Digital Image Forensics via Intrinsic Fingerprints

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

***Abstract*—Digital imaging has experienced tremendous growth in recent decades, and digital camera images have been used in a growing number of applications. With such increasing popularity and the availability of low-cost image editing software, the integrity of digital image content can no longer be taken for granted. This paper introduces a new methodology for the forensic analysis of digital camera images. The proposed method is based on the observation that many processing operations, both inside and outside acquisition devices, leave distinct intrinsic traces on digital images, and these intrinsic fingerprints can be identified and employed to verify the integrity of digital data. The intrinsic fingerprints of the various in-camera processing operations can be estimated through a detailed imaging model and its component analysis. Further processing applied to the camera captured image is modelled as a manipulation filter, for which a blind deconvolution technique is applied to obtain a linear time-invariant approximation and to estimate the intrinsic fingerprints associated with these postcamera operations. The absence of camera-imposed fingerprints from a test image indicates that the test image is not a camera output and is possibly generated by other image production processes. Any change or inconsistencies among the estimated camera-imposed fingerprints, or the presence of new types of fingerprints suggest that the image has undergone some kind of processing after the initial capture, such as tampering or steganographic embedding. Through analysis and extensive experimental studies, this paper demonstrates the effectiveness of the proposed framework for nonintrusive digital image forensics.**

***Index Terms*—Component forensics, image-acquisition forensics, intrinsic fingerprints, nonintrusive image forensics, steganalysis, tampering detection.**

I. Introduction

RECENT decades have witnessed rapid advancements in digital photography. Digital images have been used in a wide variety of applications, from military and reconnaissance to medical diagnosis and consumer photography. With such high popularity and the advent of low-cost and sophisticated image editing software, the integrity of image content can no longer be taken for granted and a number of forensic-related questions arise amidst such extensive use. For example, one can readily as indirectly or directly question ask how an image was acquired? Was it captured using a digital camera, an image scanner, or was it created artificially using image editing software? Has the image undergone any manipulation after capture? Is it authentic or has it been tampered in any way? Does it contain any hidden information or steganographic data? Many of these forensic questions are related to tracing the origin of the digital image to its creation process. Evidence obtained from such forensic analysis would provide useful forensic information to law enforcement, security, and intelligence agencies. Knowledge of image-acquisition techniques can also help answer further forensic questions regarding the nature of additional processing the image has undergone after capture.

In this work, we develop a novel methodology for digital image forensics of color images. We present techniques to identify the inherent traces that are left behind in a digital image when it goes though various processing blocks in the information processing chain. We refer to these traces as the intrinsic fingerprints, and use them to identify the source and establish the authenticity of the digital image. We classify intrinsic fingerprints into two categories, namely in-camera and postcamera fingerprints. Using a detailed imaging model and its component analysis, we estimate the intrinsic fingerprints of the various in-camera processing operations. Further processing applied to camera outputs, if any, are modeled as a filtering operation, and its coefficients are estimated to obtain the postcamera fingerprints. While the absence of in-camera fingerprints suggests that the test image is not a camera output and is possibly generated by other image production processes, any change or inconsistencies among the estimated in-camera fingerprints, or the presence of new postcamera fingerprints indicates that the image has undergone some kind of postcamera processing.

Postcamera processing operations include such manipulations as tampering and steganographic embedding. Recently, there have been an increasing number of software tools for manipulating multimedia data. While these programs enable quality enhancement, they also facilitate easy editing and tampering of data. Therefore, establishing the integrity of digital content has become particularly important when images are used as critical evidence in journalism and surveillance applications. Data authentication techniques, such as semifragile watermarking [1], [2] and robust hashing [3], require the watermark/signature or more generally extrinsic fingerprints, to be inserted at the time of creation of multimedia data. The presence or absence of the watermark in interpolated images captured by the camera can be employed to establish the authenticity of digital color images [4]. However, such techniques impose several restrictions on its applicability as many digital cameras and video recorders in the market still do not have the capabilities to add a watermark or a hash at the time of image creation. Hence, there is a strong motivation as a part of the emerging field of image forensics to devise nonintrusive methods to distinguish authentic images from manipulated ones. As we shall show later in this paper, the proposed techniques facilitate tampering forensics by determining whether there has been any additional

Manuscript received May 31, 2007; revised September 11, 2007. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Hany Farid.

The authors are with the Department of Electrical and Computer Engineering and the Institute of Advanced Computing Studies, University of Maryland, College Park, MD 20742 USA (e-mail: ashwins@eng.umd.edu; minwu@eng.umd.edu; kjrliu@eng.umd.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2007.916010

1556-6013/$25.00 © 2008 IEEE

editing and processing applied to an image after it leaves the camera.

Watermarking and steganographic embedding may also be modeled as postprocessing operations applied to camera outputs, and the estimated postcamera fingerprints can be utilized to identify them. Steganography is the art of secret communication where the hidden information is transmitted by embedding it on to the host multimedia. Over the past few years, there have been a number of steganographic embedding algorithms using digital images as hosts for covert communication [5], [6], [7]–[9]. In the same period, several steganalysis methods have been proposed to identify the presence of hidden data in multimedia. While embedding specific steganalysis [10] target-specific embedding algorithms, universal steganalysis [11], [12] is designed to identify more than one type of steganography. With an increasing number of steganographic embedding algorithms, there is a strong need for robust universal methods for blind steganalysis. As can been seen from our results, the proposed intrinsic fingerprinting techniques facilitate blind steganalysis by distinguishing authentic camera outputs from images with hidden content.

The paper is organized as follows. After reviewing the related works in Section II, we discuss the image-acquisition model and present techniques to estimate the in-camera fingerprints in Section III. In Section IV, we introduce the problem formulation and propose a new forensic framework to estimate the postcamera fingerprints. We show that the proposed method is universal and can distinguish between genuine photographs and its manipulated versions. Detailed simulation results and elaborate case studies are presented in Sections V and VI, and the final conclusions are drawn in Section VII.

II. Related Prior Work

Recently, there has been growing research on nonintrusive forensics devoted to the security and protection of multimedia information. Each technique targets addressing different aspects related to verifying the authenticity of digital data. Related prior work falls into three main categories. In the first category, there have been works on source authentication. Higher order statistical models using wavelet transform coefficients [13] and physics-motivated features based on geometry and cartoon features [14] have been proposed for classifying photographs and photorealistic computer graphics.

In the second group, there have been works in the tampering detection literature trying to define the properties of a manipulated image in terms of the distortions it goes through, and using such analysis to present methods for detecting manipulated images. In doing so, some works assume that creating a tampered image involves a series of processing operations, which might include resampling [15]; JPEG compression [16], [17]; Gamma correction [18]; and chromatic aberration [19]. Based on this observation, they propose identifying such manipulations by extracting certain salient features that would help distinguish such tampering from authentic data. For instance, when the image is upsampled, some of the pixel values are directly obtained from the smaller version of the image, and the remaining pixels are interpolated and, thus, highly correlated with its neighbors. Thus, postprocessing operations, such as resampling, can be identified by studying the induced correlations [15]. JPEG compression has been considered as quantization in the discrete cosine transform (DCT) domain and statistical analysis based on binning techniques has been used to estimate the quantization matrices [16], [20]. Image manipulations, such as contrast changes, Gamma correction, and other image nonlinearities have been modelled and higher order statistics, such as the bispectrum, have been used to identify them [21], [22]. Inconsistencies in noise patterns [21], JPEG compression [23], or lighting [24], and alternations in correlations induced by color interpolation [25] caused while creating a tampered picture have been used to identify inauthentic images.

Although these methods can be employed to identify the type and the parameters of the postprocessing operation, it would require an exhaustive search over all kinds of postprocessing operations to detect tampering. The presence of pattern noise in camera-captured images and its absence in tampered images have been used to detect forgeries [26]. Classifier-based approaches to detect image tampering were proposed in [27] and [28], where features based on the analysis of variance approaches [27] and higher order wavelet statistics [28] have been used to detect image manipulations. However, these methods require samples of tampered images for classification to distinguish manipulated images from genuine ones. Further, these methods may not be able to efficiently identify other kinds of manipulations that are not modelled or considered directly. By defining the properties of an authentic image via intrinsic fingerprints, our proposed methods provide better scalability and can help identify previously unseen distortions.

In the third group of prior art, there have been works on steganalysis to identify the presence of hidden information in multimedia data. These works can be broadly classified into two classes, namely: 1) embedding specific and 2) universal. In the class of embedding-specific steganalysis, there have been algorithms to identify different types of least-significant bit (LSB) embedding [10], [29], [30]. Statistics-based approaches for universal blind staganalysis have been introduced in [11] and [12], where features from wavelet statistics [11] or image-quality measures [12] are used to build a classifier to distinguish stegodata from cover data. As shall be seen from our results later in this paper, our proposed forensic methodology provides a combined framework for authenticating digital camera outputs and distinguishing them from scanned, computer-generated, tampered, and stegodata.

III. Estimating Intrinsic Fingerprints of In-Camera Processing

When a real-world scene is captured using a digital camera, the information about the scene passes through the various camera components before the final digital image is produced. Each component in the information processing chain modifies the input via a particular algorithm using a specific set of parameters, and leaves some intrinsic fingerprint traces on the output. In this section, we begin reviewing imaging models of digital cameras to examine various components in its information processing chain. We then discuss techniques to nonintrusively estimate the component parameters to obtain

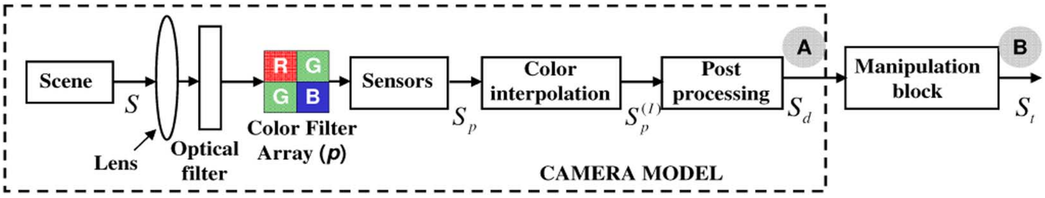


Fig. 1. System model.

the intrinsic fingerprints of the in-camera processing. Later in Section IV, we use these intrinsic in-camera fingerprints to look for any new fingerprints left behind on the final digital image through additional postcamera processing operations.

A. *Image-Acquisition Model*

Fig. 1 shows the image-acquisition model in digital cameras. The light from the scene passes through the lens and the optical filters and is finally recorded by the color sensors. Most digital cameras use a color filter array (CFA) to sample the real-world scene. The 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. To facilitate discussions, let S be the real-world scene to be captured by the camera and let p be the CFA matrix. S is a 3-D array of pixel values of size $H \times W \times C$, where H and W denote the height and the width of the image, and $C = 3$ is the number of color components (red, green, and blue). The CFA sampling converts the real-world scene S into S_p satisfying

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

After the data obtained from the CFA are recorded, the intermediate pixel values [corresponding to the points where $S_p(x,y,c) = 0$ in (1)] are interpolated using the neighboring pixel values to obtain $S_p^{(I)}$. After interpolation, the three images corresponding to the red, green, and blue components go though a postprocessing stage. In this stage, depending on the camera make and model, the images may undergo different processing operations [31], [32], which might include white balancing, color correction, gamma correction, lens vignetting correction, lens distortion removal, denoising, etc. Finally, the image may be JPEG compressed to reduce storage space to produce the output image S_d. For our work, we model all such postinterpolation processing as a combined postprocessing block as shown in Fig. 1.

B. *Estimating Camera Component Parameters*

As can be seen from Fig. 1, the data about the real-world scene pass through the various components of the information processing chain before the final digital image is created in point A. Each camera component, such as a CFA and color interpolation, employs a particular set of algorithms with an appropriate set of parameters to modify the input scene. In these processing stages, each camera uses a different algorithm (that may be proprietary to the camera manufacturer, brand, or model) and leaves intrinsic fingerprint traces on the output data. In our recent work [33], we presented methods to estimate these in-camera fingerprints from outputs corresponding to point A in the information processing chain shown in Fig. 1. We provide a brief overview of these techniques below, and later in Section IV, we build upon these methods and introduce a novel approach to estimate the postcamera fingerprints of manipulated camera outputs corresponding to point B in the processing chain.

The CFA pattern and the color interpolation coefficients can be jointly estimated from the output image [33]. A search space P for the CFA patterns is first established based on common practice in digital camera design. For every CFA pattern p in the search space P, the interpolation coefficients are computed separately in different types of texture regions by fitting linear models. Specifically, the image is divided into three types of regions based on the gradient features in a local neighborhood. Denoting $I_{x,y} = S_d(x,y,p(x,y))$, the horizontal and vertical gradients at the location (x,y) are found by using

$$\mathcal{H}_{x,y} = |I_{x,y-2} + I_{x,y+2} - 2I_{x,y}|, \tag{2}$$

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

The image pixel at location (x,y) is classified into one of the three categories: Region $\Re_1$ contains those parts of the image with a significant horizontal gradient for which $(\mathcal{H}_{x,y} - \mathcal{V}_{x,y}) > \mathcal{T}$, where $\mathcal{T}$ is a suitably chosen threshold; region $\Re_2$ contains those parts of the image with a significant vertical gradient $(\mathcal{V}_{x,y} - \mathcal{H}_{x,y}) > \mathcal{T}$; and region $\Re_3$ includes the remaining parts of the image which primarily contains the smooth regions. Using the final camera output S_d, a set of linear equations for all the pixels in each region $\Re_i (i = 1,2,3)$ is obtained and solved to obtain the interpolation coefficients $\alpha_{\Re_i}$. Once these coefficients are estimated, they are used to reinterpolate the image and find the interpolation error. The CFA pattern that gives the lowest error gives the estimate of the CFA pattern. Further, the estimates are shown to be robust to moderate levels of postprocessing operations, such as JPEG compression, and white balancing done inside the cameras [33].

IV. Estimating Intrinsic Fingerprints of Postcamera Manipulations

In this section, we build upon component forensic analysis presented in the previous section, and propose techniques to estimate the intrinsic fingerprints of postcamera manipulations. Given a test image S_t, we introduce a nonintrusive forensic methodology to identify if it has undergone any further processing after it has been captured using a digital camera. In this work, we mainly focus on digital color images that constitute a bulk of camera-captured images. We first assume that S_t is a manipulated camera output corresponding to the point B in Fig. 1, and is obtained by processing the actual camera output S_d (point A in Fig. 1) using the manipulation block. We then represent the postcamera processing applied on S_d as a combination of linear and nonlinear operations, and approximate them with a linear shift-invariant filter. The coefficients of this manipulation filter, estimated using blind deconvolution, serve as our postcamera fingerprints to answer a number of forensic questions related to the origin and the authenticity of digital images. In the following subsections, we describe the estimation algorithm in detail.

A. Computing Inverse Manipulation Filter Coefficients by Constrained Optimization

Let S_t denote the test image, and let S_{te} represent the estimate of the camera output obtained by passing the given test image through the inverse manipulation filter u, i.e.,

$$S_{te}(x,y,c) = \sum_{m,n} u(m,n,c) \times S_t(x-m, y-n, c), \quad \text{for } 1 \le c \le 3. \tag{4}$$

Here, we assume that $u(\cdot,\cdot,\cdot)$ is of dimension $N_u \times N_u \times 3$, and operates independently on each color component. The coefficients of the inverse manipulation filter u are estimated by solving an optimization problem that minimizes the camera model fitting error $E(u)$ given by

$$E(u) = \sum_{c=1}^{3} \sum_{x,y} \left(\hat{S}_{te}(x,y,c) - \sum_{m,n} u(m,n,c) S_t(x-m, y-n, c) \right)^2 \tag{5}$$

where $\hat{S}_{te}$ denotes the image formed from S_{te} by imposing the constraints that pixels from a camera output image should satisfy due to CFA-based color interpolation

$$\hat{S}_{te}(x,y,c) = \begin{cases} \sum_{m,n} \alpha_{\Re_i}(m,n,c) S_{te}(x-m,y-n,c) \\ \qquad \forall \{x,y\} \in \Re_i, \text{ and } 1 \le c \le 3, \\ S_{te}(x,y,c), \quad \text{otherwise.} \end{cases} \tag{6}$$

In these camera constraints, $\alpha_{\Re_i}$ denotes the estimates of the color interpolation coefficients and are derived from the image S_{te} using the component forensics techniques presented in Section III-B. In our work, we assume that $\sum_{m,n} u(m,n,c) = 1$ for $c = 1,2,3$ to ensure that the original image and its manipulated version have similar brightness levels. Incorporating this gain constraint into the minimization problem, we solve for u by minimizing a modified cost function $J(u)$, given by

$$J(u) = \sum_{x,y,c} \left(\hat{S}_{te}(x,y,c) - \sum_{m,n} u(m,n,c) S_t(x-m,y-n,c) \right)^2 + \eta \sum_{c=1}^{3} \left(\sum_{m,n} u(m,n,c) - 1 \right)^2 \tag{7}$$

where the value of η is chosen to adjust the weights of the relative individual costs.

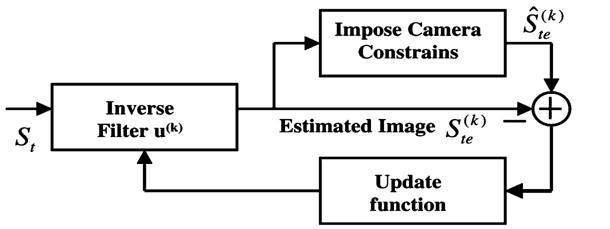


Fig. 2. Recursive algorithm to estimate the coefficients of the manipulation filter.

The filter coefficients can be directly estimated in the pixel domain through a recursive procedure illustrated in Fig. 2. We start the iteration by setting $u^{(0)}$ to be a delta function; this corresponds to direct camera outputs. In the kth iteration, we obtain an estimate of the camera output $S_{te}^{(k)}$ by passing the test image S_t through the estimate of the inverse blurring filter $u^{(k)}(\cdot,\cdot,\cdot)$. We then impose camera constraints given by (6) to obtain $\hat{S}_{te}^{(k)}$ and find the camera model fitting error. The inverse filter coefficients are then updated [34] by

$$u^{(k+1)} = u^{(k)} + t_k d_k \tag{8}$$

where

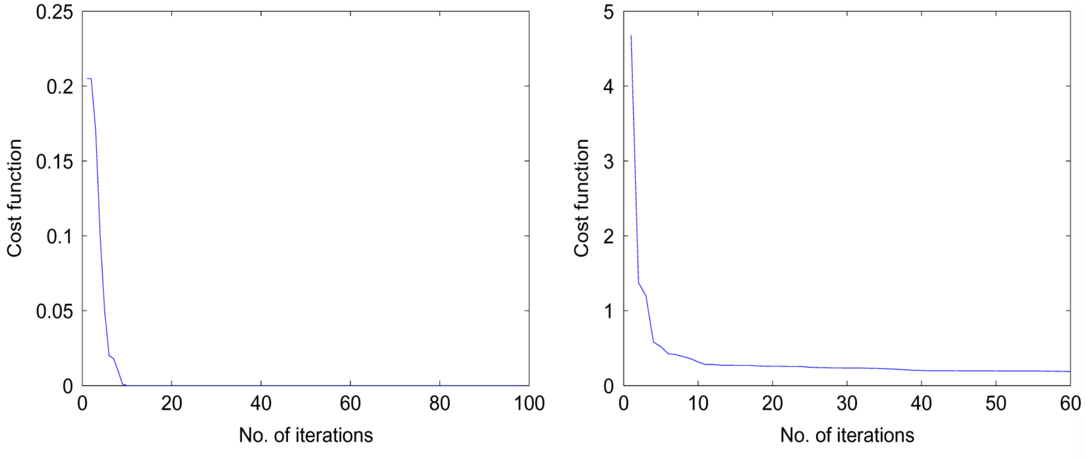
$$d_k = \begin{cases} -\nabla J\left(u^{(k)}\right), & \text{if k} = 0, \\ -\nabla J\left(u^{(k)}\right) + \lambda_{k-1} d_{k-1}, & \text{otherwise,} \end{cases} \tag{9}$$

$$\lambda_{k-1} = \frac{< \nabla J\left(u^{(k)}\right) - \nabla J\left(u^{(k-1)}\right), \nabla J\left(u^{(k)}\right) >}{\|\nabla J\left(u^{(k-1)}\right)\|^2} \tag{10}$$

and the step sizes t_k are chosen as the one that minimizes $J(u^{(k)} + t_k d_k) \le J(u^{(k)} + t d_k)$ for all t. The recursive procedure is repeated for a finite number of iterations or until convergence. In the Appendix, we show that the optimization problem is convex and converges to a unique solution for all images whose interpolation parameters $\alpha_{\Re_i}$ can be estimated accurately.

We test the blind deconvolution method for a sample direct camera output along with its filtered versions. Fig. 3(a) and (b) shows the variation of the modified cost function J given by (7) as a function of the number of iterations for a sample unmanipulated image and an image filtered with a 5 × 5 averaging filter, respectively. We observe that the cost function converges in ten iterations in both cases. The final estimated inverse filter coefficients $u(\cdot,\cdot,2)$ for the green color channel for the two cases are shown in Fig. 4(a) and (b), respectively. While the estimated coefficients from the unmanipulated camera output in Fig. 4(a) are very close to an identity transform (corresponding to no postcamera manipulations), the corresponding manipulation coefficients derived from the average filtered image, as presented in Fig. 4(b), are similar to the 5 × 5 kernel approximation of the inverse of the 5 × 5 averaging filter.

The performance of the blind deconvolution algorithm for tampering detection is, to a great extent, tied with the choice of the kernel size. In an ideal scenario, a finite-size averaging filter



(a) (b)

Fig. 3. Convergence of the cost function for an (a) unmanipulated image and (b) a manipulated image filtered with a 5 × 5 averaging filter.

–0.077	0	–0.077	0	–0.077
0	–0.077	0	–0.077	0
–0.077	0	1.923	0	–0.077
0	–0.077	0	–0.077	0
–0.077	0	–0.077	0	–0.077

(a)

0.129	0.041	–0.051	0.058	0.134
0.056	0.156	–0.273	0.159	0.022
–0.044	–0.281	0.764	–0.274	–0.032
0.026	0.156	–0.276	0.155	0.059
0.139	0.061	–0.049	0.034	0.125

(b)

Fig. 4. Estimated inverse manipulation filter coefficients for an (a) unmanipulated image and (b) a manipulated image filtered with a 5 × 5 averaging filter. The inverse filter kernel size is set to 5 × 5.

in the pixel domain would require an infinite length kernel for its inverse. Although a larger kernel gives enhanced performance improvements, it requires more iterations for convergence. In the next subsection, we present a solution to directly estimate the filter coefficients in the frequency domain.

B. *Estimating Manipulation Filter Coefficients by Iterative Constraint Enforcement*

The recursive algorithm described in Fig. 2 can be solved in the frequency domain to directly obtain the manipulation filter coefficients by iteratively applying known constraints to the input image [36]. A schematic diagram of the iterative constraint enforcement algorithm is shown in Fig. 5. The test image S_t is used to initialize the iterative process. In each iteration, the estimated camera output g and the estimated filter coefficients h are updated by repeatedly applying known constraints on the image and the filter in the pixel domain and the Fourier domain. In the kth iteration, the pixel domain constraints on the image g_k consist of

1) Real-valued constraints that enforce the image pixel values to be real.
2) Boundedness constraints restricting the image pixel values to the range $[0, 255]$.
3) Camera constraints of the CFA-based color interpolation given by

$$\hat{g}_k(x,y,c) = \begin{cases} \sum_{m,n} \alpha_{\Re_i}(m,n,c) g_k(x-m, y-n, c), \\ \qquad \forall \{x,y\} \in \Re_i, \text{ and } 1 \le c \le 3 \\ g_k(x,y,c), \quad \text{otherwise} \end{cases} \tag{11}$$

where $\alpha_{\Re_i}$ denotes the estimates of the color interpolation coefficients derived from the image g_k using the component forensics techniques presented in Section III-B. After the image $\hat{g}_k$ is obtained, it is transformed by the discrete Fourier transform (DFT) to give $\hat{G}_k$. The frequency response H_k of the estimated manipulation filter in the kth iteration is obtained by using the technique described in [35] with

$$H_k = \frac{\mathcal{F}(S_t)\hat{G}_k^*}{|\hat{G}_k|^2 + \frac{\beta_1}{|H_{k-1}|^2}} \tag{12}$$

where β_1 is an appropriately chosen constant, $\mathcal{F}(S_t)$ denotes the Fourier transform of the test image S_t, and $\hat{G}_k^*$ represents the complex conjugate of $\hat{G}_k$. The value of H_0 for the first iteration is initialized as $H_0 = \mathcal{F}(S_t)/\hat{G}_0$. The estimated filter response H_k is then inverse Fourier transformed to give h_k. We further impose filter constraints on h_k and obtain $\hat{h}_k$ to be the real part of h_k. The value of G_{k+1} for the $(k+1)^{st}$ iteration is obtained as a function of its two available estimates: 1) previous value G_k and 2) the estimate obtained by enforcing the Fourier domain constraint $(FS_t/\hat{H}_k)$, where $FS_t = \mathcal{F}(S_t)$ and $\hat{H}_k = \mathcal{F}(\hat{h}_k)$. Both of these estimates have unique properties: G_k has a nonnegative inverse transform that satisfies the image domain constraints, and $(FS_t/\hat{H}_k)$ satisfies the Fourier domain constraints. In our work, we average these two estimates separately in every iteration for each spatial frequency value and color to obtain the new estimate for G_{k+1} as described in (13), shown at the bottom of the next page, where γ and β_2 are appropriately chosen constants [36]. The value of γ represents the noise resilience of the system, and β_2 is chosen to lie in the range $[0, 1]$ to indicate the

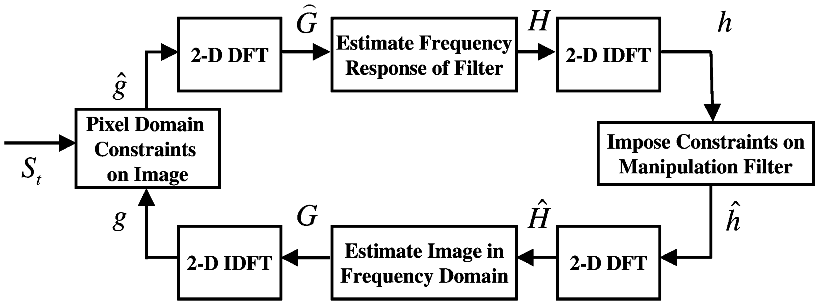

2-D DFT
$\hat{G}$
Estimate Frequency Response of Filter
H
2-D IDFT
h
$\hat{g}$
Pixel Domain Constraints on Image
S_t
Impose Constraints on Manipulation Filter
g
G
$\hat{H}$
$\hat{h}$
2-D IDFT
Estimate Image in Frequency Domain
2-D DFT

Fig. 5. Schematic diagram of the iterative constraint enforcement algorithm.

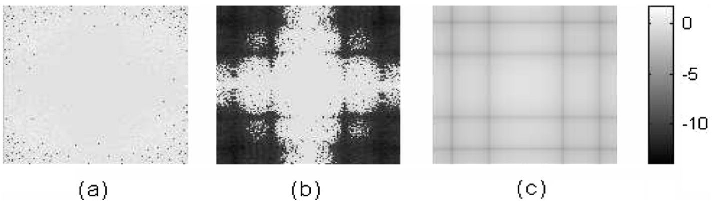


Fig. 6. Frequency response of the manipulation filter for (a) a simulated unmanipulated camera output and (b) an image lowpass filtered with a 5×5 averaging filter. (c) Actual manipulation filter coefficients of the 5×5 averaging filter shown alongside for comparison. The magnitude of the frequency response is shown in the $\log_{10}$ scale.

relative significance of the two terms in update equation [36]. In our experiments, we set $\gamma = 10^{-5}$ and $\beta_1 = \beta_2 = 0.3$. Finally, G_{k+1} is inverse Fourier transform to give g_{k+1}, the pixel domain estimate of the camera output image, and the system proceeds to the next iteration. This process is repeated for a finite number of iterations and the frequency response of the estimated manipulation filter parameters H is found to obtain the intrinsic fingerprints of postcamera manipulations. The deviation of the estimated manipulation filter parameters from an identity transform indicates that the test image has been manipulated after capture by the camera.

C. Performance Studies on Detecting Manipulations With Synthetic Data

We use synthetic data constructed from 100 representative images to study the performance of the blind deconvolution techniques for tampering detection [37]. These 100 images are first downscaled by a factor of 2×2 to remove the effects of previously applied filtering and interpolation operations, sampled on the Bayer filter [31], [32] array and then interpolated using six different interpolation algorithms to reproduce the scene capture process in cameras. For our simulations, we consider six different color interpolation methods: 1) bilinear, 2) bicubic, 3) smooth hue, 4) median filter, 5) gradient based, and 6) adaptive color plane. Details about these interpolation algorithms can be found in [31]. These 600 images that satisfy the camera model form our unmanipulated set. Processed versions are then obtained by applying average filtering to these 600 images with different filter orders from 3 to 11.

We run the proposed blind deconvolution methods on all of the images and compute the coefficients of the manipulation filter in each case using the iterative constraint enforcement algorithm. In Fig. 6(a), we show the estimated Fourier transform for a simulated unmanipulated camera output. We notice that it is almost a constant flat spectrum, representing an identity transform. The corresponding estimated frequency response for a 5×5 average filtered image is shown in Fig. 6(b), and the actual coefficients are shown in Fig. 6(c) for comparison. The similarity among the estimated and the actual coefficients justifies the performance of the blind deconvolution algorithms.

A closer look at the frequency response of the manipulation filter for an unmanipulated camera output, shown in Fig. 6(a), suggests minor deviations from an ideal flat spectrum. These deviations are attributed to the various postinterpolation processing that takes place inside the cameras, such as compression, denoising, and white balancing. To compensate for these minor deviations, we use the spectral response $H_{\rm ref}$, obtained using the blind deconvolution algorithm, from an authentic camera output as reference. Given the test input S_t, we find the frequency-domain coefficients of the manipulation filter H_t and compare it with $H_{\rm ref}$ to measure the similarity among the coefficients. More specifically, we first find $\Theta_t = \log_{10}(|H_t|)$

$$G_{k+1} = \begin{cases} G_k, & \text{if} |FS_t| < \gamma, \\ (1-\beta_2)G_k + \beta_2 \frac{FS_t}{\hat{H}_k}, & \text{if} |FS_t| \le |\hat{H}_k| \text{and } |FS_t| \ge \gamma, \\ \left(\frac{(1-\beta_2)}{G_k} + \frac{\beta_2 \hat{H}_k}{FS_t} \right)^{-1}, & \text{if} |FS_t| > |\hat{H}_k| \text{and} |FS_t| \ge \gamma. \end{cases} \tag{13}$$

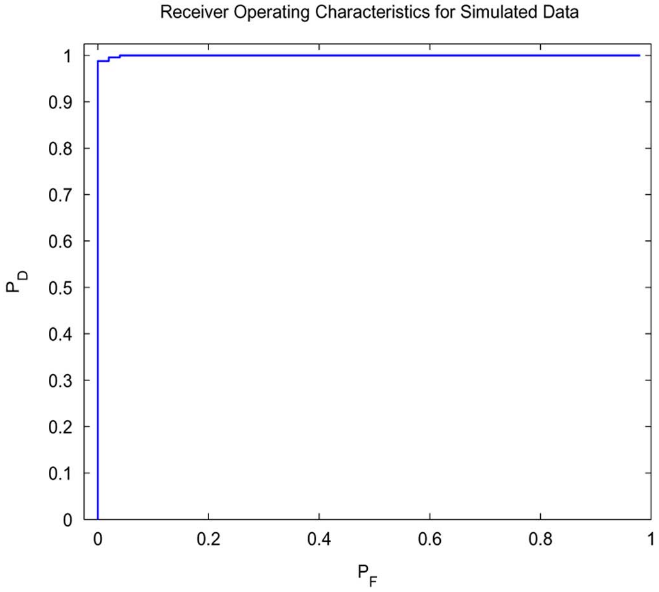


Fig. 7. Receiver operating characteristics for distinguishing between simulated camera outputs and their filtered versions.

to obtain the logarithm of the magnitude of the frequency response, and compute the similarity between the coefficients of the test input and the reference image using the similarity score defined as

$$s(\Theta_t, \Theta_{\mathrm{ref}}) = \sum_{m,n} (\Theta_t(m,n) - \mu_t) \times (\Theta_{\mathrm{ref}}(m,n) - \mu_{\mathrm{ref}}) \quad (14)$$

where μ_t denotes the mean of the Θ_t, and μ_{ref} represents the mean of the Θ_{ref}. The test input is then classified as unmanipulated if the similarity to the reference pattern is greater than a suitably chosen threshold. On the other hand, if the input image has undergone tampering or steganographic embedding operations, the estimated manipulation filter coefficients would include the effects of both the postcamera manipulation operations along with postinterpolation processing inside the camera. In this case, the manipulation filter coefficients would be less similar to the reference pattern, and the similarity score would be lower than the chosen threshold.

We examine the performance of the threshold based classifier in terms of the receiver operating characteristics (ROC) [37]. For each original image, we compute the frequency response of the equivalent manipulation filter and measure its similarity with the reference filter pattern. The fraction of original images with a similarity score lower than a threshold τ is found to give the false alarm probability P_F. Similarly, we record the fraction of manipulated images (filtered in this case) with a similarity score that is less than τ to give the probability of correct decision P_D. We repeat this process for different decision thresholds τ, and arrive at the ROC as shown in Fig. 7. We observe from the figure that the proposed scheme attains a $P_D \approx 1$ for $P_F = 0$. This suggests that the proposed scheme can effectively distinguish between direct camera outputs and its filtered versions.

TABLE I
CAMERA MODELS USED IN EXPERIMENTS

No.	Camera Model	No.	Camera Model
1	Canon Powershot A75	6	Canon EOS Digital Rebel
2	Canon Powershot S410	7	Nikon E4300
3	Canon Powershot G6	8	Fujifilm Finepix S3000
4	Canon Powershot S400	9	Sony Cybershot DSC P72
5	Canon Powershot S1 IS		

V. DETECTING TAMPERING ON CAMERA-CAPTURED IMAGES

Forensic evidence obtained by analyzing the coefficients of the manipulation filter provides clues about possible image tampering. Most often, creating a realistic tampered image involves a series of postcamera processing operations, such as filtering, compression, resampling, contrast change, and others, that may be applied globally to the entire image or locally to different regions of the image. These processing operations leave distinct traces in the final picture and can be detected using the threshold-based classifier by comparing the estimated manipulation filter coefficients with the reference pattern. In this section, we study the performance of the proposed techniques for detecting different types of global image manipulations with real camera data. The forensic methodologies discussed in this section can be extended to detect local tampering by applying the techniques on a block-by-block basis.

A. Simulation Setup

A total of nine camera models as shown in Table I is used in our experiments. For each of the nine camera models, we have collected about 100 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 the default values in each camera. The default camera settings (including image size, color correction, auto white balancing, and JPEG compression) are used in image acquisition. From each image, we randomly crop a 512 × 512 portion and use it for subsequent analysis. Thus, our camera image database consists of a total of 900 different 512 × 512 pictures. These images were then processed to generate 21 tampered versions per image to obtain 18 900 manipulated images, and the 21 manipulation settings are listed in Table II.

B. Classification Methodology and Simulation Results

We study the discriminative capabilities of our proposed schemes in terms of the ROC of the hypothesis testing problem with the following two hypotheses:

1) Υ_0: image is a direct camera output;
2) Υ_1: image is not a direct camera output and is possibly manipulated in some way.

For each image, we compute the frequency-domain coefficients of the estimated manipulation filter and determine its similarity with the chosen reference pattern. Images with a similarity score that are greater than a threshold are classified as authentic.

To choose the reference pattern, we randomly select a set of N_t training images along with its manipulated versions in the training stage. Using each N_t image, we compute the inclass and outclass similarity scores. More specifically, given the ith image $(1 \le i \le N_t)$, we calculate the inclass similarity scores by com-

TABLE II
TAMPERING OPERATIONS INCLUDED IN THE EXPERIMENTS

Manipulation Operation	Parameters of the Operation	Number of Images
Spatial Averaging	Filter orders 3-11 in steps of 2	5
Median Filtering	Filter orders {3, 5, 7}	3
Rotation	Degrees {5, 10, 15, 20}	4
Resampling	Scale factors {0.5, 0.7, 0.85, 1.15, 1.3, 1.5}	6
Additive Noise	PSNR 5dB and 10 dB	2
Histogram Equalization		1
Total		21

paring the manipulation filter estimated from the ith image and the estimates obtained from the remaining $(N_t - 1)$ images using (14). The outclass scores are then found by quantifying the similarity among the manipulation filter of the ith image and the filter coefficients derived from the remaining tampered images. Using a threshold τ, the fraction of direct camera outputs with a similarity score lower than τ is computed to give the false alarm probability $P_F = \Pr(\Upsilon_1 \mid \Upsilon_0)$, and the fraction of manipulated images with a similarity score of less than τ is found to give the probability of correct decision $P_D = \Pr(\Upsilon_1 \mid \Upsilon_1)$. We repeat this process for different decision thresholds τ to arrive at the ROC, and compute the area under the curve. These steps are performed separately with each N_t image in the training stage, and the manipulation filter coefficients that give the maximum area under the ROC curve are chosen as the reference pattern. After choosing the reference pattern in the training stage, we compute the inclass and outclass similarity scores by comparing the chosen reference pattern with the filter coefficients obtained from the remaining camera outputs and their corresponding tampered versions, respectively, in our database in the testing stage. The corresponding ROC curves are obtained through this process.

1) Testing With Images From Canon Powershot A75: We test the performance of the proposed techniques using the 100 images from Canon Powershot A75. We choose this camera for two reasons: 1) based on our experimental studies, we observe that linear shift-invariant model for the color interpolation coefficients fits well with the cameras' interpolation in each type of region and gives a very low fitting error and 2) we observe that this Canon camera uses the same JPEG quantization table for all images that it captures, invariant of the input scene. Therefore, all images from the camera undergo the same kind of postprocessing operations after color interpolation (refer to Fig. 1).

For our analysis with images from Canon Powershot A75, we use a randomly chosen set of 50 images for training, and test on the remaining 50 images along with the corresponding 50×21 tampered images. Fig. 8 shows the performance of the threshold-based detector averaged over 100 iterations. At a relatively low P_F around 10%, the probability of correct detection is about 80%–95% for most types of manipulations tested. Here, the results are based on a two-class classification problem, wherein the first class includes the direct camera outputs and the second class consists of camera outputs that have undergone a specific type of manipulation.

2) Testing With Diverse Inputs From Multiple Cameras: We now examine the performance of the proposed techniques under diverse input conditions. More specifically, we use all 900 direct camera output images for the untampered dataset. These images were captured under the default camera settings and may have undergone different kinds of in-camera postprocessing operations, such as JPEG compression after color interpolation.

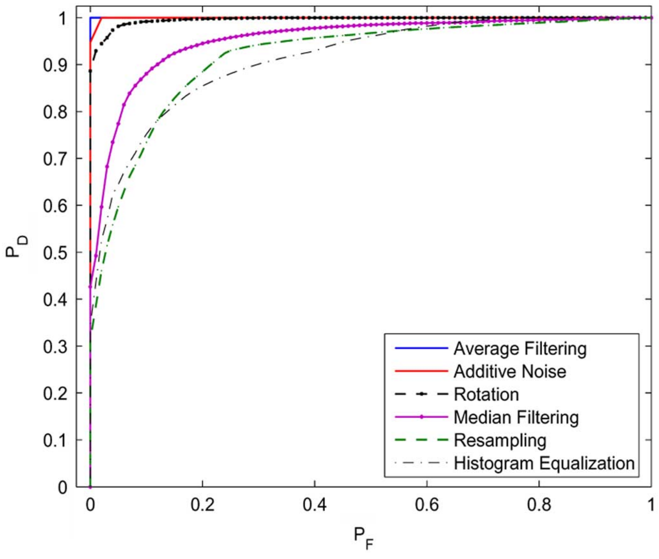


Fig. 8. Receiver operating characteristics for tampering detection for images from Canon Powershot A75 when 50 images are used in training and the remaining 50 images are used in testing.

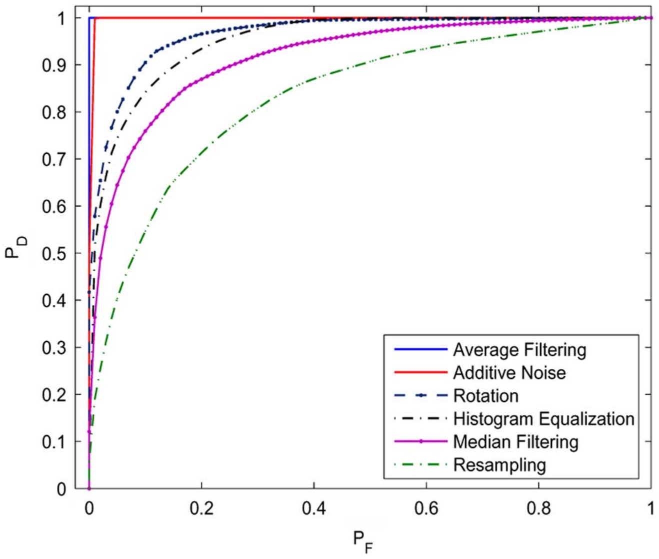


Fig. 9. Receiver operating characteristics for tampering detection when tested with all images in the database with 200 images being used in training.

Fig. 9 shows the ROC curve for detecting each manipulation. Here, we use a randomly chosen set of 200 images to train the classifier and test with the remaining 700 images; the experiments are repeated more than 100 times to obtain an average ROC curve. In this case, we observe that for P_F close to 10%,

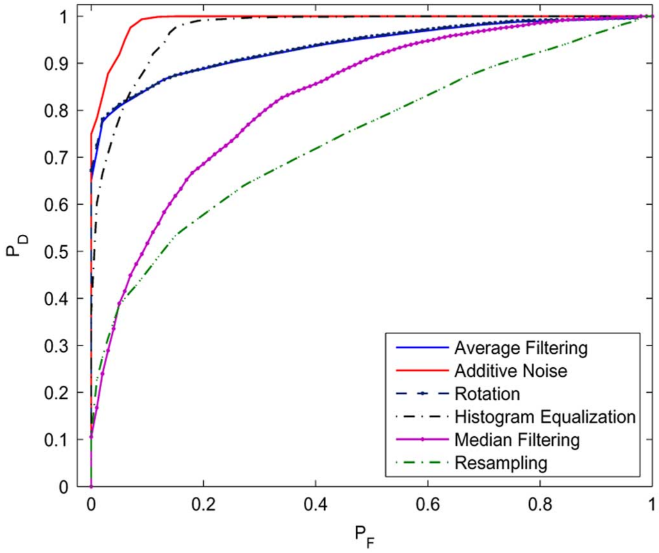


Fig. 10. Receiver operating characteristics for tampering detection when images from the Canon Powershot A75 are used in training and images from Sony Cybershot DSC P72 are used in testing.

the probability of correct detection is close to 100% for such manipulations as spatial averaging and additive noise, and around 70%–80% for median filtering, histogram equalization, and rotation. These results are better than other works in the literature that are applicable to blind tampering detection [25], [28].

Comparing the results in Fig. 9 with the results of the Canon Powershot A75 in Fig. 8, we notice around a 5%–10% performance drop in detection accuracy for the same false positive rate. This reduction in performance can be attributed to the different types of postprocessing operations performed after color interpolation in various camera brands and models. In our future work, we plan to estimate the parameters of such postinterpolation operations as JPEG compression [16] and white balancing, and include them in the system model to bridge the performance gap.

3) Training and Testing Using Inputs From Different Cameras: The proposed techniques are nonintrusive and do not require that the actual camera make/model be used in the training set. To demonstrate this aspect, we test the performance of the proposed techniques using 100 images from Canon Powershot A75 and 100 images from Sony Cybershot DSC P72. We randomly choose 50 out of 100 Canon Powershot A75 images and use them for training to identify the reference pattern; the 100 images from Sony Cybershot DSC P72 are used in testing. The performance results, averaged over 100 iterations, are shown in Fig. 10. The figure shows that the performance is good for most manipulations and for P_F around 10%, the probability of correct detection is close to 80%–90%. This result is comparable to the plots in Figs. 8 and 9. The drop in performance for some manipulations, such as resampling, can be attributed to the absence of the original camera make/model in training.

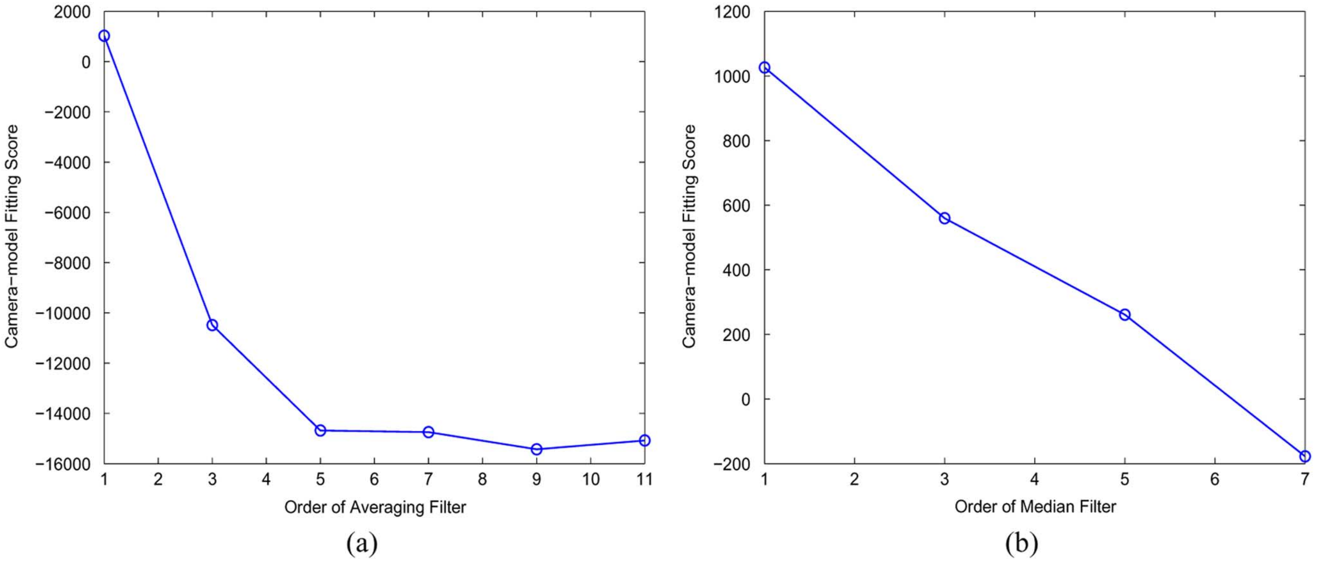
C. Tampering Forensics Using the Estimated Manipulation Filter Coefficients

The estimated filter coefficients can also be employed to quantify the likelihood and degree of tampering, and to identify the type and parameters of the tampering operation. In this subsection, we show that the similarity score can be used to define a camera-model fitting score to evaluate the amount of tampering that the test image has undergone. For our experiments, we first choose six good reference patterns that give the highest area under the ROC curve. The camera-model fitting score for the test image is then defined as the median of the similarity scores obtained by comparing the estimated coefficients of the test image with the ones obtained from each of the six reference patterns. The higher the fitting score is, the greater the likelihood that the test image is for a direct camera output without further processing.

We examine the variation of the camera-model fitting score as a function of the degree of tampering for all the manipulations listed in Table II. Fig. 11(a) and (b) shows the camera-model fitting score as a function of the filter order for spatial averaging and median filtering, respectively. In both cases, we observe that the fitting score reduces as the filter order increases and as the degree of tampering increases. Further, the score is less than -1000 for all average filtered images. This low value is because of the distinct nulls in the frequency spectrum of the manipulated filter, estimated from filtered images, making it very different from the flat reference pattern.

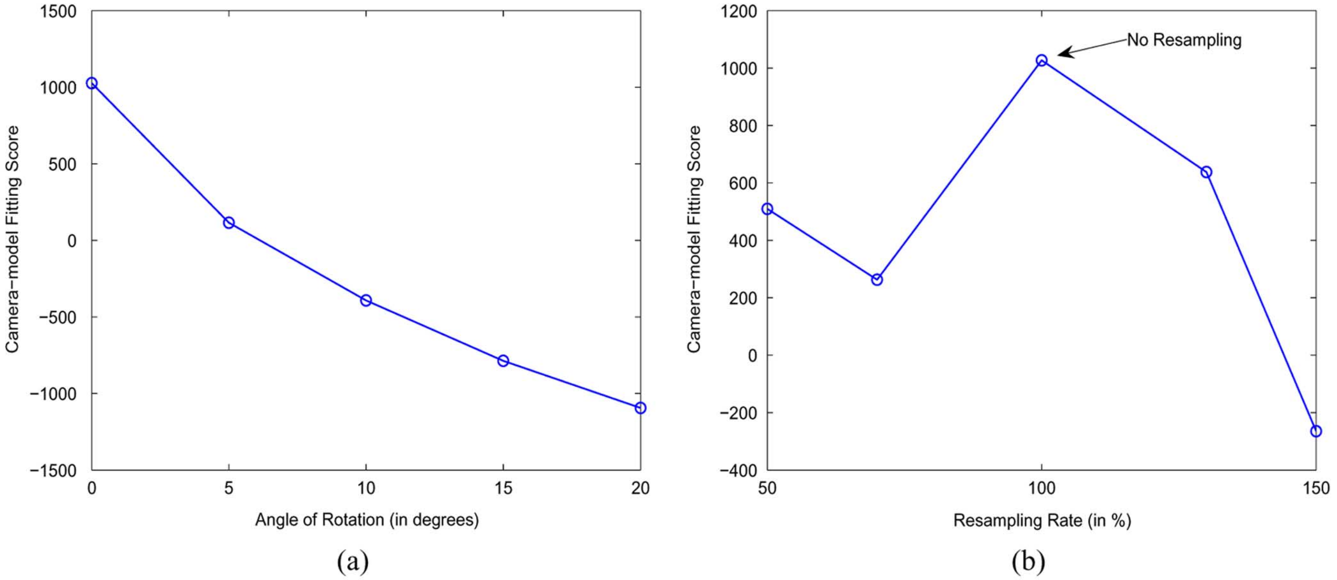
Fig. 12(a) and (b) shows the camera-model fitting score as a function of the angle of rotation and the resampling rate, respectively. For manipulations, such as rotations, the average fitting scores for manipulated images are less than zero as can be seen in Fig. 12(a) and, therefore, the detection algorithm can efficiently identify rotations by setting an appropriate threshold close to zero. For image resampling, the results from 12(b) indicate that the average camera-model fitting score reduces as the resampling rate deviates from 100% and, therefore, these manipulations can be detected with the threshold-based classifier. A similar trend is also observed for additive noise and the fitting score reduces as the strength of additive noise increases.

The estimated manipulation filter coefficients can also be employed to identify the type and parameters of postcamera processing operations. In Fig. 13, we show the frequency response of the estimated manipulation filter coefficients for the different types of manipulations listed in Table II. A closer look at the manipulation filter coefficients in the frequency domain suggests noticeable differences for the different kinds of tampering operations. For such manipulations as average filtering, we observe distinct nulls in the frequency spectrum and the gap between the nulls can be employed to estimate the order of the averaging filter and its parameters. Image manipulations, such as additive noise, result in a white noisy spectrum as shown in Fig. 13(g), and the strength of the noise can be computed from the manipulation filter coefficients. Rotation and downsampling can be identified from the smaller values in the low–high and the high–low bands of the frequency spectrum of the manipulation filter. In our future work, we plan to further investigate on employing the estimated intrinsic fingerprints of postcamera processing operations to provide forensic evidence about the nature and parameters of the tampering that the image has undergone. Such analysis may help recreate the original image from its corresponding tampered versions.



(a) (b)

Fig. 11. Variation of the camera-model fitting score as a function of the filter order for (a) average filtering and (b) median filtering.



(a) (b)

Fig. 12. Variation of the camera-model fitting score as a function of the degree of tampering for (a) image rotations and (b) resampling.

D. Attacking the Proposed Tampering Detection Algorithm

So far in this paper, we have considered direct camera outputs as authentic images and presented methods to distinguish them from other images that have undergone postcamera manipulations. In this subsection, we examine the other side of the problem from the attackers' viewpoint. Given the knowledge of the proposed tampering detection algorithm, the attacker could potentially come up with better tampering operations to foil the detector. We illustrate it with a particular attack as follows.

In Step 1 of the tampering process, the attackers estimate the color interpolation coefficients using component forensics methodologies described in Section III-B. After estimating the color interpolation coefficients, the attacker proceeds to Step 2 to tamper the image by applying such postcamera operations, such as filtering and resampling; then in Step 3, the attacker reenforces the camera constraints with (6) using the estimated camera component parameters obtained earlier in Step 1.

Fig. 14(a) shows the inclass and the outclass similarity scores obtained by comparing the reference patterns with the direct camera outputs and the tampered versions by the aforementioned three-step process, respectively, for the scenario when the camera input is tampered by downsampling to half of its original size in Step 2, before enforcing the camera constraints in Step 3. We notice from the figure that the inclass and the outclass distances are well separated, and an appropriate threshold value $\tau \approx -200$ can be used to distinguish the two classes. The ROC curve computed using the threshold-based classifier is shown alongside in Fig. 14(b). The figure suggests that the classifier still performs well and gives a P_D close to 100% even for low values of P_F close to 1%. The reason behind the superior performance is due to the tampered images that have under-

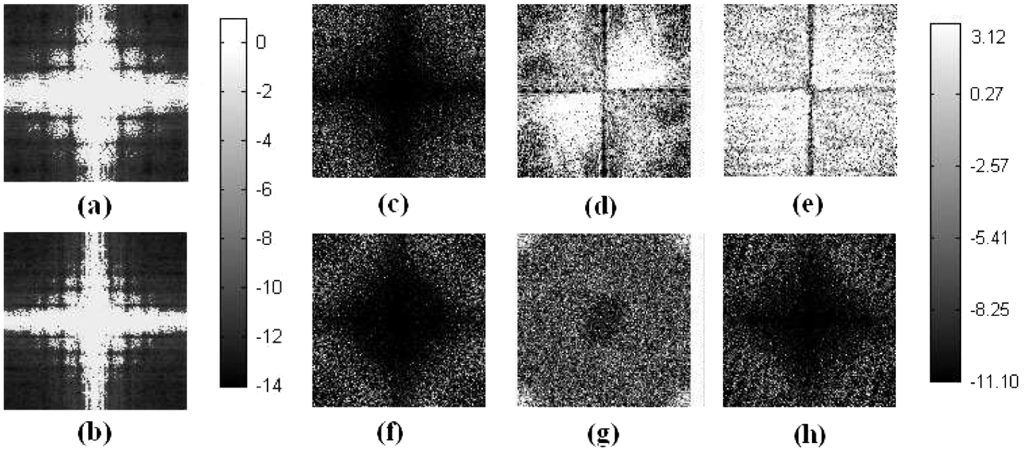


Fig. 13. Frequency response of the manipulation filter for camera outputs that are manipulated by (a) 7 × 7 averaging filter, (b) 11 × 11 averaging filter, (c) 7 × 7 median filter, (d) 20° rotation, (e) 70% resampling, (f) 130% resampling, (g) noise addition with PSNR 20 dB, and (h) histogram equalization. The frequency response is shown in the log scale and shifted so that the dc components are in the center.

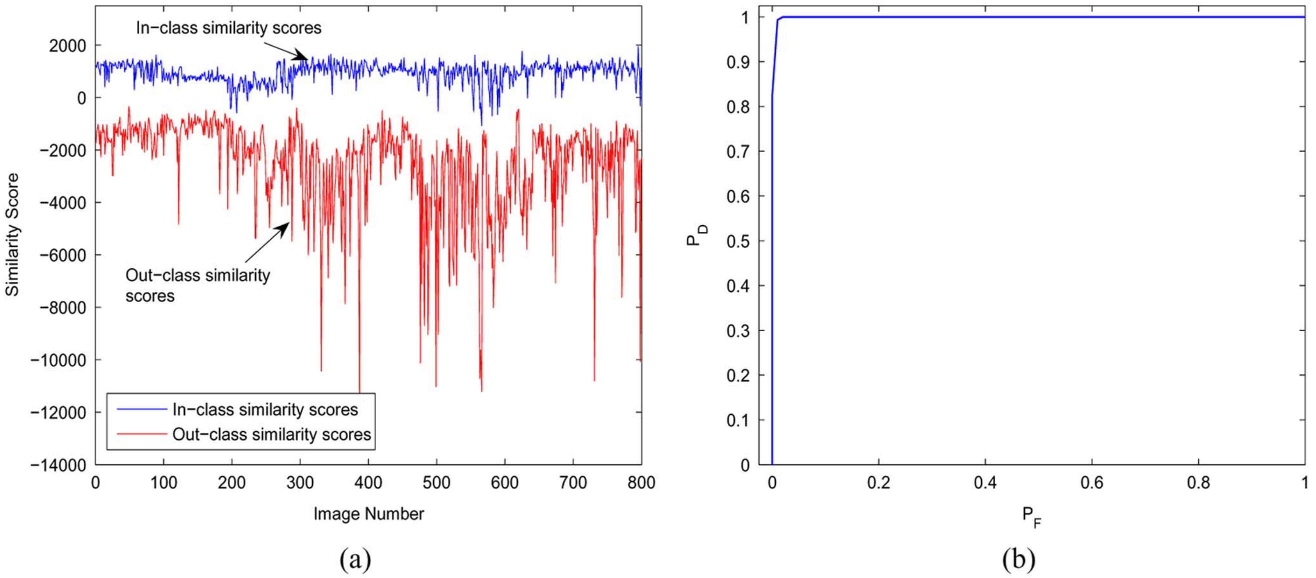


Fig. 14. Performance results for attack I: downsampling by 50% followed by camera-constraint reenforcement. (a) Inclass and outclass similarity scores. (b) Receiver operating characteristics for the tampering detection problem.

gone several manipulations, each of which introduces some inherent traces in the final output image, and the Step 3 restoration process is not able to completely disguise the attacks from the iterative forensic analysis algorithm. Thus, the proposed techniques can efficiently resist such attacks.

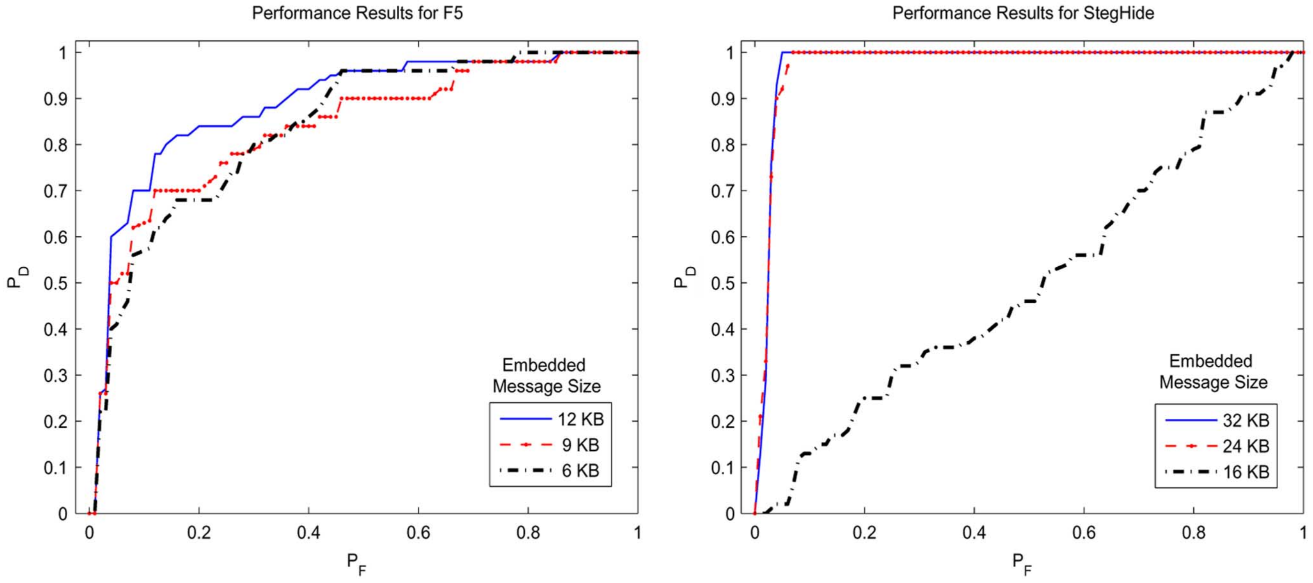
VI. Further Discussions and Applications

The results in the previous section demonstrate that the intrinsic fingerprint traces left behind in the final digital image by the postcamera processing operations can provide a tell-tale mark to robustly detect global manipulations. In this section, we show that the estimated filter coefficients can also be employed to detect other kinds of postcamera processing operations, such as steganographic embedding and watermarking. Further, any change or inconsistencies in the estimated in-camera fingerprints, or the presence of new postcamera fingerprints provides clues to detect cut–paste tampering and to determine whether the given image was produced using a camera, a scanner, or computer graphics software.

A. *Applications to Universal Steganalysis*

A common challenge of steganalysis is how to model the ground truth original nonstego image data. In our work, we consider direct camera outputs as nonstego data and apply the camera model to characterize its properties; image manipulations, such as watermarking and steganography, are then modelled as postprocessing operations applied to camera outputs. In this subsection, we show that these embedding algorithms leave behind statistical traces on the digital image that can be detected by analyzing the coefficients of the manipulation filter, and examine the performance of our proposed techniques for identifying the presence of hidden messages in multimedia data.

We test the performance of the threshold-based detector in distinguishing authentic camera outputs from stegodata. In our experiments, we use the same camera data set with 100



(a)

(b)

Fig. 15. Performance results at different embedding rates for the (a) F5 algorithm and (b) Steghide.

images of size 512 × 512 from Canon Powershot A75 camera [38]. Stego images are then generated by embedding random messages of different sizes into the cover images. Generally speaking, the maximum embedding payload depends on the nature of the cover image and the data-hiding algorithm. For our simulations, we first find the average of the maximum embedding payload across 100 images and then embed messages at 100%, 75%, and 50% of this value. For our study, we consider three popular steganographic embedding methods that employ different approaches to hide information—F5 [6], steghide [5], and spread-spectrum steganography [7].

LSB embedding methods have been widely used for data hiding. Many algorithms such as Jsteg, JPEG hide-and-seek [39], Outguess [40], and F5 [6] embed a secret message into the LSB of the DCT coefficients of the cover image. For a survey of LSB methods, see [41] and the references therein. Most LSB embedding methods, such as JPEG hide-and-seek [39] and Outguess [40], replace the LSB of the DCT coefficients with the secret message, and statistical steganalysis using the χ^2-test can be used to detect them [29]. In our work, we focus on the embedding methods of F5 and steghide.

The F5 technique that has been shown to be resilient to such statistical attacks based on the χ^2-test [6], although it was subsequently broken in [10] by the histogram analysis of DCT coefficients. The F5 embeds data through matrix encoding by decrementing the absolute value of the DCT coefficients. In our experiments with F5, we estimate the average maximum payload across 100 color images to be around 12 kB. The stegoimages are then generated by embedding secret messages of size 12, 9, and 6 kB using the software [42], respectively. The detection results are shown in Fig. 15(a) for different embedding rates. We notice that the proposed algorithms perform with reasonable accuracy giving an average detection accuracy close to 62% and 50%, respectively, at 100% and 75% average embedding rates for false alarm probabilities around 1%. These results are comparable to the wavelet statistics-based steganalysis technique [11], which reports average accuracies of 62% and 52% at the embedding rates of 100% and 78%, respectively.

Steghide preserves the first-order statistics of the image and can provide high message capacity. Steghide employs a graph-theoretic approach to embed the secret messages on multimedia data. The message is hidden by exchanging rather than overwriting pixels [5]. A graph is first constructed from the cover data to the secret message. The pixels to be modified are represented as vertices and are connected to possible partners by edges. A combinatorial problem is then solved to embed the secret message by exchanging samples. In our studies with steghide, we estimate the average maximum payload across 100 color images to be around 32 kB for a 512 × 512 color image. The stegoimages are then generated by embedding secret messages of size 32, 24, and 16 kB using the software [43], respectively. The detection results are shown in Fig. 15(b) for different embedding rates. We notice that the proposed algorithms can efficiently identify steghide at 100% and 75% embedding rates with the probability of identifying stegodata close to 100% for a false alarm probability of 1%. However, the performance reduces significantly when the secret message length is reduced to 50% capacity at 16 kB. These results are better than the wavelet statistics-based steganalysis technique [11], which reports average accuracies of 77% and 60% at 100% and 78% embedding rates, respectively.

Next, we study the performance of spread-spectrum embedding methods. Block-DCT-based spread-spectrum embedding has been widely used in literature for data hiding, watermarking, and steganography [1] for a wide variety of applications. Detecting spread-spectrum steganography has been a challenging problem over the last decade, and statistics-based schemes typically do not perform well in distinguishing original cover data and stegopictures. To our best knowledge, the only work that addresses spread-spectrum steganalysis is by Avcibas *et al.* [12],

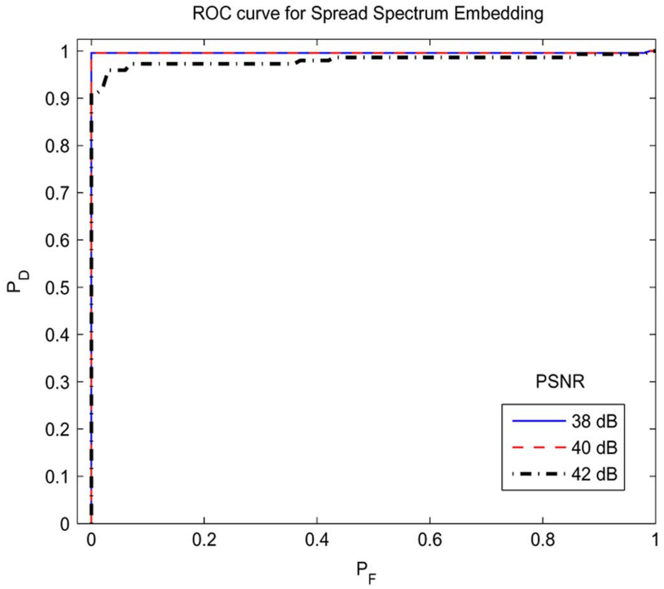


Fig. 16. Performance results for spread-spectrum embedding at different PSNRs.

where it was shown that image-quality metrics may be used as features to identify such embedding. In their work, the authors show that they can attain an average probability of correct decision of 80% with 40% false alarm probability when tested with ten images. We test the performance of the proposed intrinsic fingerprint system for spread-spectrum embedding. In our experiments, we use the same camera data set with 100 Canon Powershot A75 images of size 512×512 as our authentic set. Stegoimages are then generated by adding pseudorandom watermarks at different peak signal-to-noise ratios (PSNRs) of 38, 40, and 42 dB. The manipulation filter coefficients are estimated for the cover and the stegodata, and classified with the threshold-based classifier. Fig. 16 shows the performance results for different PSNRs. We note that the average identification accuracy is close to 100% for PSNRs of 38 and 40 dB, and reduces to 91% for 42-dB PSNR. These results demonstrate the superior performance of the proposed techniques.

In addition to the three steganographic schemes mentioned before, we also test the performance of our algorithms for such embedding techniques as stochastic modulation [8] and perturbed-quantization (PQ) steganography [9], [44]. In stochastic modulation steganography [8], a weak noise signal with a noise distribution chosen to mimic the noise produced by the image-acquisition device is added to the cover image to embed the message bits. In the case of digital cameras, it has been shown that the sensor and hardware noise are best modelled to be Gaussian distributed [8], [45] and, therefore, detecting stochastic modulation steganography can be considered equivalent to detecting the presence of additive Gaussian noise in an image captured by a digital camera. Our results suggest that such embedding can be detected with very high accuracy with a P_D that is close to 100% for low values of P_F about 1% using the proposed forensic analysis techniques. Perturbed quantization steganography embeds information in the DCT coefficients by quantizing the values either up or down depending upon the message to embed. The set of changeable coefficients is first found by identifying those coefficients whose fractional part (i.e., the difference between the actual value and the quantized value) is lower than a prechosen threshold [9]. For our experiment with PQ steganography, we use the 100 Canon Powershot A75 images of size 512×512, JPEG compressed in the camera with the default quality factor close to 97%, as our authentic set. Stegoimages are created by randomly embedding messages into these images and quantizing them to a quality factor of 70%. Steganalysis for this scheme is more challenging and the proposed techniques are able to identify such manipulations with P_D close to 70–80% under a $P_F \approx 25\%$.

B. Distinguishing Camera Capture From Other Image-Acquisition Processes

The proposed forensics methodology can be used to authenticate the source of the digital color image. Evidence obtained from such forensic analysis would provide useful forensic information to law enforcement and intelligence agencies as to when a given image was actually captured with a camera or scanner, or generated using computer graphics software. We demonstrate this application with two case studies.

1) Photographs versus Scanned Images: Digital cameras and image scanners are two main categories of image-acquisition devices. While a large amount of natural scene pictures are taken with digital cameras, scanners have been increasingly used for digitizing documents. Rapid technology development and the availability of high-quality scanners has, in part, led to more sophisticated digital forgeries. In this case study, we are interested in determining whether a digital image is produced by a camera or a scanner. The motivation behind employing the proposed techniques for device identification is based on the observation that the manipulation filter coefficients for an authentic camera output would be close to a delta function, and the corresponding coefficients for a scanned image would represent the scan process.

For our study, we choose 25 different images from four camera models to give a total of 100 images for the camera image data set. We then collect another set of 25 different photographic images from several cameras with diverse image content. These photographs are printed and then scanned back using four different scanner models: 1) Canon CanoScan D1250U2F, 2) Epson Perfection 2450 photo, 3) Microtek ScanMaker 3600, and 4) Visioneer OneTouch 5800USB. These $25 \times 4 = 100$ images form our scanned image data set. We test our proposed methods for these 200 images. The frequency response of the manipulation filter is estimated and compared with a reference pattern. The ROC obtained using the threshold-based classifier is shown in Fig. 17. Here, P_D denotes the fraction of scanned images that are correctly classified as scanned, and P_F represents the fraction of camera outputs misclassified as scanned. We observe from the figure that the probability of correct decision P_D is around 92% for a 1% false probability rate. These results indicate that our proposed methods can effectively distinguish between the camera-captured and scanned images.

2) Photographs versus Photo Realistic Computer Graphics: With an increasing number of sophisticated processing tools,

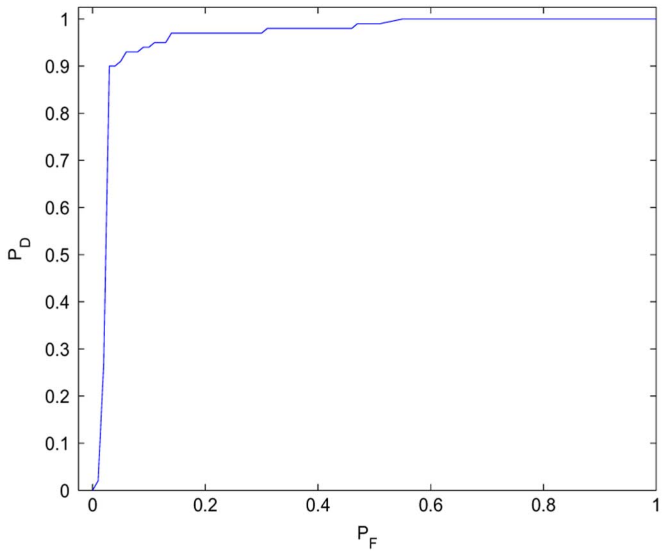


Fig. 17. Receiver operating characteristics for classifying authentic camera outputs from scanned images.

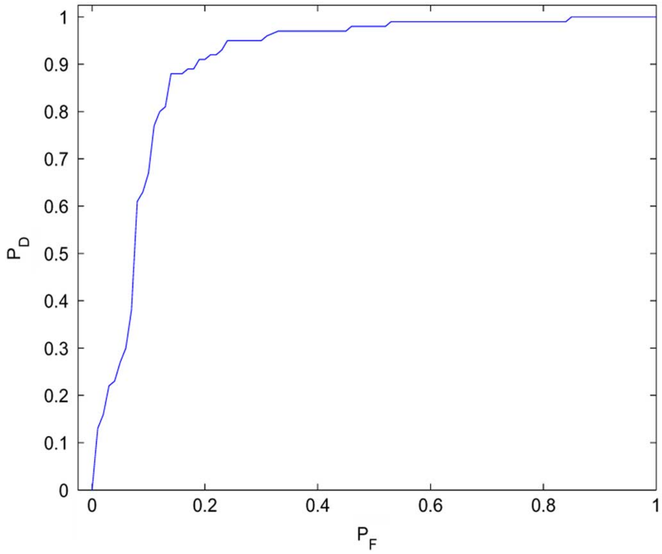


Fig. 18. ROCs for classifying authentic camera outputs from photorealistic computer graphics.

creating realistic imagery has become easier. Modern graphic synthesis and image rendering tools can be used to reproduce photographs to a very high degree of precision and accuracy and, therefore, the problem of distinguishing camera outputs from photorealistic computer graphics has become important. In this case study, we employ our proposed framework to distinguish digital photographic images and photorealistic graphics images. For our study, we use a set of 100 images from four camera models to create the camera image dataset. A randomly chosen set of 100 photorealistic computer graphics images, obtained from the Columbia dataset [46] constitute our photorealistic computer graphics data set. We use a cropped subimage of size 512×512 to estimate the coefficients of the manipulation filter. The estimated frequency response is then compared with the reference pattern and a threshold-based classifier is used to distinguish authentic camera outputs from graphics images. The results of our analysis, in terms of the ROC, are shown in Fig. 18. Here, P_D denotes the fraction of graphics images that are correctly classified as photorealistic, and P_F represents the fraction of photographs classified as computer generated. A large area under the ROC curve suggests that our proposed method can distinguish between the two classes. These results are comparable to the geometry-based features proposed in [14], and are better than the wavelet features [28] and the cartoon features-based classifiers tested in [14]. Different from the geometry-based features in [14] that are motivated by the modelling, the computer graphics creation tools, and the artifacts produced therein, our method focuses on finding the algorithms and parameters of the imaging process in digital cameras to distinguish digital photographic images from photorealistic computer graphics.

C. Detecting Cut-and-Paste Forgeries Based on Inconsistencies in Component Parameters

Creating a tampered image by cut-and-paste forgery often involves obtaining different parts of the image from pictures captured using different cameras that may employ a different set of algorithms/parameters for its internal components. Inconsistencies in the estimated intrinsic fingerprint traces left behind by camera components can be used to identify such digital forgeries as cut-and-paste operations. Here, we illustrate this with a case study. We create a tampered picture of size 2048×2036 by combining parts of two images taken using two different cameras. In Fig. 19(a) and (b), we show the tampered picture and its individual parts marked with different colors. The regions displayed in white in Fig. 19(b) are obtained from an image taken with the Canon Powershot S410 digital camera, and the black parts are cropped and pasted from a picture shot using the Sony Cybershot DSC P72 model.

To identify the intrinsic camera fingerprints in different parts of the picture, we examine the image using a sliding window of 256×256 with step size 64×64, and estimate the color interpolation coefficients in each 256×256 block [49]. The k-means clustering algorithm [47] is then employed to cluster these features into two classes. With a step size of 64, each individual 64×64 subblock would be analyzed 16 times to provide 16 different clustering results; the clustering results are represented as binary values (0 or 1) as labels for the two classes. Fig. 19(c) shows the average of the clustering labels from these 16 subblocks. As shown in Fig. 19(c), our results indicate that the features are clustered distinctly in two separate classes with the gray area in between representing the transition from one class to the other. In this particular case, we notice that the manipulated picture has tell-tale traces from two different cameras and has therefore been tampered with.

VII. Conclusion

In this paper, we propose a set of forensic signal processing techniques to verify whether a given digital image is a direct camera output. We introduce a new formulation to study the problem of image authenticity. The proposed formulation is based on the observation that each in-camera and postcamera processing operation leaves some distinct intrinsic fingerprint traces on the final image. We characterize the properties of a direct camera output using a camera model, and estimate its component parameters and the intrinsic fingerprints. We consider any further

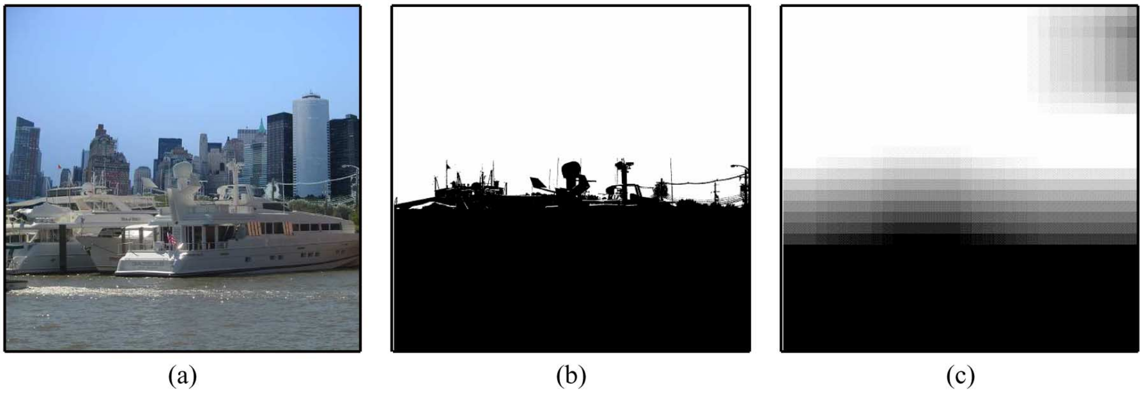


Fig. 19. Applications to source authentication showing a (a) sample-tampered image, (b) regions obtained from the two cameras, and (c) results from clustering the color interpolation coefficients (black: Sony Cybershot DSC P72; white: Canon Powershot S410; shades of gray: likelihood that the region is from Canon Powershot S410 with a value close to white denoting a higher likelihood).

postcamera processing as a manipulation filter, and find the coefficients of its linear shift-invariant approximation using blind deconvolution. A high similarity of the estimated coefficients and the reference pattern that corresponds to no manipulations certifies the integrity of the given image. We show through detailed simulation results that the proposed techniques can be used to identify different types of postcamera processing, such as filtering, resampling, rotation, etc. Evidence obtained from such forensic analysis is used to build a universal steganalyzer to determine the presence of hidden messages in multimedia data. Our results suggest that we can efficiently detect different types of embedding methods, such as least-significant bit (LSB) and spread-spectrum techniques with high accuracy. The estimated postcamera fingerprints are also employed for image-acquisition forensics to establish whether a given digital image is from a digital camera, a scanner, or computer graphics software. Overall, our proposed techniques provide a common framework for a broad range of forensic analyses on digital images.

APPENDIX
CONVEXITY OF THE OPTIMIZATION PROBLEM AND UNIQUENESS OF SOLUTION

In this Appendix, we show that the optimization formulation in (7) is convex if the camera's color interpolation coefficients are known. A function J is said to be convex if for any u_1, u_2 and $0 \leq \lambda \leq 1$, we have

$$J(\lambda u_1 + (1-\lambda)u_2) \leq \lambda J(u_1) + (1-\lambda)J(u_2).$$

Since $J(u)$ in (7) is a sum of two quadratic functions, it is sufficient to show that these two functions are convex. Let

$$J(u) = \sum_{c=1}^{3} \left(J_c^1(u) + J_c^2(u)\right)$$

where

$$J_c^1(u) = \sum_{x,y} \left[\sum_{m,n} u(m,n,c) \times \left(\hat{S}_t(x-m,y-n,c) - S_t(x-m,y-n,c)\right)\right]^2$$

and

$$J_c^2(u) = \left(\sum_{m,n} u(m,n,c) - 1\right)^2.$$

Here, $\hat{S}_t$ denotes the estimate of the test image S_t obtained by imposing the camera constraints as shown in (15), at the bottom of the page, where $\alpha_{\Re_i}$ denotes the color interpolation coefficients employed in the camera to render the test image S_t. In the absence of additional information, the values of $\alpha_{\Re_i}$ can be nonintrusively estimated from the test image as long as S_t is a direct camera output or an image that has undergone minor levels of postinterpolation processing. Now, defining

$$\varphi_i(x,y,c) = \sum_{m,n} u_i(m,n,c)(\hat{S}_t(x-m,y-n,c) - S_t(x-m,y-n,c))$$

$$\hat{S}_t(x,y,c) = \begin{cases} \sum_{m,n} \alpha_{\Re_i}(m,n,c) S_t(x-m,y-n,c), & \forall\{x,y\} \in \Re_i, \quad \text{and } 1 \leq c \leq 3 \\ S_t(x,y,c), & \text{otherwise} \end{cases} \qquad (15)$$

we get

$$\begin{aligned}
&J_c^1(\lambda u_1 + (1-\lambda)u_2)\\
&\quad= \sum_{x,y} [\lambda\varphi_1(x,y,c) + (1-\lambda)\varphi_2(x,y,c))]^2\\
&\quad= \lambda\sum_{x,y}\varphi_1(x,y,c)^2 + (1-\lambda)\sum_{x,y}\varphi_2(x,y,c)^2\\
&\qquad - \lambda(1-\lambda)\times\sum_{x,y}(\varphi_1(x,y,c) - \varphi_2(x,y,c))^2\\
&\quad= \lambda J_c^1(u_1) + (1-\lambda)J_c^1(u_2) - \lambda(1-\lambda)\\
&\qquad \times\sum_{x,y}(\varphi_1(x,y,c) - \varphi_2(x,y,c))^2\\
&\quad\le \lambda J_c^1(u_1) + (1-\lambda)J_c^1(u_2)
\end{aligned}$$

where the last inequality follows from $0 \le \lambda \le 1$. This shows that J_c^1 is convex. Similarly, we can show that the quadratic function J_c^2 is also convex and, therefore, establishes the convexity of J.

To show that the solution of the optimization problem is unique, we make use of a theorem in optimization theory that states that the solution of a convex optimization problem with a cost function J is unique if the cost function is unimodal [34], [48] (i.e., $\nabla^2 J(u) > 0$ for all u). Defining $\Psi(x,y,c) = S_t(x,y,c) - \hat{S}_t(x,y,c)$, we can show that

$$\begin{aligned}
&\frac{\partial^2 J}{\partial u(a_i,b_i,c)\partial u(a_j,b_j,c)}\\
&\quad= 2\sum_{x,y}\Psi(x-a_i,y-b_i,c)\Psi(x-a_j,y-b_j,c)\\
&\qquad + 2u(a_i,b_i,c)u(a_j,b_j,c),\\
&\quad= 2 < \Lambda_{(a_i,b_i,c)}, \Lambda_{(a_j,b_j,c)} >
\end{aligned}$$

where $\Lambda_{(a_i,b_i,c)}$ represents a vector of length $(H \times W + 1)$ consisting of all the elements of $\Psi(x-a_i, y-b_i, c)$ for all x and y along with the element $u(a_i,b_i,c)$. Arranging the vectors $\Lambda_{(a_i,b_i,c)}$ column-wise, we construct the matrix $\Omega_c = [\Lambda_{(a_1,b_1,c)}\Lambda_{(a_1,b_2,c)}\ldots]$ of dimension $(H \times W + 1) \times (N_u^2)$ for $c = 1,2,3$. We can then show that $\nabla^2 J(u) = 2\sum_{c=1}^{3}\Omega_c\Omega_c^T > 0$. Thus, the cost function is unimodal and, therefore, its solution is unique.

Acknowledgment

The authors would like to thank Dr. H. Gou for her valuable discussions and help on scanner forensics.

References

[1] M. Wu and B. Liu, *Multimedia Data Hiding*. New York: Springer-Verlag, 2002.

[2] J. Fridrich, "Image watermarking for tamper detection," in *Proc. IEEE Int. Conf. Image Processing*, Chicago, IL, Oct. 1998, vol. 2, pp. 404–408.

[3] A. Swaminathan, Y. Mao, and M. Wu, "Robust and secure image hashing," *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 2, pp. 215–230, Jun. 2006.

[4] A. Giannoula, N. V. Boulgouris, D. Hatzinakos, and K. N. Plataniotis, "Watermark detection for noisy interpolated images," *IEEE Trans. Circuits Syst. II*, vol. 53, no. 5, pp. 359–363, May 2006.

[5] S. Hetzl and P. Mutzel, "A graph-theoretic approach to steganography," presented at the 9th IFIP Conf. Communications Multimedia Security, Salzburg, Austria, Sep. 2005.

[6] A. Westfeld, "F5-A steganographic algorithm: High capacity despite better steganalysis," presented at the Information Hiding Workshop, Pittsburgh, PA, Apr. 2001.

[7] L. M. Marvel, C. G. Boncelet, Jr., and C. T. Retter, "Spread spectrum image steganography," *IEEE Trans. Image Process.*, vol. 8, no. 8, pp. 1075–1083, Aug. 1999.

[8] J. Fridrich and M. Goljan, "Digital image steganography using stochastic modulation," in *Proc. SPIE, Security Watermarking Multimedia Contents*, San Jose, CA, Jan. 2003, vol. 5020, pp. 191–202.

[9] J. Fridrich, M. Goljan, and D. Soukal, "Perturbed quantization steganography," *Multimedia Syst.*, vol. 11, no. 2, pp. 98–107, Dec. 2005.

[10] J. Fridrich, M. Goljan, and D. Hogea, "Steganalysis of JPEG images: Breaking the F5 algorithm," in *Proc. Int. Workshop Information Hiding*, Oct. 2002, pp. 310–323.

[11] S. Lyu and H. Farid, "Steganalysis using higher-order image statistics," *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 1, pp. 111–119, Mar. 2006.

[12] I. Avcibas, N. Memon, and B. Sankur, "Steganalysis using image quality metrics," *IEEE Trans. Image Process.*, vol. 12, no. 2, pp. 221–229, Feb. 2003.

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

[14] T.-T. Ng, S.-F. Chang, J. Hsu, L. Xie, and M.-P. Tsui, "Physics-motivated features for distinguishing photographic images and computer graphics," presented at the ACM Multimedia, Singapore, Nov. 2005.

[15] A. C. Popescu and H. Farid, "Exposing digital forgeries by detecting traces of re-sampling," *IEEE Trans. Signal Process.*, vol. 53, no. 2, pp. 758–767, Feb. 2005.

[16] J. Lukas and J. Fridrich, "Estimation of primary quantization matrix in double compressed JPEG images," presented at the Digital Forensics Research Workshop, Cleveland, OH, Aug. 2003.

[17] C. Y. Lin and S. F. Chang, "A robust image authentication method distinguishing JPEG compression from malicious manupulation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 2, pp. 153–168, Feb. 2001.

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

[19] M. K. Johnson and H. Farid, "Exposing digital forgeries through chromatic aberration," in *Proc. Workshop Multimedia Security*, Geneva, Switzerland, 2006, pp. 48–55.

[20] 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.

[21] A. C. Popescu and H. Farid, "Statistical tools for digital forensics," in *Proc. 6th Int. Workshop Information Hiding Lecture Notes in Computer Science*, Toronto, ON, Canada, May 2004, vol. 3200, pp. 128–147.

[22] 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.

[23] H. Farid, "Digital image ballistics from JPEG quantization," Dept. Comput. Sci., Dartmouth College, Tech. Rep. TR2006-583, 2006.

[24] M. K. Johnson and H. Farid, "Exposing digital forgeries by detecting inconsistencies in lighting," presented at the ACM Multimedia Security Workshop, New York, 2005.

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

[26] J. Lukas, J. Fridrich, and M. Goljan, "Detecting digital image forgeries using sensor pattern noise," in *Proc. SPIE Conf. Security, Steganography, Watermarking of Multimedia Contents*, Jan. 2006, vol. 6072, pp. 362–372.

[27] I. Avcibas, S. Bayram, N. Memon, M. Ramkumar, and B. Sankur, "A classifier design for detecting image manipulations," in *Proc. Int. Conf. Image Processing*, Singapore, Oct. 2004, vol. 4, pp. 2645–2648.

[28] 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, Jun. 2003.

[29] A. Westfeld and A. Pfitzmann, "Attacks on steganographic systems," presented at the Information Hiding Workshop, Dresden, Germany, Sep. 1999.

[30] J. Fridrich and M. Goljan, "Practical steganalysis of digital images—State of the art," in *Proc. SPIE Conf. Security and Watermarking of Multimedia Contents*, San Jose, CA, Jan. 2002, vol. 4675, pp. 1–13.

[31] J. Adams, "Interaction between color plane interpolation and other image processing functions in electronic photography," in *Proc. SPIE Cameras Systems for Electronic Photography Scientific Imaging*, San Jose, CA, Feb. 1995, vol. 2415, pp. 144–151.

[32] J. Adams, K. Parulski, and K. Spaulding, "Color processing in digital cameras," *IEEE Micro.*, vol. 18, no. 6, pp. 20–30, Nov./Dec. 1998.

[33] A. Swaminathan, M. Wu, and K. J. R. Liu, "Non-intrusive component forensics of visual sensors using output images," *IEEE Trans. Inf. Forensics Security*, vol. 2, no. 1, pp. 91–106, Mar. 2007.

[34] D. Kundur and D. Hatzinakos, "A novel blind deconvolution scheme for image restoration using recursive filtering," *IEEE Trans. Signal Process.*, vol. 46, no. 2, pp. 375–390, Feb. 1998.

[35] D. Kundur and D. Hatzinakos, "Blind image deconvolution," *IEEE Signal Process. Mag.*, vol. 13, no. 3, pp. 43–64, May 1996.

[36] G. R. Ayers and J. C. Dainty, "Iterative blind deconvolution method and its applications," *Opt. Lett.*, vol. 13, no. 7, pp. 547–549, Jul. 1988.

[37] A. Swaminathan, M. Wu, and K. J. R. Liu, "Image tampering identification using blind deconvolution," in *Proc. IEEE Int. Conf. Image Processing* , Atlanta, GA, Oct. 2006, pp. 2311–2314.

[38] A. Swaminathan, M. Wu, and K. J. R. Liu, "Intrinsic fingerprints for image authentication and steganalysis," presented at the SPIE Conf. Security, Steganography Watermarking of Multimedia Contents, San Jose, CA, Jan. 2007.

[39] A. Latham, Jpeg Hide and Seek. [Online]. Available: linux01.gwdg.de/alatham/stego.

[40] N. Provos, Outguess. [Online]. Available: www.outguess.org.

[41] N. Provos and P. Honeyman, "Hide and seek: An introduction to steganography," *IEEE Security Privacy Mag.*, vol. 1, no. 3, pp. 32–44, May/Jun. 2003.

[42] A. Westfeld, F5. [Online]. Available: wwwwrn.inf.tu-dresden.de/westfeld/f5.

[43] S. Hetzl, Steghide. [Online]. Available: steghide. sourceforge.net.

[44] J. Fridrich, M. Goljan, and D. Soukal, "Perturbed quantization steganography with wet paper codes," in *Proc. ACM Multimedia Security Workshop*, Magdeburg, Germany, Sep. 2004, pp. 4–15.

[45] G. E. Healey and R. Kondepudy, "Radiometric CCD camera calibration and noise estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 3, pp. 267–276, Mar. 1994.

[46] T.-T. Ng, S.-F. Chang, J. Hsu, and M. Pepeljugoski, "Columbia photographic images and photorealistic computer graphics dataset," Columbia Univ., New York, ADVENT Tech. Rep. 205–2004–5, Feb. 2005.

[47] R. O. Duda, P. E. Hart, and D. G. Stork*, Pattern Classification*, 2nd ed. New York: Wiley, 2000.

[48] J. B. H-Urrutty and C. Lemarecal*, Convex Analysis and Minimization Algorithms*. New York: Springer-Verlag, 1993.

[49] A. Swaminathan, M. Wu, and K. J. R. Liu, "Component forensics of digital cameras: A non-intrusive approach," presented at the Conf. Information Sciences Systems, Princeton, NJ, Mar. 2006.

Ashwin Swaminathan (S'05) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Madras, India, in 2003 and is currently pursuing the Ph.D. degree in signal processing and communications in the Department of Electrical and Computer Engineering at the University of Maryland, College Park.

He was a Research Intern with Hewlett-Packard Labs, Palo Alto, CA, in 2006 and with Microsoft Research, Redmond, WA, in 2007. His research interests include multimedia forensics, information security, and authentication.

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

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, University of Maryland at College Park, where she is currently an Associate Professor. Previously, she was with the NEC Research Institute and Panasonic Laboratories. 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.

Dr. Wu received a National Science Foundation CAREER Award in 2002, a University of Maryland at College Park 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 a 2005 IEEE Signal Processing Society Best Paper Award. She served as Finance Chair for the 2007 IEEE International Conference on Acoustic, Speech, and Signal Processing (ICASSP), and is an Associate Editor of IEEE SIGNAL PROCESSING LETTERS and an Area Editor for the E-Newsletter of *IEEE Signal Processing Magazine*.

K. J. Ray Liu (F'03) received the B.S. degree from the National Taiwan University, Taipei, Taiwan, R.O.C., and the Ph.D. degree from the University of California, Los Angeles, both in electrical engineering.

He is Professor and Associate Chair, Graduate Studies and Research, of the Electrical and Computer Engineering Department, University of Maryland, College Park, where he is Director of Communications and the Signal Processing Laboratory. He leads the Maryland Signals and Information Group, conducting research that encompasses broad aspects of information technology, including signal processing, communications, networking, information forensics and security, and biomedical and bioinformatics imaging.

Dr. Liu is the recipient of 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 the National Science Foundation Young Investigator Award. He also received various teaching and research recognitions from the University of Maryland, including university-level Distinguished Scholar–Teacher Award, Invention of the Year Award, and college-level Poole and Kent Company Senior Faculty Teaching Award. He is Vice President—Publications and on the Board of Governors of the IEEE Signal Processing Society. 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*.