \end{bmatrix}, h_g = \frac{1}{256} \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -9 & 0 & -9 & 0 & 0 \\ 0 & -9 & 0 & 81 & 0 & -9 & 0 \\ 1 & 0 & 81 & 256 & 81 & 0 & 1 \\ 0 & -9 & 0 & 81 & 0 & -9 & 0 \\ 0 & 0 & -9 & 0 & -9 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

The *Smooth Hue* interpolation algorithm is based on the observation that the hue varies smoothly in natural images. In this case, the green channel is interpolated using bilinear interpolation. The red (and blue) components are obtained by interpolating the ratios  $R/G$  (and  $B/G$ ) as given by

$$\frac{R(x, y)}{G(x, y)} = \frac{1}{2} \left( \frac{R(x, y-1)}{G(x, y-1)} + \frac{R(x, y+1)}{G(x, y+1)} \right). \quad (16)$$

In *Median filter* based algorithms, the three channels are first interpolated using bilinear interpolation. Then the differences  $R-G$ ,  $R-B$ , and  $G-B$ , are median filtered to produce  $M_{rg}$ ,  $M_{rb}$ , and  $M_{gb}$  respectively. At each pixel location, the missing color values are obtained by linearly combining the original color sensor value and the appropriate median filter result [22]. For example, the green color component at the location of the red color filter is obtained as

$$G(x, y) = R(x, y) - M_{rg}(x, y). \quad (17)$$

All the methods described above are non-adaptive in nature and does not depend on the characteristics of particular regions. In contrast to these techniques, the *Gradient Based* algorithms are more complex. Here, the horizontal gradient ( $J_h$ ) and the vertical gradient ( $J_v$ ) at the point  $(x, y)$  are first estimated using

$$J_h(x, y) = |Y(x, y-2) + Y(x, y+2) - 2Y(x, y)|, \quad (18)$$

$$J_v(x, y) = |Y(x-2, y) + Y(x+2, y) - 2Y(x, y)|, \quad (19)$$

where

$$Y(x, y) = \begin{cases} R(x, y) & \text{if } p(x, y) = 1, \\ G(x, y) & \text{if } p(x, y) = 2, \\ B(x, y) & \text{if } p(x, y) = 3, \end{cases} \quad (20)$$

and  $p$  is the CFA pattern matrix (e.g. Bayer pattern) with  $p(x, y) = 1$ , indicating that if the CFA pattern at the  $(x, y)^{th}$  pixel is red, green, or blue. The edge direction is then estimated from the gradient values, and the missing pixel values in the green component of the image are obtained in such a way that the interpolation is done along the edge and not across the edge, using only pixel values from the green channel. The missing red (and blue) components are found by interpolating the difference,  $R-G$  (or  $B-G$ ) along the edge.

*Adaptive Color Plane* interpolation method is an extension of the gradient based method. Here, the horizontal and vertical gradients are estimated using

$$J_h(x, y) = |Y(x, y-1) - Y(x, y+1)| + |Y(x, y-2) + Y(x, y+2) - 2Y(x, y)|, \quad (21)$$

$$J_v(x, y) = |Y(x-1, y) - Y(x+1, y)| + |Y(x-2, y) + Y(x+2, y) - 2Y(x, y)|. \quad (22)$$

In this case, unlike the gradient based method, the interpolation of one color component, also uses the other colors and the output is a linear combination of sampled sensor outputs in the neighborhood across the three color channels [27].

## APPENDIX II: PROBABILISTIC SUPPORT VECTOR MACHINES

We employ the probabilistic SVM framework proposed in [31] to find the likelihood  $q_i$  that a given data sample comes from the  $i^{th}$  class. Let the observation feature vector be denoted as  $\mathbf{x}$  and the class label as  $y$  where  $1 \leq y \leq c$  for a  $c$ -class problem. With the assumption that the class-conditional densities  $Pr(\mathbf{x}|y)$  are exponentially distributed [37], the estimate  $\hat{\mu}_{ij}$  of the pairwise class probabilities  $\mu_{ij} \triangleq Pr(y = i|y = j, \mathbf{x})$  is found by fitting a parametric model to the posterior probability density functions  $\hat{\mu}_{ij} = 1/(1 + \exp(\hat{\mathbf{A}}\mathbf{x} + \hat{\mathbf{B}}))$ . The values of  $\hat{\mathbf{A}}$  and  $\hat{\mathbf{B}}$  are estimated by minimizing the *Kullback-Leibler* distance between the parametric pdf and the observed pdf obtained from the training samples. We then find  $q_i \triangleq Pr(y = i|\mathbf{x})$ , the probability that the data sample comes from the  $i^{th}$  class for a  $c$  class SVM, by solving the optimization problem that minimizes the estimation error

$$\begin{aligned} \min_{q_1, q_2, \dots, q_c} \sum_{i=1}^c \left( \sum_{j, j \neq i} (1 - \hat{\mu}_{ij}) q_i - \sum_{j, j \neq i} \hat{\mu}_{ij} q_j \right)^2 \\ \text{subject to } \sum_{i=1}^c q_i = 1, \quad q_i \geq 0, i = 1, 2, \dots, c. \end{aligned}$$

Further details of the algorithm can be found in [31], and a possible implementation is available at [33].

## REFERENCES

- [1] CNet News article: "Who will become the Intel of photography?" at [http://news.com.com/Picture+this+A+new+breed+of+cameras/2009-1006\\_3-5559249.html](http://news.com.com/Picture+this+A+new+breed+of+cameras/2009-1006_3-5559249.html), Feb 2005.
- [2] Business Week News: "How Ampex Squeezes Out Cash: It's Suing High-Tech Giants that Rely on its Patents. Will its Stock keep on Soaring?" at <http://www.businessweek.com>, Apr 2005.
- [3] General Information Concerning Patents. Brochure available online at the U.S. patents website at <http://www.uspto.gov/web/offices/pac/doc/general/infringe.htm>
- [4] D. F. McGahn, "Copyright Infringement of Protected Computer Software: An Analytical Method to Determine Substantial Similarity," *Rutgers Computer and Technology Law Journal*, Vol. 21, No. 1, pp. 88-142, 1995.
- [5] J. L. Wong, D. Kirovski, and M. Potkonjak, "Computational Forensic Techniques for Intellectual Property Protection," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 23, No. 6, pp. 987-994, Jun 2004.
- [6] A. C. Popescu and H. Farid, "Exposing Digital Forgeries by Detecting Traces of Re-sampling," *IEEE Trans. on Signal Processing*, Vol. 53, No. 2, pp. 758-767, Feb 2005.
- [7] J. Fridrich, D. Soukal, and J. Lukas, "Detection of Copy-Move Forgery in Digital Images," *Proc. of Digital Forensics Research Workshop*, Cleveland, OH, Aug 2003.
- [8] A. C. Popescu and H. Farid, "Statistical Tools for Digital Forensics," *Intl. Workshop on Info. Hiding*, Toronto, Canada, 2004.
- [9] M. K. Johnson and H. Farid, "Exposing Digital Forgeries by Detecting Inconsistencies in Lighting," *ACM Multimedia and Security Workshop*, New York, NY, 2005.
- [10] Z. Fan and R. L. de Queiroz, "Identification of Bitmap Compression History: JPEG Detection and Quantizer Estimation," *IEEE Trans. on Image Processing*, Vol. 12, No. 2, pp. 230-235, Feb 2003.
- [11] J. Lukas and J. Fridrich, "Estimation of Primary Quantization Matrix in Double Compressed JPEG Images," *Proc. of Digital Forensics Research Workshop (DFRWS) 2003*, Cleveland, OH, USA, Aug 2003.
- [12] H. Farid, "Blind Inverse Gamma Correction," *IEEE Trans. on Image Processing*, Vol. 10, No. 10, pp. 1428-1433, Oct 2001.
- [13] H. Farid and A.C. Popescu, "Blind Removal of Image Non-Linearities," *Intl. Conference on Computer Vision*, Vancouver, Canada, 2001.
- [14] H. Farid and S. Lyu, "Higher-Order Wavelet Statistics and their Application to Digital Forensics," *IEEE Workshop on Statistical Analysis in Computer Vision*, Feb 2003.
- [15] A. C. Popescu and H. Farid, "How Realistic is Photorealistic?," *IEEE Trans. on Signal Processing*, Vol. 53, No. 2, pp. 845-850, Feb 2005.

- [16] T-T. Ng, S-F. Chang, and Q. Sun, "Blind Identification of Photomontage Using Higher Order Statistics," *IEEE Intl. Symposium on Circuits and Systems*, May 2004.
- [17] Z. J. Geradts, J. Bijhold, M. Kieft, K. Kurosawa, K. Kuroki, and N. Saitoh, "Methods for Identification of Images Acquired with Digital Cameras," *Proc. of the SPIE, Enabling Technologies for Law Enforcement and Security*, Vol. 4232, pp. 505–512, Feb 2001.
- [18] K. Kurosawa, K. Kuroki, and N. Saitoh, "CCD Fingerprint Method - Identification of a Video Camera from Videotaped Images," *Proc. of Intl. Conference on Image Processing*, Vol. 3, pp. 537–540, Kobe, Japan, Oct 1999.
- [19] J. Lukas, J. Fridrich, and M. Goljan, "Determining Digital Image Origin Using Sensor Imperfections," *Proc. of the SPIE, Security, Steganography and Watermarking of Multimedia Contents VII*, Vol. 5681, pp. 249–260, Jan 2005.
- [20] M. Kharrazi, H. T. Sencar, and N. Memon, "Blind Source Camera Identification," *Proc. IEEE Intl. Conference on Image Processing*, Vol. 1, pp. 709–712, Oct 2004.
- [21] S. Bayram, H. T. Sencar, N. Memon, and I. Avcibas, "Source Camera Identification Based on CFA Interpolation," *Proc. IEEE Intl. Conference on Image Processing*, Vol. 3, pp. 69–72, Sep 2005.
- [22] A. C. Popescu and H. Farid, "Exposing Digital Forgeries in Color Filter Array Interpolated Images," *IEEE Trans. on Signal Processing*, Vol. 53, No. 10, part 2, pp. 3948–3959, Oct 2005.
- [23] J. Adams, K. Parulski, and K. Spaulding, "Color Processing in Digital Cameras," *Proc. of IEEE*, Vol. 18, No. 6, pp. 20–30, Nov-Dec 1998.
- [24] J. Adams, "Interaction between Color Plane Interpolation and Other Image Processing Functions in Electronic Photography," *Proc. of the SPIE, Cameras and Systems for Electronic Photography and Sci. Imaging*, Vol. 2416, pp. 144–151, Feb 1995.
- [25] S. Kawamura, "Capturing Images with Digital Still Cameras," *IEEE Micro*, Vol. 18, No. 6, pp. 14–19, Nov-Dec 1998.
- [26] T. A. Matraszek, D. R. Cok Gray, and R. Terry, "Gradient Based Method for Providing Values for Unknown Pixels in a Digital Image," *U.S. Patent no. 5,875,040*, Feb 1999.
- [27] J. F. Hamilton and J. E. Adams, "Adaptive Color Plane Interpolation in Single Sensor Color Electronic Camera," *U.S. Patent no. 5,629,734*, May 1997.
- [28] A. Swaminathan, M. Wu, and K. J. Ray Liu, "Non-Intrusive Forensic Analysis of Visual Sensors Using Output Images," *Proc. of IEEE Conference on Acoustic, Speech and Signal Processing*, Toulouse, France, May 2006.
- [29] A. Swaminathan, M. Wu, and K. J. Ray Liu, "Component Forensics of Digital Cameras: A Non-Intrusive Approach," *Proc. of Conference on Information Sciences and Systems*, Princeton, NJ, Mar 2006.
- [30] C. F. van Loan, *Introduction to Scientific Computing*, Prentice Hall, 2000.
- [31] T-F. Wu, C-J. Lin, and R. C. Weng, "Probability Estimates for Multi-class Classification by Pairwise Coupling," *Journal of Machine Learning Research*, Vol. 5, pp. 975–1005, 2004.
- [32] C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery*, Vol. 2, pp. 121–168, 1998.
- [33] C-C. Chang and C-J. Lin, "LIBSVM: A Library For Support Vector Machines," 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [34] Information about Fujifilm Super CCD cameras. Online available: <http://www.fujifilm.com/supercdd/>
- [35] A. K. Jain, *Fundamentals of Digital Image Processing*, Pentice Hall, 2000.
- [36] E. Chang, S. Cheung, and D. Y. Pan, "Color Filter Array Recovery using a Threshold-Based Variable Number of Gradients," in *Proc. of the SPIE, Sensors, Cameras, and Applications for Digital Photography*, Vol. 3650, pp. 36–43, Mar 1999.
- [37] J. Platt, "Probabilistic Outputs For Support Vector Machines and Comparison to Regularized Likelihood Methods," in *Advances in Large Margin Classifiers*, Cambridge, MA, 2000.
- [38] H. Gou, A. Swaminathan, and M. Wu, "Robust Scanner Identification based on Noise Features," submitted to 2007 Proc. of the SPIE, Security, Steganography and Watermarking of Multimedia Contents IX (under review).
- [39] Information about Foveon X3 sensors. Online available: <http://www.foveon.com/>