Nonintrusive Component Forensics of Visual Sensors Using Output Images

Ashwin Swaminathan, *Student Member, IEEE*, Min Wu, *Senior Member, IEEE*, and K. J. Ray Liu, *Fellow, IEEE*

Abstract—**Rapid technology development and the widespread use of visual sensors have led to a number of new problems related to protecting intellectual property rights, handling patent infringements, authenticating acquisition sources, and identifying content manipulations. This paper introduces nonintrusive component forensics as a new methodology for the forensic analysis of visual sensing information, aiming to identify the algorithms and parameters employed inside various processing modules of a digital device by only using the device output data without breaking the device apart. We propose techniques to estimate the algorithms and parameters employed by important camera components, such as color filter array and color interpolation modules. The estimated interpolation coefficients provide useful features to construct an efficient camera identifier to determine the brand and model from which an image was captured. The results obtained from such component analysis are also useful to examine the similarities between the technologies employed by different camera models to identify potential infringement/licensing and to facilitate studies on technology evolution.**

Index Terms—**Camera identification, component forensics, digital forensic signal processing, evolutionary forensics, infringement/licensing forensics, nonintrusive image forensics.**

I. INTRODUCTION

VISUAL sensor technologies have experienced tremendous growth in recent decades. The resolution and quality of electronic imaging has been steadily improving, and digital cameras are becoming ubiquitous. The shipment of digital cameras alone has grown from U.S.$46.4 million in 2003 to U.S.$62 million in 2004, forming an approximately U.S.$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 has arisen 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 nonintrusive approach. In this paper, we propose to develop a new forensic methodology called nonintrusive 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 prosecutions have also grown tremendously, sometimes in the range of billions of dollars. For example, the Ampex Corporation has more than 600 patents related to digital cameras; and based on one of the patents, it has received more than U.S.$275 million compensation from lawsuits and settlements involving patent infringement cases with many digital camera vendors [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 considered one of the most 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 integrated-circuit (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 an idea rather than the idea itself and, thus, could potentially lead to misleading conclusions [4], [5]. Component forensics is an important methodology 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 to build 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 images obtained directly from a camera. This, in turn, will facilitate tampering forensics to determine if there has been 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 evidence:

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 its processing blocks.

2) **Semi Nonintrusive Forensics**: An analyst has access to the device as a black box. He or 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) **Nonintrusive Forensics**: An analyst does not have access to the device in question. He or 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 nonintrusive component forensic methodology on visual sensors, while the suggested techniques can be appropriately modified and extended to other types of acquisition models, and sensing technologies.[1] 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 of different models released at different years and/or at various 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 the image capturing process in

digital cameras and our problem formulation in Section III. 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. Final conclusions are drawn in Section VIII.

II. RELATED PRIOR WORKS ON FORENSIC ANALYSIS

While a growing amount of signal processing research in 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 nonintrusive image forensics falls into the following two main categories.

In the forgery detection literature, there have been works that consider a tampered picture as an image that 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, or lighting inconsistencies [7], copy-paste operations [8], irregular noise patterns [9], and alterations in the correlation introduced by color interpolation [10]. For image compression, such as JPEG that involves quantization in the discrete cosine transform (DCT) domain, the statistical analysis based on binning techniques has been used to estimate the quantization matrices [11], [12]. Higher order statistics, such as the bispectrum, have been proposed to identify contrast changes, gamma correction [13], and other nonlinear point operations on images [14]; wavelet-based features have been used to detect image tampering [15] and identify photorealistic images [16], and physics-motivated features have been introduced to distinguish photographic images and computer graphics [17]. Most of these techniques mentioned before primarily target finding the processing steps that occur 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 nonintrusive image forensics concerns camera identification. Camera pixel defects [18], pattern noise associated with the nonuniformity of dark currents on camera CCDs [19], and pattern noise [20] 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 for 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 [21], [22]. Features, such as average pixel values, RGB pairs correlation, and neighbor center of mass are used in [21]. In [22], the authors employ the Expectation–Maximization algorithm from [6], [10] to extract spectral features related to color interpolation and use these features to build a camera-brand classifier. Although good results

---

[1]For example, our recent work examined a modified image acquisition model that is tailored to digital scanners and has obtained good forensics results for scanners [41].
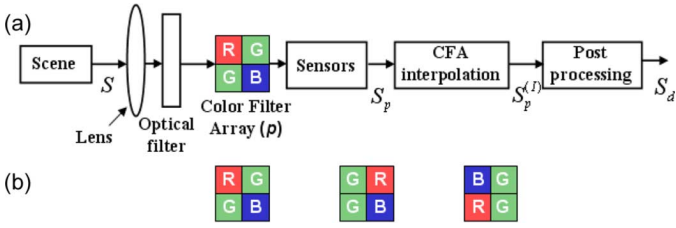
Fig. 1. (a) Image capturing model in digital cameras. (b) Sample color filter arrays.

were reported in distinguishing pictures taken in controlled scenarios with three different cameras, its ability to differentiate cameras needs to be further investigated when under nonintrusive testing conditions and in the presence of compression noise. 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. IMAGE CAPTURING MODEL AND PROBLEM FORMULATION

In this section, we discuss the image capturing model by digital cameras and present our problem formulation. As illustrated by the image capturing model in Fig. 1(a), light from a scene passes through a lens and optical filters, and is finally recorded by an array of sensors. Few consumer-level color cameras directly acquire full-resolution information for all three primary colors (usually red, green, and blue).[2] This is not only because of the high cost in producing a full-resolution sensor for each of the 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 CFA to sample real-world scenes.

A CFA 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 the left corner of Fig. 1(b), is one of the most popular CFA patterns. It uses a square 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 (also known as demosaicking) is an important step to produce an output image with full resolution for all three color components [10], [24]. After interpolation, the three images corresponding to the red, green, and the blue components go though a post-processing stage. In this stage, various 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. $S(x, y, c)$ 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 3-D matrix S_p of the form

$$S_p(x, y, c) = \begin{cases} S(x, y, c), & \text{if } p(x, y) = c, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

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

The performance of color interpolation directly affects the quality of the image captured by a camera [23]–[25]. There have been several commonly used algorithms for color interpolation. These algorithms can be broadly classified into two categories, namely, nonadaptive and adaptive algorithms. Nonadaptive algorithms apply the same type for interpolation for all pixels in a group. Some typical examples of nonadaptive 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 have 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 A. The CFA interpolated image undergoes postprocessing 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 nonintrusive forensic analysis where we use sample images obtained from a digital camera under diverse and uncontrolled scene settings to determine the algorithms (and their parameters) employed in internal processing blocks. In particular, given an output image S_d, we focus on finding the CFA 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 subsequent sections, we describe our proposed methodology and algorithms, and demonstrate their effectiveness with detailed simulation results and case studies.

IV. FORENSIC ANALYSIS AND PARAMETER ESTIMATION OF CAMERA COMPONENTS

In this section, we develop a robust and nonintrusive 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

---

[2]New digital cameras employing the Foveon X3 sensor, such as Sigma SD9 and Polaroid x530, capture all three colors at each pixel location [42].

local region through texture classification and linear approxima-tion, 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 an RGB type of CFA with a fixed periodicity of 2×2 that can be represented as

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

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 2×2 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]. We divide the image into three kinds of regions based on the gradient features in a local neighbor-hood. Defining $I_{x,y} = S_d(x, y, p(x, y))$, the horizontal and vertical gradients at the location (x, y) can be found from the second-order gradient values using

$$
H_{x,y} = |I_{x,y-2} + I_{x,y+2} - 2I_{x,y}|, \tag{2}
$$

$$
V_{x,y} = |I_{x-2,y} + I_{x+2,y} - 2I_{x,y}|. \tag{3}
$$

The image pixel at location (x, y) is classified into one of the three categories.

1) Region $\Re_1$ contains those parts of the image with a sig-nificant horizontal gradient for which $(H_{x,y} - V_{x,y}) > T$, where T is a suitably chosen threshold.
2) Region $\Re_2$ contains those parts of the image with a sig-nificant vertical gradient and is defined by the set of points for which $(V_{x,y} - H_{x,y}) > T$.
3) Region $\Re_3$ consists of 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 $\Re_m (m = 1, 2, 3)$ and three color channels (R, G, B) of the image.

Let the set of N_e equations with N_u unknowns for a partic-ular region and color channel be represented as $\mathbf{Ax} = \mathbf{b}$, 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 interpo-lation coefficients to be estimated. To cope with possible noisy pixel values in $\mathbf{A}$ and $\mathbf{b}$ due to other in-camera operations fol-lowing 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, this can be written as

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

Here, $\|.\|_F$ denotes the Frobenius norm of the matrix, so that

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

The solution to the minimization problem can be written as

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

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

B. Finding the Interpolation Error and the CFA Sampling Pattern

Once we find the interpolation coefficients in each region, we use them to reinterpolate the sampled CFA output in the corre-sponding regions $\Re_m$ to obtain an estimate of the final output image $\hat{S}_d^{(p)}$. Here, the superscript p denotes that the output esti-mate is based on the choice of the CFA pattern p. The pixel-wise difference between the estimated final output and the actual camera output image is $e^{(p)} = \hat{S}_d^{(p)} - S_d$. The interpolation error matrix $e^{(p)}$ of dimension $H \times W \times C$ is obtained for all candi-date 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

$$
\varepsilon(p) = w_R \left\| e^{(p)}(., ., 1) \right\|_F^2 + w_G \left\| e^{(p)}(., ., 2) \right\|_F^2
$$
$$
+ w_B \left\| e^{(p)}(., ., 3) \right\|_F^2. \tag{7}
$$

The CFA 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 es-timated pattern. The constants w_R, w_G, and w_B denote the cor-responding weights used for the three color components (red,
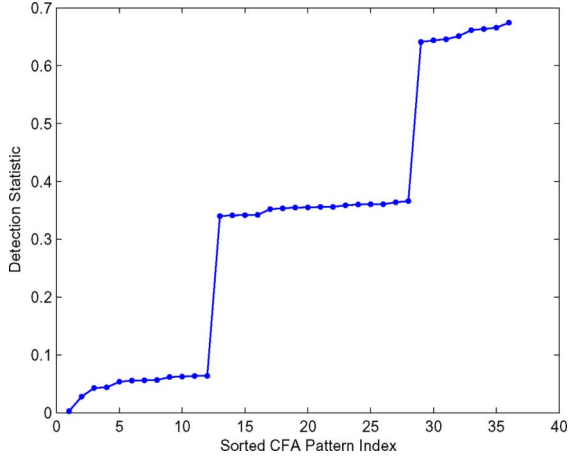
Fig. 2. Sorted detection statistics in terms of normalized overall error for different candidate search patterns.

Fig. 3. Sample CFA patterns from the three clusters.

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 also be processed to obtain further forensic evidence, as 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 hierarchial approach. As an example, we synthetically generate a 512×512 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

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

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 at 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. A 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 A. This observation forms the basis for the heuristic discussed in this subsection to reduce the search space of the CFA patterns.

Fig. 3 shows sample patterns from these three clusters. Cluster 1 includes all 2×2 patterns that have the same color along diagonal directions (either along the main diagonal or offdiagonal), chosen among the three colors (red, green, or
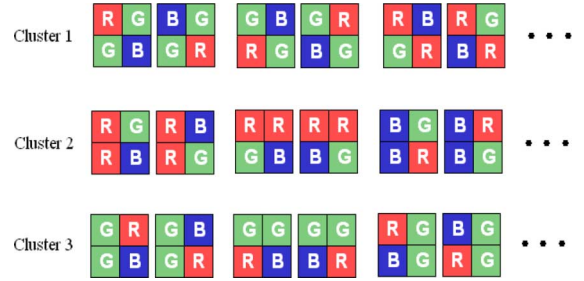
blue). The remaining two spots can be filled in two different ways, giving a total of 12 such patterns in the first cluster. Cluster 2 and Cluster 3 consist of patterns that have the same color along the horizontally (or vertically) adjacent blocks of the 2×2 grid. Cluster 2 has either red or blue color 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 the first cluster give lower errors compared to the other clusters. The patterns from Cluster 3 give 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 (7).

The observation of clustering of patterns into three groups helps us develop a heuristic to reduce the search space of CFA patterns. We first divide 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 cluster that the actual CFA pattern is most likely to belong. Finally, a full search is performed on the chosen cluster 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 is 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 two green color components if he or 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 that is trained on the coefficients of the c candidate interpolation methods. The probability that a given test sample comes from the ith 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 B for the readers' reference. The confidence score η on the decision is then defined as

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

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 $0 \leq z \leq (1/2)$. The argument to the Υ function in (9) measures the entropy difference between the distribution $\{q_i\}$ and a discrete uniform distribution, and the final value of η is normalized to the range of [0, 1] to represent a probability.

To verify that the proposed metric η can reflect the confidence level, we examine two extreme cases. When $\mathbf{q} = [1, 0, 0, \ldots, 0]$, the decision of choosing the first class is made with a very high confidence and $\eta = 1$. And when $\mathbf{q} = [(1/c)+\epsilon, (1/c)-(\epsilon/(c-1)), (1/c) - (\epsilon/(c - 1)), \ldots, (1/c) - (\epsilon/(c - 1))]$ where ϵ 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 very low confidence and η also approaches zero. For other values of $\mathbf{q}$ between these two extreme cases, the value of η 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 sampled image is interpolated using one of the six interpolation methods reviewed in Appendix A, 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 512×512.

1) Simulation Results Under No Postprocessing: We test the proposed CFA pattern and color interpolation identification algorithms on this synthetic dataset. In the noiseless case without postprocessing, we observe no errors in estimating the CFA pattern. We use a 7×7 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) classifier with a third-degree polynomial kernel [32], [33] is used to identify the interpolation method. We randomly choose 8 out of the 20 images 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 random set of images each time. The classifier is 100% accurate in identifying the correct color interpolation algorithm without any errors.

2) Simulation Results With Postprocessing: As mentioned before, postprocessing, such as color correction and compression, is commonly accomplished 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 postprocessing operations performed in cameras.

In this work, we primarily focus on JPEG compression and additive noise, and study the performance under these distortions. Other postprocessing operations, such as color correction and white balancing, are typically multiplicative, where the final image is obtained by multiplying the color-interpolated image by appropriately chosen constants in the camera color space. In most commercial cameras, white balancing is performed in the XYZ color space [34], and the inverse transformation may be applied before estimating the color interpolation coefficients. The multiplicative factors used in white balancing operations operate on each color channel separately [35] and, therefore, white balancing operations do not significantly affect our solution of the color interpolation coefficients. Gamma correction can be estimated from the final output images [13] and can be undone before computing the interpolation coefficients. For the results presented in this subsection, 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.

a) Performance Results Under JPEG Compression: JPEG compression is an important postprocessing operation that is commonly 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 $\{20, 30, \ldots, 80, 90, 99\}$. We find 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 or 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 the multiple-scene case. The performance of the proposed color interpolation identification 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 distinct images for each of the six interpolation methods) for training, and test with the remaining eight 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 of 80–100 and the average performance reduces to 80%–85% for quality factors from 50 to 80.

Alternatively, if a forensic analyst has access to the camera, he or she can perform 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 of the cameras under study
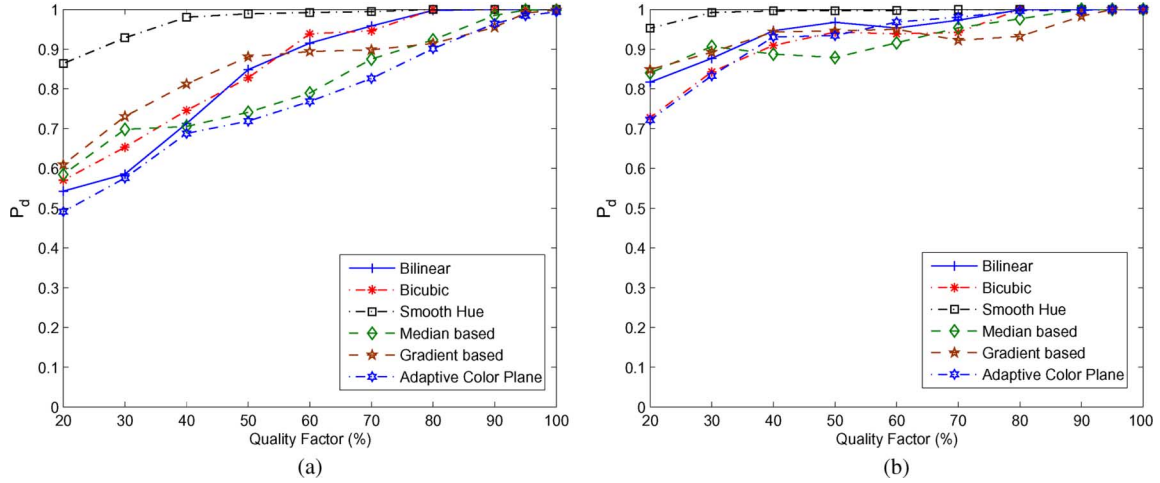
Fig. 4. Fraction of images for which the color interpolation technique is correctly identified under different JPEG compression quality factors for (a) the multiple-scene case and (b) the single-scene case. The testing results here are for the synthetic dataset.
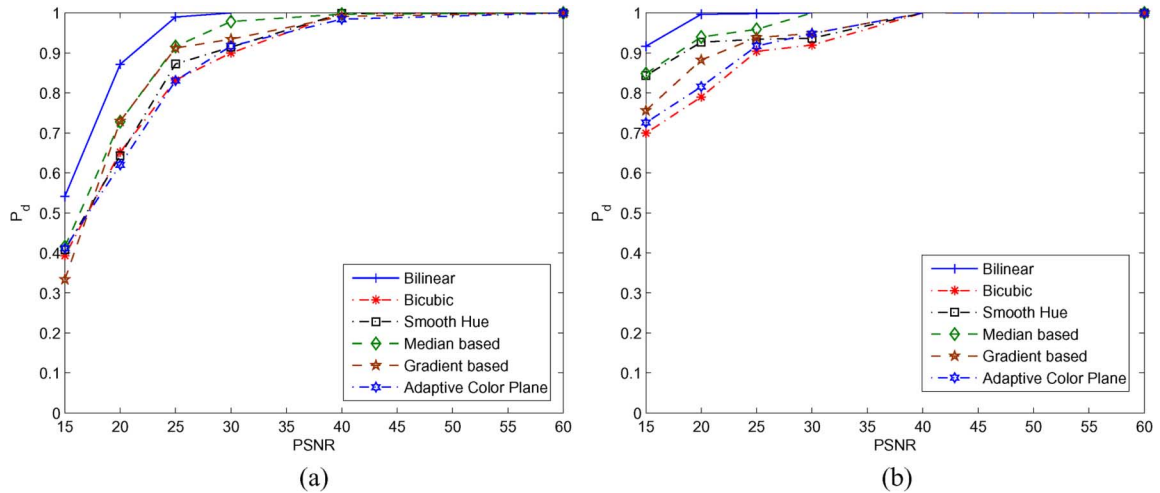

Fig. 5. Fraction of images for which the color interpolation technique is correctly identified under different noise PSNRs for (a) the multiple-scene case and (b) the single-scene case. The testing results here are for the synthetic dataset.

in order to improve the estimation accuracy and increase the confidence level on his or her final judgement. We call this situation the single-scene case. The single-scene case corresponds to the semi nonintrusive 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 eight 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 higher confidence compared to the multiple-scene case. The accuracy can be further improved using more images with representative characteristics for training. This suggests that with an increasing number of well-designed image inputs to the system, the detection performance can be enhanced.

b) Performance Results Under Additive Noise: Additive noise can be used to model other kinds of random postprocessing 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 (PSNRs) of 15, 20, 30, and 40 dB, respectively. The correct CFA pattern was identified in all but one case, and the only error occurred at an extremely low PSNR of 15 dB for an image interpolated with the adaptive color plane method. Even in this case, the correct pattern came in the top three results.

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 (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.

B. Results on Camera Data

A total of 19 camera models as shown in Table I are included in our experiments. For each of the 19 camera models, we have

TABLE I
CAMERA MODELS USED IN EXPERIMENTS

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 Cybershot DSC P7	19	Epson PhotoPC 650
10	Sony Cybershot DSC P72		

Fig. 6. Super CCD sensor pattern.

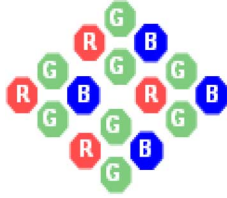Fig. 7. Sample CFA patterns for (a) Canon EOS digital rebel and (b) Fujifilm Finepix S3000.

collected about 40 images. The images from different camera models are captured under uncontrolled conditions-different sceneries, different lighting situations, and compressed under different JPEG quality factors as specified by default values in each camera. The default camera settings (including image size, color correction, auto white balancing, JPEG compression, etc.) are used in image acquisition. From each image, we randomly choose five nonoverlapping 512×512 blocks per image and use them for subsequent analysis. Thus, our data base consists of a total of 3800 different 512×512 pictures with 200 samples for each of the 19 camera models.

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

Among RGB-type CFA patterns, several layouts of the three types of color filters have been used in practice. The 2×2 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 [36] 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 17) that uses super CCD [36] in our data base.

As an initial step, we try to estimate the CFA pattern from the output images using the algorithm described in Section IV. The estimation results show with high confidence that all of the cameras except Fujifilm Finepix A500 (camera 17) use shifted versions of the Bayer color filter array as their CFA pattern. For instance, the estimated 2×2 CFA that minimized the fitting
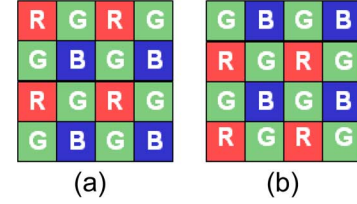
errors on JPEG images from Canon EOS Digital Rebel (camera 6) and the Fujifilm Finepix S3000 (camera 16) are shown in Fig. 7(a) and (b), respectively. The estimation results perfectly match these cameras' ground-truth data obtained by reading the headers of the raw images files produced by the two cameras.

When testing the images from Fujifilm Finepix A500 (camera 17) with the same 36 square patterns in the CFA pattern search space, we notice that the best 2×2 pattern in the search space is still a shifted version of the Bayer pattern. However, we observe that the minimum error ε, as given by (7), is larger than the ones obtained from other square-CFA cameras. Therefore, the overall decision confidence is lower for this super CCD camera compared to the other cameras in the data base. 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 be an indication 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 7×7 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 $\Re_1$ that corresponds to areas having a significant horizontal gradient, we observe that the value of the coefficients in the vertical direction (0.435 and 0.441) are significantly higher than those in the horizontal directions (0.218 and 0.204). This indicates that the interpolation is done along the edge which, in this case, is oriented along the vertical direction. Similar corresponding inferences can be made from coefficients in region $\Re_2$ of a significant vertical gradient. Compared to these two regions, the coefficients in region $\Re_3$ have almost equal values in all four directions, and do not have any directional properties. Moreover, careful observation of the coefficients in region $\Re_3$ reveals their close resemblance to the bicubic interpolation coefficients shown in Fig. 8(d). This suggests that it is very 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

0	0.007	0	0.027	0	0.014	0
0.008	0	-0.062	0	-0.070	0	0.003
0	-0.041	0	0.435	0	-0.032	0
-0.004	0	0.218	1.000	0.204	0	0.001
0	-0.034	0	0.441	0	-0.038	0
0.005	0	-0.075	0	-0.064	0	0.002
0	0.013	0	0.030	0	0.012	0

(a)

0	0.006	0	0.009	0	0.009	0
0.005	0	-0.070	0	-0.069	0	0.007
0	-0.023	0	0.213	0	-0.019	0
0.019	0	0.414	1.000	0.399	0	0.029
0	-0.014	0	0.214	0	-0.020	0
0.002	0	-0.074	0	-0.064	0	0.006
0	0.008	0	0.011	0	0.005	0

(b)

0	-0.005	0	0.008	0	0.001	0
0.018	0	-0.079	0	-0.050	0	0.006
0	-0.021	0	0.315	0	-0.023	0
0.003	0	0.325	1.000	0.301	0	0.023
0	-0.023	0	0.301	0	-0.015	0
-0.006	0	-0.061	0	-0.051	0	0.010
0	0.002	0	0.018	0	-0.001	0

(c)

0	0	0	0.004	0	0	0
0	0	-0.035	0	-0.035	0	0
0	-0.035	0	0.316	0	-0.035	0
0.004	0	0.316	1.000	0.316	0	0.004
0	-0.035	0	0.316	0	-0.035	0
0	0	-0.035	0	-0.035	0	0
0	0	0	0.004	0	0	0

(d)

Fig. 8. Interpolation coefficients for the green channel for one sample image taken with the Canon Powershot A75 camera for (a) Region $\Re_1$ with significant horizontal gradient, (b) Region $\Re_2$ with significant vertical gradient, (c) smooth region $\Re_3$, (d) coefficients of bicubic interpolation.

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 [37].

VI. CASE STUDIES AND APPLICATIONS OF NONINTRUSIVE FORENSIC ANALYSIS

In this section, we present case studies to illustrate the applications of the proposed nonintrusive 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 focus on nonsmooth regions and use the coefficients obtained from the horizontal gradient regions $\Re_1$ and vertical gradient regions $\Re_2$ as features to construct a camera brand identifier.

To obtain more reliable forensic evidence from the input image for camera identification, we first preprocess the image by edge detection to locate five significant 512×512 blocks with the highest absolute sum of gradient values. The interpolation coefficients corresponding to the regions $\Re_1$ and $\Re_2$, from all three color channels, estimated from these 512×512 blocks are used as features for identification.

We use a classification-based framework to identify the camera brand. For each camera in the data base, we collect 40 different images and obtain 200 different 512×512 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. A 9-camera-brand SVM classifier with a polynomial kernel function [33] is constructed with 50% of the images randomly chosen 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 classification rate of 90% for nine camera brands.

The above results demonstrate the effectiveness of using 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-A2. 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.[3] 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 negligible difference from the nongamma correction case of 90% suggests that the camera identification results are invariant to gamma correction in digital cameras.

As the problem of camera brand identification only received attention recently, there is a very limited amount of related work to compare with. Some algorithms were developed recently in [21] and [22], where the authors test their algorithms for 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). The best

[3]In a general scenario, the value of γ can be estimated from the output images [13] 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%

TABLE III
CONFUSION MATRIX FOR ALL CAMERAS. THE MATRIX IS DIVIDED BASED ON DIFFERENT CAMERA MAKES. THE VALUES BELOW THE THRESHOLD $\lambda = 1/19$ 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	*	*	*	*
Fujifilm	16	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	0.96	*	*	*
	17	*	*	0.08	*	*	*	*	*	*	*	*	*	0.20	*	*	*	0.68	*	*
Kodak	18	*	*	*	*	*	*	*	*	*	*	*	*	*	0.08	*	*	*	0.88	*
Epson	19	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	1.00

performance initially reported is 84% on three brands [22], and sensitive to other in-camera processing such as compression owing to the dependence on image content by the null-based spectral features employed in [22]. In concurrent of the present paper, further improvement has been made to [22] by separately obtaining the coefficients from smooth and nonsmooth regions of each image, leading to an enhanced classification accuracy of 96% for three camera brands [38]. Compared to these alternative 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 mitigate the impact from noise, compression, and other in-camera processing. As a result, the features obtained from the proposed component forensics methodologies are able to achieve a high classification accuracy over a much larger data base with 19 camera models from nine different brands. Further, as to be demonstrated later in this section, the proposed component forensic techniques has a broader goal of identifying the algorithms and parameters employed in various components in digital cameras, and are not restricted to camera brand identification.

B. Identifying Camera Model From Output Images

Our results in the previous subsection demonstrate the robustness of nonintrusively 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 a 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. Out of these 200 images, a randomly chosen 125 images are used for training and the remaining are for testing with a 19-camera model SVM classifier. The simulation is repeated 500 times with different training sets and the average confusion matrix is shown in Table III. 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 nonzero value greater than the threshold of 1/19. For example, among the Canon Powershot S410 (camera 3) images, 20% were classified as belonging to Canon Powershot S400 (camera 2). A similar trend is also observed for images from other Canon models. 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 of the 200 images from 18 cameras except Canon Powershot S410 (camera 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 as one of the remaining Canon models. The reverse trend is also observed when we train with all of the images except Canon Powershot S400 (camera 2) and use these images for testing. 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.

A similar trend is also observed for the two Sony cameras in our data base. We note that around 66% of the Sony Cybershot DSC P7 models are classified as Sony Cybershot DSC P72 models 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 the Nikon E4300 camera model. This suggests closeness between the interpolation coefficients in the feature space.

2) Quantifying Similarity in Color Interpolation With a Divergence Score: From our preliminary analysis in Section V-B, we observe that the majority of cameras use similar kinds of interpolation techniques in handling smooth regions. We thus focus our attention on the type of interpolation used by a camera in the nonsmooth regions. We extend our interpolation coefficient estimation model in Section IV-B to explicitly target nonsmooth regions in the image. To do so, we divide the image into eight types of regions depending on the relative gradient estimates in eight directions (namely north, east, west, south, northeast, northwest, southeast, and southwest). The gradient values can be obtained following the threshold-based variable number of gradients (VNG) algorithm [39]. For example, the gradient in the north direction J_N is obtained by using

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

where $I_{x,y} = S_d(x,y,p(x,y))$ represents the image pixel sample. Similar expressions for gradients in the remaining seven directions can be developed to find the local gradient values [39]. Once these gradients are obtained, they are compared to a threshold to divide the image into eight types of texture regions. The interpolation coefficients are obtained in each region by solving a set of linear equations as given by (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 reinterpolate 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 200 images taken by every camera in the 19-camera dataset, we estimate the CFA parameters (eight sets of coefficients each with a dimension of 5×5),[4] feed them as input to the above classifier and record the classification results. The probabilistic SVM framework is used in classification and the soft decision values are recorded for each image [31] (refer to Appendix B 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 data base, 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 $\underline{g} = [g_1, g_2, \ldots, g_N]$, where g_k gives the probability that the input image employs the interpolation algorithm-k. Such probability vectors are obtained for every image in the data base and the average performance is computed for each camera model. Let the average classification results for camera model i be represented by the vector $\underline{\pi_i} = [\pi_{i1}, \pi_{i2}, \ldots, \pi_{iN}]$, where π_{ik} is the average probability for an image from camera model i to be classified as using the interpolation algorithm k. The π_{ik}'s are estimated using soft decision values obtained by 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 φ_{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 separately obtained in each of the eight types of regions by training with synthetic data and

---

[4]A kernel size of 5×5 is chosen in this case to limit the total number of coefficients, and to make the total number of features to be on the same order of magnitude as the previous case in Section VI-B where we used a kernel size of 7×7 and three gradient-based regions.

TABLE IV
DIVERGENCE SCORES FOR DIFFERENT CAMERA MODELS AS INDEXED IN TABLE I. THE VALUES BELOW OR EQUAL TO 0.06 ARE SHADED, AND THE * INDICATES ZERO SIMILARITIES BETWEEN THE SAME CAMERA MODELS BY DEFINITION

		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	0.42	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
Fujifilm	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
	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
Kodak	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	*

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 an overall divergence score indicates that the two cameras are similar and are likely to use very similar kinds 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 shaded. We observe from the table that most cameras from the same brand are likely to use similar kinds of interpolation algorithms. This is especially evident for some models of Canon and Minolta used in our analysis.

The divergence score between the two Canon models—S400 and S410—is 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 S304 and DiMage F100—and between the two Sony models—Cybershot DSC P7 and P72. The metric is close to zero in all of 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 Olympus C765UZ (camera 12) and Casio QV2000UX (camera 15) is only 0.01, which suggests a close resemblance in the type of interpolation used by these two cameras. We also see that the two cameras showing similarity in the leave-one-out experiment, Nikon E4300 (camera 7) and Minolta DiMage S304 (camera 13), give quite a low divergence score as a quantitative indication of their similarity.

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 in-
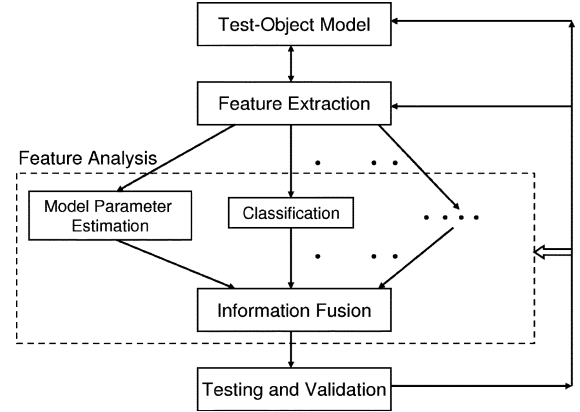

Fig. 9. Proposed forensic analysis methodology.

formation 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 nonintrusive forensic analysis to a methodology applicable to a broad range of devices. Let $O_1, O_2, \ldots, O_{N_o}$ be the sample outputs obtained from the test device that we model as a black box, and $C_1, C_2, \ldots, C_{N_c}$ 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 processing block C_y. A general forensic analysis framework is composed of the following processing steps as shown in Fig. 9.

1) Modeling of the test device: As the first step of forensic analysis, a model is constructed for the object under study. This modeling helps break down the test device into a set of individual processing components $C_1, C_2, \ldots, C_{N_c}$ and systematically study the effect of each block on the final outputs obtained with the test object.

2) Feature extraction: The forensic analyst identifies a set of features that has good potential to help identify the algorithms used in the yth device component C_y. These features are based on the final output data and are chosen to uniquely represent each algorithm used. For the case of digital cameras, we have used the estimated color interpolation coefficients as features for forensic analysis in this paper. The parameters of other components, such as white balancing constants and gamma correction values, are also possible features to incorporate.

3) Feature analysis and information fusion: We analyze the features extracted 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 analysis technique 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 a 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 be proven to be useful forensic evidence and 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 type 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 nonintrusive 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 datasets 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 nonintrusively 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 provide a great source of information for patent infringement cases, intellectual property-rights management, and technology evolution studies 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 nonintrusive estimations of the raw data acquired directly by the imaging sensor prior to 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 A
SOME POPULAR COLOR INTERPOLATION ALGORITHMS

There have been many algorithms employed in practice for CFA interpolation. In this appendix, we briefly review some of the popular methods. For a detailed survey, the readers are referred to [24]. Color interpolation methods can be broadly classified into two main categories, namely, adaptive and nonadaptive methods, depending on their adaptability to the image content. While nonadaptive methods use the same pattern for all pixels in an 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 nonadaptive interpolation schemes. In these algorithms, the pixel values are interpolated according to the following equation [10]:

$$S_{\text{int}}(x,y,c) = \sum_{u,v \in -N_g}^{N_g} h_c(u,v) S_{\text{raw}}(x-u, y-v, c)$$

where S_{raw} are the original raw values obtained from the sensor with $S_{\text{raw}}(.,.,1)$ representing the red color and so on, S_{int} de-

notes the interpolation results, and h_c denotes the 2-D filters of dimension $N_g \times N_g$ used in interpolation. In a general case, h_c may be dependent on the color channel. Let h_r, h_g, and h_b denote the values taken by h_c for red, green, and blue colors, respectively. For the bilinear case, these filters are given by

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

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 algorithm, the green channel is first interpolated by using bilinear interpolation to yield $S_{int}(.,.,2)$. The red components are then obtained by interpolating the ratios of "red/green" via

$$\frac{S_{int}(x,y,1)}{S_{int}(x,y,2)} = \frac{1}{2} \left(\frac{S_{raw}(x,y-1,1)}{S_{int}(x,y-1,2)} + \frac{S_{raw}(x,y+1,1)}{S_{int}(x,y+1,2)} \right).$$

The blue components can be obtained similarly by interpolating the "blue/green" ratios.

In median filter-based algorithms, the three channels are first interpolated using bilinear interpolation. Then, the differences "red-green," "red-blue," and "green-blue," 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 [10]. For example, the green color component at the location of the red color filter is obtained as

$$S_{int}(x,y,2) = S_{raw}(x,y,1) - M_{rg}(x,y).$$

All of the methods described above are nonadaptive in nature and do 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

$$J_h(x,y) = |I_{x,y-2} + I_{x,y+2} - 2I_{x,y}|$$
$$J_v(x,y) = |I_{x-2,y} + I_{x+2,y} - 2I_{x,y}|$$

where $I_{x,y} = S_{raw}(x,y,p(x,y))$ and p is the CFA pattern matrix (e.g., Bayer pattern) with $p(x,y) = 1, 2,$ or 3, indicating that the CFA pattern at the $(x,y)^{th}$ pixel is red, green, or blue, respectively. 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, "red-green" and "blue-green" along the edge, respectively.

The adaptive color plane interpolation method is an extension of the gradient based method. Here, the horizontal and vertical gradients are estimated using

$$J_h(x,y) = |I_{x,y-1} - I_{x,y+1}| + |I_{x,y-2} + I_{x,y+2} - 2I_{x,y}|,$$
$$J_v(x,y) = |I_{x-1,y} - I_{x+1,y}| + |I_{x-2,y} + I_{x+2,y} - 2I_{x,y}|.$$

Unlike the simple gradient-based method, the interpolation of one color component here 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 B
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 ith class. Let the observation feature vector be denoted as $\mathbf{x}$ and the class label as y, where $1 \le y \le c$ for a c-class problem. With the assumption that the class-conditional densities $Pr(\mathbf{x}|y)$ are exponentially distributed [40], the estimate $\hat{\mu}_{ij}$ of the pairwise class probabilities $\mu_{ij} \triangleq Pr(y = i|y = i \text{ or } j, \mathbf{x})$ is found by fitting a parametric model to the posterior probability density functions $\hat{\mu}_{ij} = 1/(1 + \exp(\hat{\mathbf{a}}\mathbf{x} + \hat{b}))$. The values of $\hat{\mathbf{a}}$ and $\hat{b}$ are estimated by minimizing the Kullback–Leibler distance between the parametric pdf defined earlier and the one observed 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 ith class for a c-class SVM, by solving the optimization problem that minimizes the following:

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

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

REFERENCES

[1] CNET News article, Who Will Become the Intel of Photography? [Online]. Available: http://www.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? [Online]. Available: http://www.businessweek.com, Apr. 2005.

[3] General Information Concerning Patents Brochure available online at the U.S. patents website [Online]. Available: 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 Comput. Technol. Law J.*, 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. Comput.-Aided Design Integr. Circuits Syst.*, 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. Signal Process.*, vol. 53, no. 2, pt. 2, pp. 758–767, Feb. 2005.

[7] M. K. Johnson and H. Farid, "Exposing digital forgeries by detecting inconsistencies in lighting," in *Proc. 7th ACM Multimedia and Security Workshop*, New York, Aug. 2005, pp. 1–9.

[8] J. Fridrich, D. Soukal, and J. Lukas, "Detection of copy-move forgery in digital images," in *Proc. Digital Forensics Research Workshop*, Cleveland, OH, Aug. 2003.

[9] A. C. Popescu and H. Farid, "Statistical tools for digital forensics," in *Proc. 6th Intl. Workshop on Info. Hiding*, Toronto, ON, Canada, May 2004, vol. 3200, pp. 128–147.

[10] ——, "Exposing digital forgeries in color filter array interpolated images," *IEEE Trans. Signal Process.*, vol. 53, no. 10, pt. 2, pp. 3948–3959, Oct. 2005.

[11] Z. Fan and R. L. de Queiroz, "Identification of bitmap compression history: JPEG detection and quantizer estimation," *IEEE Trans. Image Process.*, vol. 12, no. 2, pp. 230–235, Feb. 2003.

[12] J. Lukas and J. Fridrich, "Estimation of primary quantization matrix in double compressed JPEG images," in *Proc. Digital Forensics Research Workshop*, Cleveland, OH, Aug. 2003.

[13] H. Farid, "Blind inverse gamma correction," *IEEE Trans. Image Process.*, vol. 10, no. 10, pp. 1428–1433, Oct. 2001.

[14] H. Farid and A. C. Popescu, "Blind removal of image non-linearities," in *Proc. IEEE Int. Conf. Computer Vision*, Vancouver, BC, Canada, Jul. 2001, vol. 1, pp. 76–81.

[15] H. Farid and S. Lyu, "Higher-order wavelet statistics and their application to digital forensics," presented at the IEEE Workshop on Statistical Analysis in Computer Vision, Madison, WI, Feb. 2003.

[16] A. C. Popescu and H. Farid, "How realistic is photorealistic?," *IEEE Trans. Signal Process.*, vol. 53, no. 2, pt. 2, pp. 845–850, Feb. 2005.

[17] T.-T. Ng, S.-F. Chang, J. Hsu, L. Xie, and M.-P. Tsui, "Physics-motivated features for distinguishing photographic images and computer graphics," in *Proc. ACM Multimedia*, Singapore, Nov. 2005, pp. 239–248.

[18] Z. J. Geradts, J. Bijhold, M. Kieft, K. Kurosawa, K. Kuroki, and N. Saitoh, "Methods for identification of images acquired with digital cameras," in *Proc. SPIE, Enabling Technologies for Law Enforcement and Security*, Feb. 2001, vol. 4232, pp. 505–512.

[19] K. Kurosawa, K. Kuroki, and N. Saitoh, "CCD fingerprint method—identification of a video camera from videotaped images," in *Proc. IEEE Int. Conf. Image Processing*, Kobe, Japan, Oct. 1999, vol. 3, pp. 537–540.

[20] J. Lukas, J. Fridrich, and M. Goljan, "Determining digital image origin using sensor imperfections," in *Proc. SPIE, Image and Video Communications and Processing*, San Jose, CA, Jan. 2005, vol. 5685, pp. 249–260.

[21] M. Kharrazi, H. T. Sencar, and N. Memon, "Blind source camera identification," in *Proc. IEEE Int. Conf. Image Processing*, Singapore, Oct. 2004, vol. 1, pp. 709–712.

[22] S. Bayram, H. T. Sencar, N. Memon, and I. Avcibas, "Source camera identification based on CFA interpolation," in *Proc. IEEE Int. Conf. Image Processing*, Genoa, Italy, Sep. 2005, vol. 3, pp. 69–72.

[23] J. Adams, K. Parulski, and K. Spaulding, "Color processing in digital cameras," *IEEE Micro*, 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," in *Proc. SPIE, Cameras and Systems for Electronic Photography and Sci. Imaging*, Feb. 1995, vol. 2416, pp. 144–151.

[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, and R. T. Gray, "Gradient Based Method for Providing Values for Unknown Pixels in a Digital Image," U.S. Patent 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 5 629 734, May 1997.

[28] A. Swaminathan, M. Wu, and K. J. R. Liu, "Non-intrusive forensic analysis of visual sensors using output images," in *Proc. IEEE Conf. Acoustic, Speech and Signal Processing*, Toulouse, France, May 2006, vol. 5, pp. 401–404.

[29] ——, "Component forensics of digital cameras: a non-intrusive approach," in *Proc. Conf. Information Sciences and Systems*, Princeton, NJ, Mar. 2006, pp. 1194–1199.

[30] C. F. van Loan, *Introduction to Scientific Computing*. Upper Saddle River, NJ: Prentice-Hall, 2000.

[31] T.-F. Wu, C.-J. Lin, and R. C. Weng, "Probability estimates for multiclass classification by pairwise coupling," *J. Mach. Learning Res.*, 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, no. 2, pp. 121–168, 1998.

[33] C.-C. Chang and C.-J. Lin, LIBSVM: A Library for Support Vector Machines 2001, Software available at [Online]. Available: http://www.csie.ntu.edu.tw/cjlin/libsvm.

[34] F. Xiao, J. E. Farell, J. M. DiCarlo, and B. A. Wandell, "Preferred color spaces for white balancing," *SPIE Sensors and Camera Systems for Scientific, Industrial and Digital Photo. Applications IV*, vol. 5017, pp. 342–350, May 2003.

[35] G. D. Finlayson, M. S. Drew, and B. V. Funt, "Diagonal transform suffice for color constancy," in *Proc. IEEE Conf. Computer Vision*, Berlin, Germany, May 1993, pp. 164–171.

[36] Information About Fujifilm Super CCD Cameras [Online]. Available: http://www.fujifilm.com/superccd/.

[37] A. K. Jain, *Fundamentals of Digital Image Processing*. Upper Saddle River, NJ: Prentice-Hall, 2000.

[38] S. Bayram, H. T. Sencar, and N. Memon, "Improvements on source camera-model identification based on CFA interpolation," presented at the Working Group 11.9 Int. Conf. Digital Forensics, Orlando, FL, Jan. 2006.

[39] E. Chang, S. Cheung, and D. Y. Pan, "Color filter array recovery using a threshold-based variable number of gradients," in *Proc. SPIE, Sensors, Cameras, and Applications for Digital Photography*, Mar. 1999, vol. 3650, pp. 36–43.

[40] J. Platt, "Probabilistic outputs for support vector machines and comparison to regularized likelihood methods," in *Advances in Large Margin Classifiers (Neural Information Processing)*. Cambridge, MA: MIT Press, 2000, pp. 61–74.

[41] H. Gou, A. Swaminathan, and M. Wu, "Robust scanner identification based on noise features," presented at the SPIE, Security, Steganography and Watermarking of Multimedia Contents IX, San Jose, CA, Jan. 2007.

[42] Information About Foveon X3 Sensors [Online]. Available: http://www.foveon.com/.
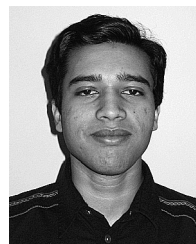
Ashwin Swaminathan (S'05) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology (IIT), Madras, India, in 2003, and is currently pursuing the Ph.D. degree in signal processing and communications at the Department of Electrical and Computer Engineering, University of Maryland, College Park.

He was a Research Intern with Hewlett-Packard Labs (Palo Alto, CA) in 2006. His research interests include multimedia forensics, information security, and authentication.

Mr. Swaminathan's paper on multimedia security was selected as a winner of the Student Paper Contest at the 2005 IEEE International Conference on Acoustic, Speech, and Signal Processing.

Min Wu (S'95–M'01–SM'06) received the B.E. degree (Hons.) in electrical engineering and B.A. degree in economics (Hons.) from Tsinghua University, Beijing, China, in 1996 and the Ph.D. degree in electrical engineering from Princeton University, Princeton, NJ, in 2001.

Since 2001, she has been with the faculty of the Department of Electrical and Computer Engineering and the Institute of Advanced Computer Studies at the University of Maryland, College Park, where she is currently an Associate Professor. Previously, she was with NEC Research Institute and Panasonic Laboratories, Princeton. She coauthored *Multimedia Data Hiding* (Springer-Verlag, 2003) and *Multimedia Fingerprinting Forensics for Traitor Tracing* (EURASIP/Hindawi, 2005), and holds five U.S. patents. Her research interests include information security and forensics, multimedia signal processing, and multimedia communications. She served as a Guest Editor of a 2004 special issue in the *EURASIP Journal on Applied Signal Processing*.

Dr. Wu received the National Science Foundation CAREER Award in 2002, a University of Maryland George Corcoran Education Award in 2003, a Massachusetts Institute of Technology Technology Review's TR100 Young Innovator Award in 2004, and an Office of Naval Research (ONR) Young Investigator Award in 2005. She was a corecipient of the 2004 EURASIP Best Paper Award and the 2005 IEEE Signal Processing Society Best Paper Award. She is an Associate Editor of the IEEE SIGNAL PROCESSING LETTERS.

K. J. Ray Liu (F'03) is Professor and Associate Chair, Graduate Studies and Research, of Electrical and Computer Engineering Department, University of Maryland, College Park.

His research contributions encompass broad aspects of wireless communications and networking, information forensics and security, multimedia communications and signal processing, bioinformatics and biomedical imaging, and signal processing algorithms and architectures. He was the Editor-in-Chief of IEEE SIGNAL PROCESSING MAGAZINE and the founding Editor-in-Chief of the *EURASIP Journal on Applied Signal Processing*.

Dr. Liu is Vice President—Publications and on the Board of Governor of IEEE Signal Processing Society. He is the recipient of many honors and awards including best paper awards from the IEEE Signal Processing Society (twice), IEEE Vehicular Technology Society, and EURASIP; IEEE Signal Processing Society Distinguished Lecturer, EURASIP Meritorious Service Award, and National Science Foundation Young Investigator Award. He also received various teaching and research awards from the University of Maryland, including the Distinguished Scholar–Teacher award, Poole and Kent Company Senior Faculty Teaching Award, and the Invention of the Year award.