Intrinsic Fingerprints for Image Authentication and Steganalysis

Ashwin Swaminathan, Min Wu, and K. J. Ray Liu
{ashwins, minwu, kjrliu}@eng.umd.edu
Electrical and Computer Engineering Department
University of Maryland, College Park, U.S.A

ABSTRACT

With growing popularity of digital imaging devices and low-cost image editing software, the integrity of image content can no longer be taken for granted. This paper introduces a methodology for forensic analysis of digital camera images, based on the observation that many in-camera and post-camera processing operations leave distinct traces on digital images. We present methods to identify these intrinsic fingerprint traces of the various processing operations and employ them to verify the authenticity of digital data. We develop an explicit imaging model to characterize the properties that should be satisfied by a direct camera output, and model any further processing applied to the camera captured image by a manipulation filter. Utilizing the manipulation filter coefficients and reference patterns estimated from direct camera outputs using blind deconvolution techniques, the proposed methods are capable of detecting manipulations made by previously unseen operations and steganographic embedding.

1. INTRODUCTION

Over the last decade, digital images have become increasingly popular and have been used in a growing number of applications, from military and reconnaissance to medical diagnosis and consumer photography. With such widespread popularity and the advent of low-cost and sophisticated image editing softwares, the integrity of image content can no longer be taken for granted and a number of forensic related questions arise amidst such widespread use. For example, given the final digital image after a chain of processing, can we determine how the image was created? What technology(s) were used? What processing/alterations were done to the image content after capture? Is the image tampered or manipulated in some way? Does it contain any hidden information?

Some of these forensic questions are related to identifying the source of the digital image, and determining possible tampering. Evidence obtained from such forensic analysis would provide useful forensic information to law enforcement and intelligence agencies as to if the given image was actually captured with a camera (or generated by other means) and to establish the authenticity of the digital image. Techniques such as semi-fragile image watermarking and robust image hashing [1] have been proposed to establish the authenticity of the data. However, these methods require that an external watermark or signature, also referred to as an *extrinsic fingerprint*, be inserted at the time of creation of multimedia data. This imposes several restrictions on its usage 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 non-intrusive methods to distinguish authentic images from manipulated ones.

In this work, we develop a novel methodology for forensic analysis. Our proposed techniques are based on the observation that each in-camera and post-camera processing operation leaves some inherent traces on the final output image that are characteristic of the processing block. We shall refer to these traces as the *intrinsic fingerprints* as they directly capture the changes made to the image by the various processing operations during acquisition and after it has been shot by a camera. By characterizing the properties of a direct camera output using an image acquisition model, we present methods to derive the intrinsic fingerprints of in-camera and post-camera processing operations. The presence or absence of possible post-camera manipulations is then detected by examining the similarities between the intrinsic fingerprints estimated from the test image and the ones obtained from direct camera outputs. The proposed algorithm does not require any prior knowledge of the nature of the processing operation. Thus, it can identify previously unseen manipulations and can act as a front-end for detecting digital forgeries.

Related prior works fall into two main categories. In the tampering detection literature, there have been works that try to identify inauthentic manipulations by defining the properties of a tampered image in terms of the distortions it goes through. In doing so, some works assume that creating a tampered image involves a series of processing operations, which might include resampling, JPEG compression [2], lens distortions, gamma correction [3], inconsistent noise patterns [4], and alternations in correlations introduced by color interpolation [5]. Based on this observation, they propose techniques to identify such manipulations by extracting certain salient features that would help distinguish such tampering from authentic data [4]. Although these methods can be employed to identify the type, and the parameters of the post-processing operation, it would require an exhaustive search over all the numerous kinds of post-processing operations to detect tampering. Unlike the existing work that aims to derive intrinsic fingerprints of each processing module separately, the proposed forensic analysis methodology does not distinguish between different types of processing, and thus provides a universal framework to identify a several possible tampering operations.

Image manipulations such as watermarking and steganography can also be considered as post-processing operations applied to camera outputs. A steganographic system embeds hidden messages into the host image by performing appropriate modifications on the data. Over the past decade, with the growth of internet and widespread popularity of multimedia data, digital images have been used as a common medium for conveying secret information. A large number of steganographic embedding methods [6, 7] have been proposed to enable covert communication. At the same time, numerous steganalysis techniques [8–12] to identify hidden messages have also been developed to counter steganography. While embedding-specific steganalysis algorithms [8–10] target specific steganography methods, universal steganalysis techniques [11, 12] aim to detect covert communication independent of the embedding algorithm. With an increasing number of new evolving steganographic embedding algorithms, there is a strong need for robust methods for blind steganalysis. Proposed techniques using intrinsic fingerprints help identify the traces left behind by such post-camera manipulations, and provides an universal methodology for blind steganalysis.

The paper is organized as follows. In Section 2, we present the system model and problem formulation. Section 3 introduces the proposed methodology to estimate the intrinsic fingerprints. We show that the proposed method is universal and can distinguish genuine photographs from manipulated images. Detailed simulation results are presented in Section 4, and the final conclusions are drawn in Section 5.

2. SYSTEM MODEL AND PROBLEM FORMULATION

In this section, we discuss the *image acquisition* model for direct camera outputs and present our problem formulation. The imaging process in digital cameras is shown in Fig. 1. The light from the scene pass through the lens and the optical filters and is finally recorded by the detectors. Most digital cameras use a color filter array (CFA) on top of the sensors to capture the real-world scene. The CFA enables the sensors to sample a particular color component of visible light from the real-world scene S. Assuming a CFA pattern p, the sampled image S_p can be represented as

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

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 its neighboring pixel values to obtain $S_p^{(I)}$. This interpolated color image $S_p^{(I)}$ goes though a post-processing stage which might include operations such as white balancing, color correction, and compression [13] to form the camera output S_d at the *point A* in the information processing chain as shown in Fig. 1.

In our recent work [14], we propose *component forensics* as a methodology to reverse-engineer the imaging process and to estimate the algorithms employed in various components inside the digital camera. We show that the parameters of such important camera components as color filter array and color interpolation can be robustly estimated solely using output images [15]. Our algorithm estimates the color interpolation coefficients through texture classification and linear approximation, and finds the CFA pattern that minimizes the interpolation errors.

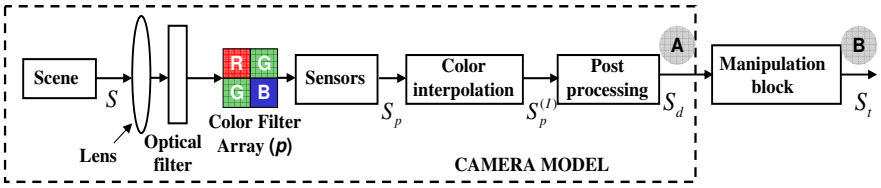

Figure 1. System Model for Image Authentication. *Point A* corresponds to a direct camera output and *point B* represents a manipulated image.

In this work, we build upon component forensics to develop robust image authentication systems for *verifying* if a given digital image is a direct camera output (corresponding to *point A* in Fig. 1) or if it has undergone further processing (as in *point B* in Fig. 1). Using the test image, we estimate the camera model parameters employing component forensic methodologies [14]. We represent any further processing on the camera outputs (if any) as a *manipulation filter*, and find its coefficients using blind deconvolution. The estimated intrinsic fingerprints are then compared to an *identity* transform (corresponding to the fingerprints of a direct camera output with no additional processing), and a high similarity verifies that the test image is an unmanipulated camera output. The proposed algorithm does not require any prior knowledge about the nature of the processing block, and therefore provides a universal technique to distinguish authentic camera outputs from tampered pictures, watermarked, and stego data.

3. ESTIMATING INTRINSIC FINGERPRINTS

In this section, we present methods to estimate the intrinsic fingerprints. Given a test image S_t, we consider it to correspond to the *point B* in Fig. 1, and obtained by filtering the actual camera output S_d (*point A* in Fig. 1) by the manipulation filter. We estimate the inverse of the manipulation filter coefficients, u, using blind deconvolution techniques, and a high similarity of u with the identity transform suggests that the test image is not manipulated [16].

The inverse of the manipulation filter coefficients u can be obtained by formulating it as a constrained optimization problem. Let S_{te} represent the estimate of the camera output obtained by passing S_t through the inverse manipulation filter u, *i.e.,*

$$S_{te}(x,y,c) = \sum_{m,n} u(m,n,c) S_t(x-m, y-n, c), \text{ for } 1 \le c \le 3. \tag{2}$$

The coefficients of the inverse filter are obtained by minimizing the overall camera model fitting error,

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

where $\hat{S}_{te}$ denotes the image formed from S_{te} by imposing *camera constraints*. To impose camera constraints, the camera component parameters such as the interpolation coefficients $\alpha_{\Re_i}$ in each of the three texture based regions $\Re_i$* are estimated† from S_{te} using component forensic methodologies [15] , and the image S_{te} is re-interpolated using these coefficients as

$$\hat{S}_{te}(x,y,c) = \begin{cases} \sum_{m,n} \alpha_{\Re_i}(m,n,c) S_{te}(x-m, y-n, c) & \forall \{x,y\} \in \Re_i, \text{ and } 1 \le c \le 3, \\ S_{te}(x,y,c) & \text{otherwise.} \end{cases} \tag{4}$$

*In [15], we consider three types of texture based regions such as regions with significant horizontal gradient, vertical gradient, and smooth regions. The image pixels are classified into these three regions $\Re_i$ based on the gradient values in the local neighborhood.

†The estimated camera model parameters obtained from the test image would be close to the actual parameters because the estimation algorithm is robust to moderate levels of post-camera processing [15]

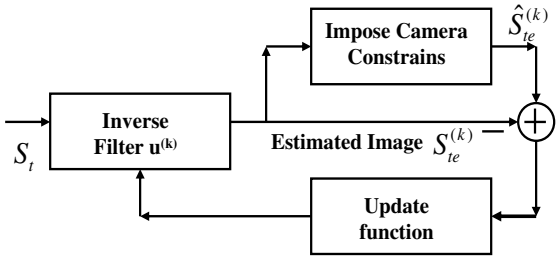

Figure 2. A recursive algorithm to estimate the coefficients of the manipulation block

In general, we may assume that $\sum_{m,n} u(m,n,c) = K$ for $c = 1,2,3$, where K is a constant. A value of $K = 1$ ensures that the original image and its manipulated version have similar brightness levels. Incorporating this constraint into the minimization problem, we solve for u by minimizing a modified cost function J given by

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

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

The filter coefficients can be obtained through a recursive procedure presented in Fig. 2. In the k^{th} iteration, we obtain an estimate of the camera output $S_{te}^{(k)}$ by passing the test image S_t though the estimate of the inverse manipulation filter $u^{(k)}$. We then impose camera constraints given by (4) to obtain $\hat{S}_{te}^{(k)}$ and find the camera model fitting error. The inverse filter coefficients are then updated by [17],

$$u^{(k+1)} = u^{(k)} + t_k d_k, \quad \text{where} \tag{6}$$

$$d_k = \begin{cases} -\nabla J(u^{(k)}) & \text{if k = 0,} \\ -\nabla J(u^{(k)}) + \beta_{k-1} d_{k-1} & \text{otherwise,} \end{cases} \tag{7}$$

$$\beta_{k-1} = \frac{< \nabla J(u^{(k)}) - \nabla J(u^{(k-1)}), \nabla J(u^{(k)}) >}{||\nabla J(u^{(k-1)})||^2}, \tag{8}$$

and the step sizes t_k are chosen as the one that minimizes $J(u^{(k)} + t_k d_k) \leq J(u^{(k)} + t d_k)$ for all t. The recursive procedure is repeated until convergence. It can be shown in theory that the optimization problem is convex and converges to a unique solution.

4. SIMULATION RESULTS AND DISCUSSIONS

In this section, we present experimental results to validate the performance of the proposed algorithms.

4.1. Applications to Identifying Image Manipulations

Most often, creating a realistic tampered image involves a series of post-camera processing operations such as filtering, compression, resampling etc. Such processing operations leave distinct intrinsic fingerprint traces in the final processed image, which when analyzed provide evidence to detect tampering. In this subsection, we study the performance of our proposed authentication techniques for different types of image manipulations and compare our results with existing work. For our study, we collect a total of 100 representative images from Canon Powershot A75. These images are captured under completely random conditions − different scenes, different lighting conditions, and compressed under different JPEG quality factors as specified by default values used in the camera. These images form our *camera data set*. These images were then processed to generate 22 different tampered versions per image as shown in Table 1 to obtain 2200 manipulated versions.

Table 1. Set of Tampering Operations Considered

Manipulation Operation	Parameters of the Operation	Number of Images
Spatial Averaging	Filter orders 3-11 in steps of 2	5
Rotation	Degrees 1, 5, 10, 20	4
JPEG compression	Quality factors $\{30, 40, \ldots, 80, 95\}$	7
Resampling	Scale factors 0.5–1.5	6
Total		22

We run the proposed blind deconvolution methods on all the images and compute the coefficients of the manipulation block in each case. Fig. 3 shows the variation of the modified cost function J given by (5) as a function of the number of iterations for a sample unmanipulated image. We notice that the cost function converges in 10 iterations to a value very close to zero. The final estimated inverse filter coefficients u for the green color channel are very close to a delta function, and the frequency response of the estimated manipulation filter is almost a constant as shown in Fig. 4(a). The corresponding estimated frequency response when tested with an image filtered with a 5×5 averaging filter is shown in Fig. 4(b). The similarity among the estimated and the actual coefficients (shown in Fig. 4(c)) suggests that the proposed techniques can efficiently estimate the inverse filter coefficients quite accurately, and this result justifies the performance of the the blind deconvolution algorithms.

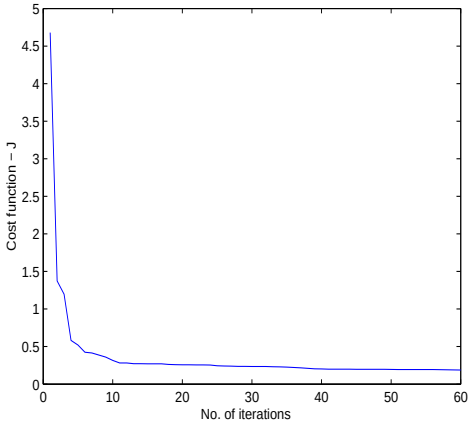

Figure 3. Convergence of the cost function for unmanipulated image

We design a *threshold based classifier* to distinguish direct camera outputs from manipulated ones. Given the test input S_t, we find the frequency domain coefficients of the manipulation filter H_t and compare it with the spectral response H_{ref} obtained from a reference direct camera output to measure the similarity among the coefficients. A similarity value greater than a chosen threshold indicates that the image satisfies the camera model. More specifically, we first find $LH_t = \log(H_t)$ and re-scale it to a $[0, 1]$ range to obtain the normalized logarithm of the frequency response Θ_t. The *similarity score* between the coefficients of the test input and the reference image is then found by comparing the corresponding normalized values

$$d(\Theta_t, \Theta_{ref}) = \sum_{m,n} (\Theta_t(m,n) - \mu_t) \times (\Theta_{ref}(m,n) - \mu_{ref}), \tag{9}$$

where μ_t denotes the mean of the Θ_t and so on. The test input is then classified as unmanipulated if the similarity to the reference pattern is *greater* than a suitably chosen threshold.

We examine the performance of the threshold based classifier in terms of the receiver operating characteristics (ROC). For each original image, we compute the frequency response of the equivalent manipulation filter and

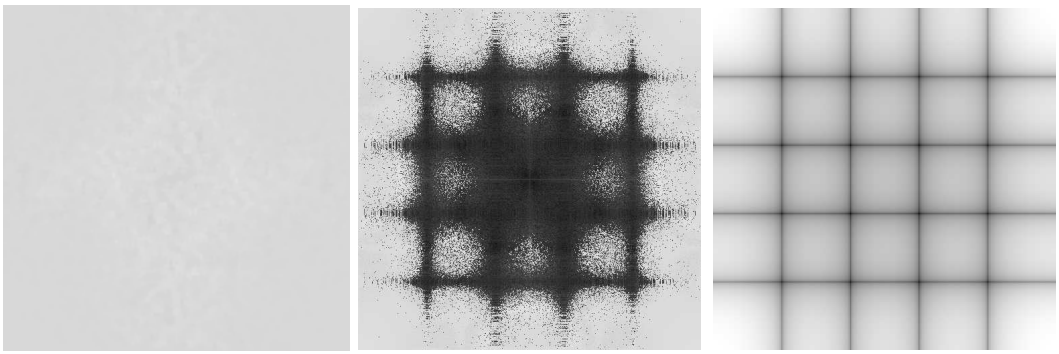

Figure 4. (a) Frequency response of the manipulation filter for an *unmanipulated camera output*. The figure shows that the frequency response is almost flat indicating that the corresponding impulse response is a delta function, (b) Estimated frequency response of the manipulation filter for an image low-pass filtered with a 5×5 averaging filter, (c) Actual frequency response of a 5×5 average filter shown for comparison. The frequency responses are shown in the log scale and appropriately scaled for display.

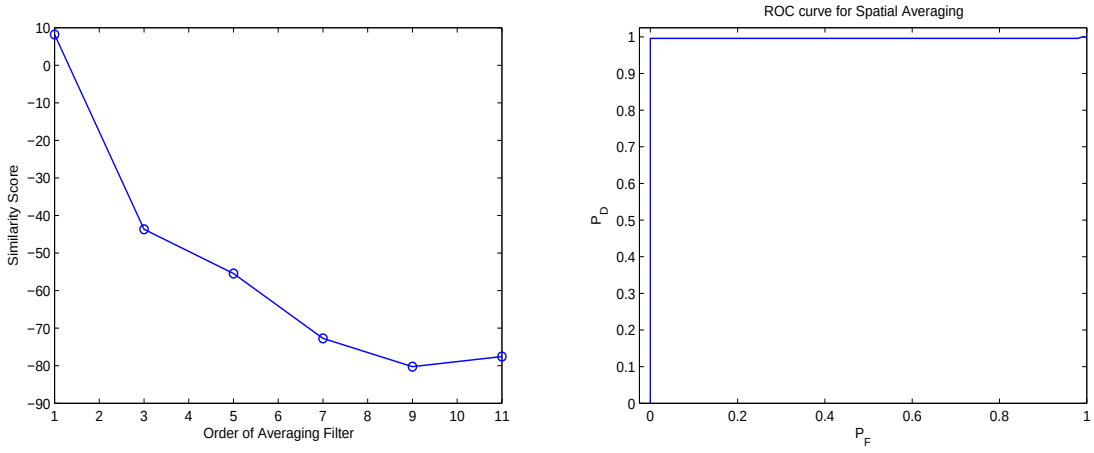

Figure 5. Performance results for Spatial Averaging (a) Variation of the similarity score as a function of the order of averaging filter, (b) Receiver Operating Characteristics.

measure its similarity with the reference filter pattern. The fraction of original images with a similarity score lower than a threshold η is found to give the false alarm probability P_F. Similarly, we record the fraction of manipulated images with a similarity score less than η to give the probability of correct decision P_D. We repeat this process for different decision thresholds η, and arrive at the ROC.

We test the performance under various types of manipulations in Table 1 and find the ROC in each case. Fig. 5(a) shows the average variation of the similarity score as a function of the order of the averaging filter. We observe that the similarity score reduces as the filter order increases. Further, the score is less than (-40) for all filtered images. This low value is because of the distinct nulls in the spectral response of the manipulated filter, estimated from filtered images, making it very different from the flat reference pattern. The ROC curve for filtering distortions is shown in Fig. 5(b), and it shows close to 100% classification accuracy with a P_D close to 1 for $P_F \approx 0.01$.

The performance of the threshold based classifier for image rotations is shown in Figs. 6(a) and 6(b). Rotations in image domain involves resampling the image on a rotated lattice followed by subsequent interpolation. Such manipulations can be equivalently modeled as a filtering operation with a significant linear component and its coefficients can be estimated using blind deconvolution. Thus, our proposed system model fits well for these distortions, and the detection algorithm efficiently identifies even minor degrees of rotation as demonstrated by

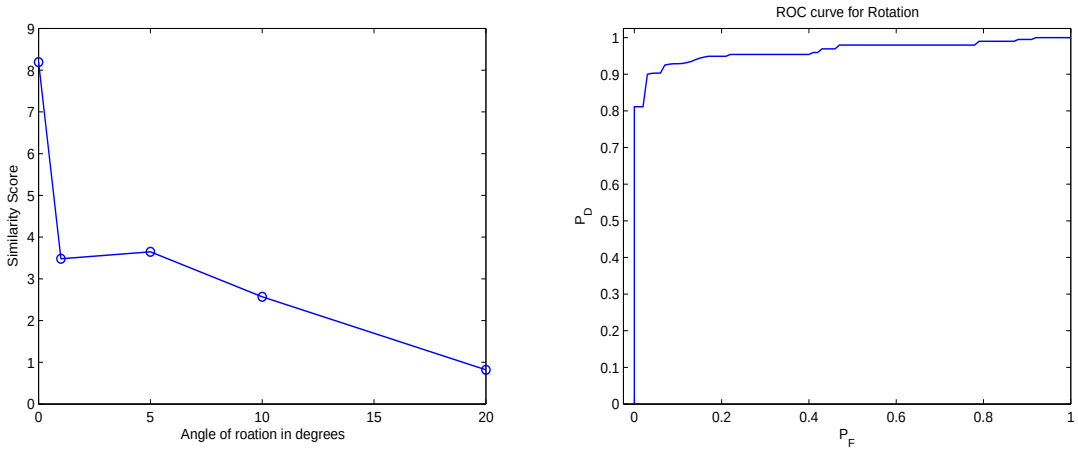

Figure 6. Performance results for Rotation (a) Variation of the similarity score as a function of angle of rotation in degrees, (b) Receiver Operating Characteristics.

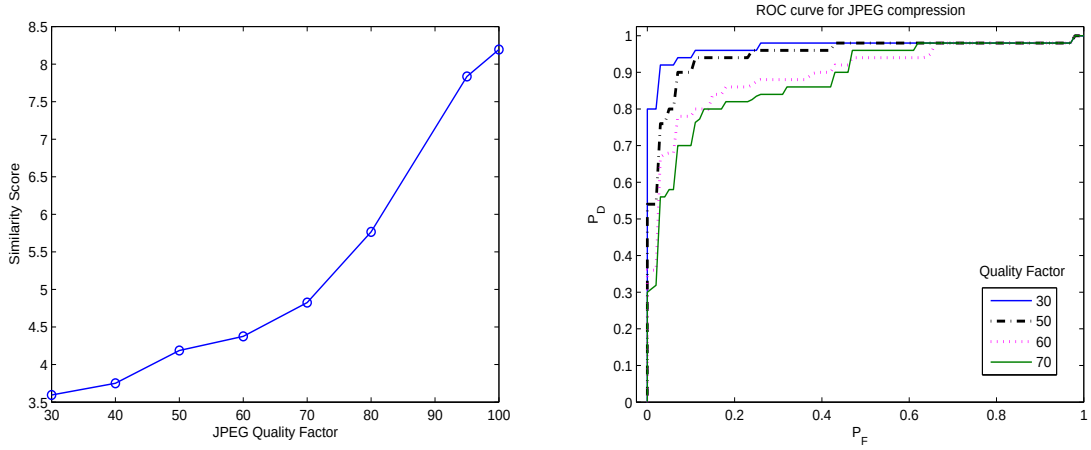

Figure 7. Performance results for JPEG compression (a) Variation of the similarity score as a function of JPEG Quality factor, (b) Receiver Operating Characteristics.

the superior performance in terms of the ROC curve shown in Fig. 6(b). The performance results for different levels of JPEG compression are shown in Figs. 7(a) and 7(b). We notice that as the quality factors decrease and the distortion levels increase, the similarity reduces and the performance improves. The results in Figs. 5-7 justify that most image manipulations can be analyzed by modeling it as a linear post-camera processing block, and by examining the similarity of its estimated coefficients with a reference pattern. Further, the value of the similarity score can also be used as a metric to measure and quantify the degree of distortion applied to the image.

We compare our proposed tampering detection algorithm with our implementation of the wavelet statistics based classifier approach presented in [18]. Here, the authors create a statistics vector by first extracting higher order moments from multiple level wavelet decompositions of the image. A linear predictor is then used to capture correlations that exist across orientation, space, and scale. An additional set of features is computed from the prediction error and the combined statistics vector is used in classification. In our experiments with [18], we calculate the statistics vector for the entire set of 100 original images and the corresponding 2200 manipulated ones. A support vector machine (SVM) with a radial basis function (RBF) kernel is used for classification [19]. We train the SVM with the data obtained from *ten* sample original images and its corresponding processed versions. From each manipulation type as listed in Table 1, we choose one processed version (for every original

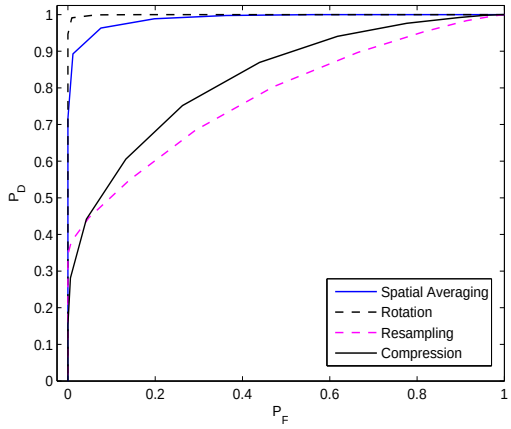

Figure 8. Performance of the Farid-Lyu's scheme [18] under various image manipulations when trained with ten sets of training images

image) with maximum amount of distortion for training the classifier, e.g., 11×11 spatial averaged image, 20 degrees rotated image, and so on. This gives us a total of 10 original images and 40 processed pictures in the training set. The remaining 90 images and its 22 manipulated versions are used in testing and the experiment is repeated 100 times by choosing a different training image each time. Fig. 8(a) shows the average performance of the Farid-Lyu's scheme. The results indicate that the proposed scheme can perform better that the method in [18], and can attain a higher P_D for the same P_F for distortions such as filtering, compression, and resampling. For the case of detecting image rotations, the performance of the proposed methods is still good, while [18] does better. Moreover, it is to be noted that while the SVM classifier in [18] requires processed images under all possible types of manipulations for training, it is not necessary for the proposed technique as the decision is made just by comparing the estimated coefficients from the test image with a reference pattern obtained from the camera output. Thus, the suggested method is more universal as it can more efficiently classify even tampering distortions that are not previously considered.

4.2. Applications to Blind Steganalysis

Image manipulations such as watermarking and steganography can be considered as post-processing operations applied to camera outputs. Such embedding algorithms leave behind statistical traces on the digital image that can be detected by analyzing the coefficients of the manipulation filter. We test the performance of the threshold based detector in distinguishing authentic camera outputs from stego data. In our experiments, we use the same camera data set with 100 color images of size 512×512 as our authentic set. Stego images are then generated by embedding random messages of different sizes into the cover images. Generally speaking, the maximum embedding capacity 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 capacity across 100 images and then embed messages at 100%, 75%, and 50% of this value.

For our study, we consider two popular steganographic embedding methods – Least Significant Bit (LSB) and spread spectrum. LSB embedding methods have been widely used for hiding data. Steghide [6] employs LSB embedding and a graph-theoretic approach to hide the secret messages on multimedia data. The message is hidden by exchanging rather than overwriting pixels. In our studies with steghide, we estimate the average maximum capacity across 100 color images to be around 32 KB for a 512×512 color image. The stego images are then generated by embedding secret messages of size 32 KB, 24 KB, and 16 KB using the software [20]. The detection results are shown in Fig. 9(a) 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 stego data close to 100% for false alarm probabilities around 1%. However, the performance reduces when the secret message length is reduced to 50% capacity at 16 KB. These results are better than the wavelet statistics based

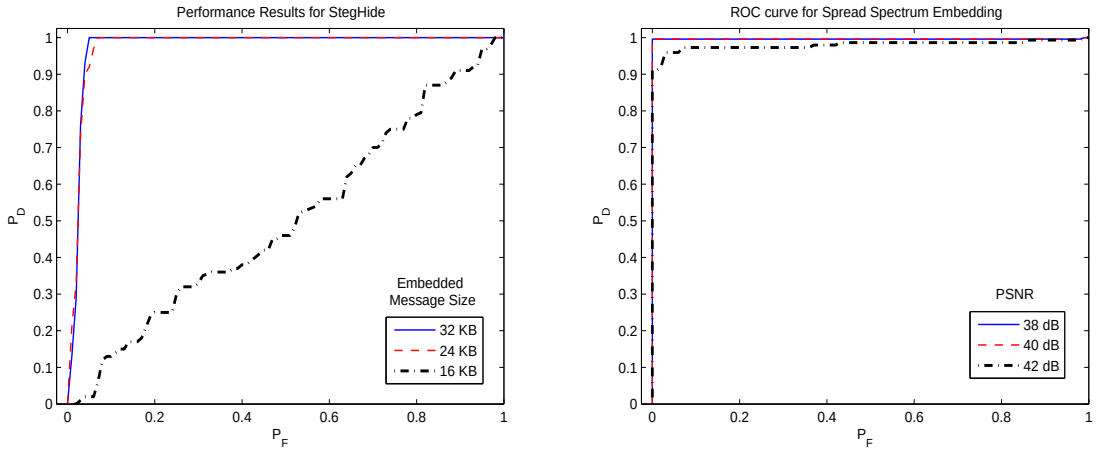

Figure 9. Performance results for (a) Steghide LSB embedding at different embedding rates, (b) Spread Spectrum embedding at different PSNR

steganalysis technique [12], which reports average accuracies of 77% and 60% at 100% and 78% embedding rates, respectively. While the results presented in [12] are for 32000 images with 30000 images used for training a SVM classifier with RBF kernel, our proposed methods employ only one reference pattern to distinguish between stego images and authentic camera outputs.

Next, we study the performance for spread spectrum embedding methods. Block-DCT based spread spectrum embedding have been widely used in literature for data hiding, watermarking, and steganography for a wide variety of applications. Detecting spread spectrum steganography has been a challenging problem, and statistics based schemes typically do not perform well in distinguishing original cover data and stego pictures. To our best knowledge, the only work that addresses spread spectrum steganalysis is [11], where they show that image quality metrics could 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 10 images. We test the performance of our authentication system for spread spectrum embedding. In our experiments, we use the same camera data set with 100 images of size 512×512 from four cameras as our authentic set. Stego images are then generated by adding pseudo-random watermarks at different peak signal-to-noise ratios (PSNR) of 38dB, 40dB, and 42dB. The manipulation block coefficients are estimated for the cover and the stego data, and classified with the threshold based classifier. Fig. 9(b) shows the performance results for different PSNRs. We note that the average identification accuracy is close to 100% for PSNRs of 38dB and 40dB, and reduces to 91% for 42dB PSNR at $P_F \approx 0$. These results demonstrate the superior performance of our proposed techniques.

5. CONCLUSIONS

In this work, we propose a set of forensic signal processing techniques to estimate the *intrinsic fingerprint* traces left behind in the digital image after it goes through various manipulation operations in *information processing* chain. We introduce a new formulation to study the problem of image authenticity. We characterize the properties of a genuine photograph by modelling the imaging process, and estimate its parameters by component forensic analysis. We consider any further post-camera processing as a manipulation block, and find the coefficients of its linear part using blind deconvolution. A high similarity of the estimated coefficients and the reference pattern, corresponding 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 post-camera processing, such as filtering, compression, resampling, rotation, etc. Evidence obtained from such forensic analysis is used to build a blind steganalyzer to determine the presence of hidden messages in multimedia data. Our results suggest that we can efficiently identify different types of embedding methods such as least significant bit (LSB)

and spread spectrum techniques with a high accuracy. Thus, our proposed techniques provides a more general framework for verifying the authenticity of images.

REFERENCES

1. A. Swaminathan, Y. Mao, and M. Wu, "Robust and Secure Image Hashing," *IEEE Trans. on Information Forensics and Security*, June 2006.
2. J. Lukas and J. Fridrich, "Estimation of Primary Quantization Matrix in Double Compressed JPEG Images," *Proc. of the Digital Forensics Research Workshop*, Cleveland, OH, Aug 2003.
3. H. Farid, "Blind Inverse Gamma Correction," *IEEE Trans. on Image Processing*, Vol. 10, No. 10, pp. 1428–1433, Oct 2001.
4. A. C. Popescu and H. Farid, "Statistical Tools for Digital Forensics," *Proc. of the 6th International Workshop on Information Hiding & Lecture Notes in Computer Science*, Vol. 3200, pp. 128–147, Toronto, Canada, May 2004.
5. A. C. Popescu and H. Farid, "Exposing Digital Forgeries in Color Filter Array Interpolated Images," *IEEE Trans. on Signal Processing*, Vol. 53, No. 10, pp. 3948–3959, Oct 2005.
6. S. Hetzl and P. Mutzel, "A Graph-Theoretic Approach to Steganography," *9th IFIP Conference on Communications and Multimedia Security*, CMS 2005, Springer LNCS 3677.
7. L. M. Marvel, C. G. Boncelet Jr., and C. T. Retter, "Spread Spectrum Image Steganography," *IEEE Trans. on Image Processing*, Vol. 8, No. 8, pp. 1075–1083, August 1999.
8. J. Fridrich, M. Goljan, and D. Hogea, "Steganalysis of JPEG Images: Breaking the F5 Algorithm," *Proc. of International Workshop on Information Hiding*, 2002.
9. J. Fridrich and M. Goljan, "Practical Steganalysis State of the Art," in *Proc. of SPIE Photonics West and Electronic Imaging, Security and Watermarking of Multimedia Contents*, Vol. 4675, pp. 1-13, San Jose, CA, Jan 2002.
10. A. Westfeld and A. Pfitzmann, "Attacks on Steganographic Systems," *Proc. of Information Hiding Workshop*, and *Lecture Notes in Computer Science*, pp. 61–76, 1999.
11. I. Avcibas, N. Memon, and B. Sankur, "Steganalysis Using Image Quality Metrics," *IEEE Trans. on Image Processing*, Vol. 12, No. 2, Feb 2003.
12. S. Lyu and H. Farid, "Steganalysis Using Higher-Order Image Statistics," *IEEE Trans. on Information Forensics and Security*, Vol. 1, No. 1, pp. 111–119, March 2006.
13. J. Adams, "Interaction between Color Plane Interpolation and other Image Processing Functions in Electronic Photography," *SPIE Cameras and Systems for Electronic Photography & Scientific Imaging*, pp. 144–151, San Jose, CA, Feb 1995.
14. A. Swaminathan, M. Wu, and K. J. Ray Liu, "Component Forensics of Digital Cameras: A Non-Intrusive Approach," *Proc. of the Conference on Information Sciences and Systems (CISS)*, Princeton, NJ, Mar 2006.
15. A. Swaminathan, M. Wu, and K. J. Ray Liu, "Non-Intrusive Component Forensics of Visual Sensors Using Output Images," *IEEE Trans. on Information Forensics and Security*, Mar 2007.
16. A. Swaminathan, M. Wu, and K. J. Ray Liu, "Image Tampering Identification Using Blind Deconvolution," *Proc. of the IEEE International Conference on Image Processing (ICIP)*, Atlanta, GA, Oct 2006.
17. D. Kundur and D. Hatzinakos, "A Novel Blind Deconvolution Scheme for Image Restoration using Recursive Filtering," *IEEE Trans. on Signal Processing*, Vol. 46, No. 2, pp. 375–390, Feb 1998.
18. H. Farid and S. Lyu, "Higher-Order Wavelet Statistics and their Application to Digital Forensics," *IEEE Workshop on Statistical Analysis in Computer Vision*, Madison, WI, June 2003.
19. C-C. Chang and C-J. Lin, *LIBSVM: A Library For Support Vector Machines: 2001*, software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.
20. S. Hetzl, *Steghide*, software available at `steghide.sourceforge.net`.