COMPONENT FORENSICS OF DIGITAL CAMERAS: A NON-INTRUSIVE APPROACH

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

Electrical and Computer Engineering Department, University of Maryland, College Park

ABSTRACT

This paper considers the problem of component forensics and proposes a methodology to identify the algorithms and parameters employed by various processing modules inside a digital camera. The proposed analysis techniques are non-intrusive, using only sample output images collected from the camera to find the color filter array pattern; and the algorithm and parameters of color interpolation employed in cameras. As demonstrated by various case studies in the paper, the features obtained from component forensic analysis provide useful evidence for such applications as detecting technology infringement, protecting intellectual property rights, determining camera source, and identifying image tampering.

1. INTRODUCTION

Digital imaging has experienced tremendous growth in recent decades. The resolution and quality of electronic imaging devices have been steadily improving, and digital cameras are becoming more popular every year. Digital images taken by various imaging devices have been used in a growing number of applications, from military and reconnaissance to medical diagnosis and consumer photography. Such rapid technological development and popularity has led to a growing amount of signal processing research devoted towards the security and protection of multimedia information. In this paper, we introduce *component forensics* as a new methodology for forensic analysis. The proposed techniques aim at identifying the algorithms and parameters employed by various components in a visual sensing device using its output data, and help answer a number of forensic related questions related to protecting intellectual property rights, discovering technology infringement, and identifying content tampering.

For centuries, intellectual property protection has played a crucial role in fostering technology innovation. Fierce competition in the electronic imaging industry has led to an increasing number of infringement cases filed in courts. According to the U.S. patent law, *infringement of a patent* consists of the unauthorized making, using, offering for sale, or selling any patented invention during the term of its validity. Patent infringement is usually difficult to detect, and even harder to prove in the court of law. A common way to perform infringement analysis is to examine the design and implementation of a product and look for similarities with what have been claimed in existing patents, through some type of reverse engineering. However, this approach could be very cumbersome and is often limited to the *implementation of the idea* rather than the *idea* itself, and thus might potentially lead to misleading conclusions [1]. Component forensics studies can detect patent infringement by obtaining forensic evidence about the algorithms employed in a digital imaging device, using only its output data.

Component forensics also has applications in detecting tampering, and in establishing the trustworthiness of imaging devices. With the fast development of tools to manipulate multimedia data, the integrity of both content and acquisition device has become particularly important when images are used as critical evidence in journalism, surveillance, and law enforcement applications. The process of creating a tampered object often involves combining parts of several images, each of which is obtained from a separate acquisition device. Forensic evidence about the digital cameras collected from different regions of an image in question can suggest possible discrepancies and thus help validate the authenticity of the data with regard to cut-and-paste forgery.

While a growing amount of signal processing research in the recent years has been devoted to the security and protection of multimedia information (e.g. through encryption, hashing, and data embedding), component forensics is still in its infancy. Related prior art on non-intrusive image forensics fall in two main categories. In the forgery detection literature, there are works targeted at finding the type of post processing operations that occurs after an image has been captured by a camera, such as resampling, JPEG compression, luminance and brightness change, lens distortions, and copy-paste operations [2]. However, these methods cannot be directly applied to identify the algorithms (and its parameters) employed inside the components of the digital camera.

A second group of prior art on non-intrusive image forensics concerns camera identification. A technique was developed recently to use noise patterns inherent to an image sensor as a unique identifier to each camera [3]. While useful in some forensic tasks, this approach does not provide information about the internal components and cannot be

Email contact: {ashwins, minwu, kjrliu} @eng.umd.edu.

used for identifying common features tied to the same camera models. Another blind-source approach employs statistics generated from visually similar images taken with different cameras to train classifiers for identifying the image origin [4]. Although good results were reported in distinguishing pictures taken in controlled scenarios, its ability to differentiate under diverse training sets and non-intrusive testing conditions needs further investigation.

In this paper, we propose a set of non-intrusive techniques for component forensics of digital cameras. We use the sample output images obtained from the camera to gather forensic evidence about the color filter array (CFA) pattern and the CFA interpolation algorithms employed in the digital camera. We show that the results obtained through such analysis can provide very useful clues to detect potential infringement and protect intellectual property rights. The parameters of the image acquisition device are also used to construct a robust camera identifier for determining from which make/type of camera an image was taken and to establish potential tampering.

The paper is organized as follows. The proposed system model and problem formulation are presented in Section 2. In Section 3, we present methods to identify the CFA pattern and the parameters of the interpolation algorithms. The simulation results are presented in Section 4, wherein we present several case studies and discuss applications of the proposed forensic methodology for analyzing infringement/licensing and detecting tampering. We extend component forensics as a general methodology applicable to several devices in Section 5, and conclude the paper in Section 6.

2. SYSTEM MODEL AND PROBLEM FORMULATION

Fig. 1(a) shows the image capture process in digital cameras. The light from the scene pass through the lens and the optical filters and is finally recorded by an array of charge coupled device (CCD) detectors. Most digital cameras use a color filter array (CFA) to sample the real-world scene. Some examples of CFA patterns are shown in Fig. 1(b). 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 where it is located [5]. The remaining pixel values are interpolated using the sampled data. CFA interpolation (or demosaicking) is an important step to maintain the quality of the final output image [6, 7]. After interpolation, the three images corresponding to the red, the green, and the blue components are passed through a postprocessing stage. In this stage, white balancing and color correction are done to remove unrealistic color casts so that objects that appear white in reality appear white in the photograph. Finally, the image may be JPEG compressed to reduce storage space.

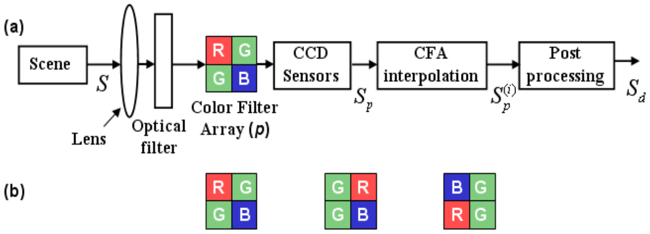

Fig. 1. (a) Imaging Process in Digital Cameras; (b) Sample Color Filter Arrays.

More specifically, let S be the real-world scene to be captured by the camera and let p be the CFA pattern matrix. $S(i,j,c)$ can be represented as a 3-D array of pixel values of size $H \times W \times C$. Here H and W represent the height and the width of the image, and $C = 3$ denotes the number of color levels (red, green, and blue). The CFA sampling converts the real-world scene S into a three dimensional matrix S_p of the form

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

After the data obtained from the CFA is recorded, the intermediate pixel values (corresponding to the points where $S_p(i,j,c) = 0$ in equation (1)) are interpolated using its neighboring pixel values to obtain $S_p^{(i)}$. This is an important step and certain types of residual errors caused in this step may be significantly amplified by subsequent processing [5]. There have been several algorithms for CFA interpolation, e.g. bilinear, bicubic, smooth hue, gradient based, etc. These algorithms can be broadly classified into two categories, non-adaptive and adaptive algorithms based on their ability to adjust to the image content [6].

The problem of component forensics deals with a methodology and a systematic procedure to find the algorithms and parameters employed in the various components inside the device. In this work, we consider the problem of non-intrusive forensic analysis where we use the sample images obtained from a digital camera under diverse and uncontrolled set up to determine the nature of its internal processing modules. In particular, we focus on finding the CFA pattern and the CFA interpolation algorithms, and show that these features can be used the first step in *reverse engineering* the making of a digital camera.

3. MODEL PARAMETER ESTIMATION

Proposed Algorithm We develop a robust and non-intrusive algorithm to jointly estimate the CFA pattern and the interpolation coefficients by using only the output images from cameras. Our algorithm estimates the CFA interpolation coefficients in each local image region through texture classification and linear approximation, and finds the CFA pattern that minimizes the interpolation errors.

More specifically, we establish the search space of CFA patterns based on common practice in camera design. For example, most commercial cameras use the RGB type of CFA with a fixed periodicity of 2×2, and each of the three types of sensors should appear at least once in each 2×2 cell. This results in a total of 36 possible patterns in the search space. For each CFA pattern p in the search space, we identify the locations of pixels in $S_p^{(i)}$ that are obtained directly from CCD sensors and those obtained by interpolation. The pixels values for the ones obtained directly from the CCD sensor can be found using $S_p(i,j,p(i,j)) = S_d(i,j,p(i,j))$.

We approximate the interpolation algorithms by multiple linear models that are linear in different regions of the image, where the regions are divided based on the gradient values in a local neighborhood. Defining $I_{i,j} = S_p(i,j,p(i,j))$, we can find the horizontal and vertical gradients at the location (i,j) by

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

The image pixel at location (i,j) is classified into one of the three categories: *Region* $\Re_1$ contains those parts of the image with a significant horizontal edge for which $(H_{i,j} - V_{i,j})$ is greater than a suitably chosen threshold T; *Region* $\Re_2$ contains those parts of the image with a significant vertical edge and is defined by the set of points for which $(V_{i,j} - H_{i,j}) > T$; *Region* $\Re_3$ includes the remaining parts of the image that are mostly smooth.

Using the output image S_d from camera, we establish a set of linear equations for all the pixels in each region and solve for the interpolation coefficients in each type of region. To cope with the noisy values in the equation set that are due to operations following the interpolation (such as JPEG compression), we employ a least squares method to estimate the interpolation coefficients. These coefficient estimates are then used to re-interpolate the sampled CFA output in the corresponding regions to obtain estimated interpolated image $\hat{S}_d^{(p)}$. The difference between $\hat{S}_d^{(p)}$ and the actual camera output image S_d is the interpolation error, denoted as $e^{(p)} = \hat{S}_d^{(p)} - S_d$. We compute the interpolation errors for all candidate search patterns and choose the pattern $\hat{p}$ that gives the lowest overall absolute value of error.

Finally, we remark that post-processing operations such as white balancing and color correction would not affect our estimates of the interpolation coefficients (and CFA sampling pattern). This is because these operations are multiplicative [5], and the same multiplicative factor would appear on both sides of the set of linear equations.

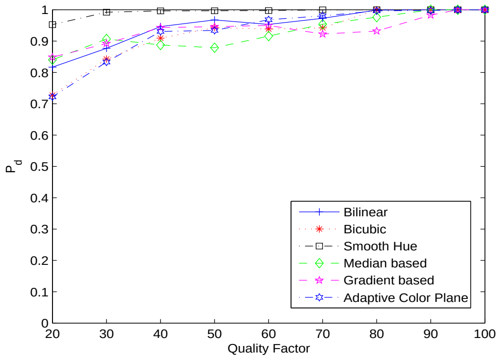

Fig. 2. Probability of correctly identifying the CFA interpolation technique for different JPEG quality factors

Experimental Results To study the effectiveness of the proposed algorithm, we use 20 representative images to first construct simulated data for various interpolation algorithms [8]. The images are first downsampled to remove the effect of previously applied filtering and interpolation operations, and then re-sampled using each of the three CFA patterns shown in Fig. 1(b). Each of the sampled images were then interpolated using one of 6 chosen interpolation methods: (1) Bilinear, (2) Bicubic, (3) Smooth Hue, (4) Median Filter, (5) Gradient based, and (6) Adaptive Color Plane (see [6] for details). Thus, our overall dataset contains $20 \times 3 \times 6 = 360$ images, each of size 512×512. Using the proposed algorithm, we observed no errors in estimating the CFA pattern. We then used a 7×7 neighborhood to estimate the interpolation coefficients for the three color components in the three regions. A Support Vector Machine (SVM) classifier with a third-degree polynomial kernel was used for identifying the interpolation method. We randomly choose 8 images out of the 20 images and use them for training; the remaining 12 images are for testing. We repeated the experiment for 100 times by choosing a different set of images each time. Fig. 2 shows the probability of correctly identifying the CFA interpolation technique for different JPEG quality factors. For moderate to high JPEG quality factors (40-100), the probability of correct identification is above 90%. These results suggest that the proposed estimation algorithm is not affected by the typical compression applied inside the camera after color interpolation.

4. CASE-STUDIES AND APPLICATIONS OF COMPONENT FORENSICS

In this section, we demonstrate the applications of the proposed component forensic methodology with case studies and experiments in camera identification, infringement/ licensing forensics, and tampering detection.

Table 1. Camera models used in simulation, ordered by camera make

	Camera Model		Camera Model
1	Canon A75	9	Sony Cybershot
2	Canon Powershot S400	10	Sony P72
3	Canon Powershot S410	11	Olympus C3100Z/C3020Z
4	Canon Powershot S1 IS	12	Olympus C765UZ
5	Canon Powershot G6	13	Minolta DiMage S304
6	Canon EOS Digital REBEL	14	Casio QV-UX2000
7	Nikon E4300	15	FujiFilm S3000
8	Nikon E5400	16	Epson PhotoPC 650

Table 2. Confusion matrix for classifying different camera makes. * denotes values smaller than 1%

	(C)	(N)	(S)	(O)	(M)	(Cs)	(F)	(E)
Canon (C)	98%	*	*	*	*	*	*	*
Nikon (N)	6%	85%	5%	3%	*	*	*	*
Sony (S)	3%	3%	93%	*	*	*	*	*
Olympus (O)	6%	6%	*	85%	*	*	*	*
Minolta (M)	2%	2%	4%	*	91%	*	*	*
Casio (Cs)	3%	*	*	5%	*	91%	*	*
Fuji (F)	*	*	*	*	3%	*	95%	*
Epson (E)	*	*	*	*	*	*	*	100%

4.1. Camera Identification Using Interpolation Features

We first demonstrate how to use the camera component analysis result as features to build a classifier to identify which camera has been used to capture a given image. We considered 16 different cameras as shown in Table 1, and collected 40 images taken by each camera in an uncontrolled environment. More specifically, the images taken by different cameras generally have different scene content and are compressed under default JPEG quality factors as specified by the cameras.

The CFA interpolation coefficients are estimated using the proposed algorithm and are used in classification. In our study, we considered each manufacturer as one class (which may consist of different models of the same manufacturer), and built an 8-class classifier. We randomly choose 25 images to train an SVM and then test on the remaining 15 images. This process is repeated 200 times and the average classification results are computed. We put together the results as a confusion matrix shown in Table 2. Here, the $(i,j)^{th}$ element in the confusion matrix gives the probability of being classified as camera make$-j$ when the picture actually comes from camera make$-i$. We notice that main-diagonal elements have an average value of 92.25% for eight camera makes. In comparison, the best performance in the literature that we can find so far is 84% on three makes [4]. We can see that camera identification based on the proposed component forensic framework can achieve considerably higher accuracy and better scalability to a large number of bands.

4.2. Similarities among Camera Models

The classification results in Table 2 also reveal some similarity between camera makes in handling interpolation. For example, 6% of Nikon cameras were classified as Canon and 5% as Sony make. As discussed earlier in the introduction, an important step toward infringement forensics is to analyze the similarity between the techniques and parameters employed inside cameras from different makes/models. In the following case study, we demonstrate how to use classification results from output images to quantitatively evaluate the similarities among the interpolation algorithms used by different cameras.

To construct classifiers, we start with 20 representative images, downsample them (by a factor of 2) and then reinterpolate with each of the 6 interpolation methods discussed in Section 3. With a total of 120 images synthetically generated in this way, we run the CFA estimator to obtain the estimated interpolation coefficients for each image. The estimated coefficients is then used to train a 6-class SVM classifier, where each class represents one interpolation method.

For each of the 40 images taken by every camera in the 16-camera dataset collected in Section 4.1, we estimate CFA parameters, feed them as input to the above classifier, and record the classification results. The average classification performance for camera$-i$ is represented by a vector $\underline{\pi_i} = [\pi_{i1}, \pi_{i2}, \ldots, \pi_{iN}]$, where π_{ik} is the fraction of images from camera$-i$ that have been classified as using the interpolation algorithm$-k$. Here, we have $1 \leq k \leq N$, with N being the number of possible choices of the interpolation algorithms studied. The similarities of the interpolation algorithms used by any two cameras (with indices i and j) can be measured in terms of a *divergence* score φ_{ij}, defined as symmetric *Kullback-Leibler* (KL) distance between the two distributions,

$$\varphi_{ij} \quad = \quad D(\pi_i || \pi_j) + D(\pi_j || \pi_i), \qquad (4)$$

$$\text{where } D(\pi_i || \pi_j) \quad = \quad \sum_{k=1}^{N} \pi_{ik} \log_2 \left(\frac{\pi_{ik}}{\pi_{jk}} \right). \qquad (5)$$

The symmetric KL distance is obtained in each of the regions $\Re_m$ $(m = 1,2,3)$ by separately training and testing the camera images in the selected region using the appropriately chosen features. The overall divergence score is obtained as the sum of the individual scores in each region. A low value of overall divergence score indicates that the two cameras are similar and are possibly using very similar kind of interpolation methods.

The divergence scores of the 16 different cameras are shown in Table 3. Here, the $(i,j)^{th}$ element in the matrix represents the symmetric KL distance between the interpolation algorithms in camera$-i$ and camera$-j$. The divergence scores below a threshold of 0.04 have been shaded for reference. We observe from the table that most cameras from the same make seem to be using similar kinds of interpolation algorithms.

To explore further we train a 15-class SVM using the interpolation coefficients obtained from 15 cameras (40 images per camera), and tested it with the data from the camera that was left out during training. In this case, the images from the excluded camera would be classified into its *nearest neighbor* class as measured in the feature space. When trained with all cameras except Canon S410, we observed that 72% of the 40 images from Canon S410 got classified as Canon S400, and an additional 25% were labelled as one of the remaining Canon models. A similar trend was also observed among the two Sony cameras, whereby 60% of the Sony Cybershot model were classified as Sony P72 model when the former was not used in training. These results suggest that there is a high degree of similarity in the interpolation algorithms employed by the the two cameras of the same make.

Another interesting aspect that we observe from Table 3 is that some divergence scores among cameras from different companies are low. For example, the distance between Nikon model E4300 (Camera no. 7) and the Olympus model C3100Z (Camera no. 11) is 0.03, suggesting a potentially high degree of similarity in how the two camera models perform color interpolation. By finding the parameters of the algorithms employed in various processing stages inside the digital camera, the proposed component forensics methodology can provide quantitative evidence on technology infringement/licensing.

4.3. Applications to Image Tampering

In creating a tampered image by cut-and-paste forgery, different parts of the image are often obtained from different cameras and therefore can be identified by finding the source that created each part of the image. In this section, we use a case study to demonstrate how the proposed component forensic algorithms can facilitate the detection of image tampering.

We first create a tampered picture by combining parts of two images taken with two different cameras. In Fig. 3(a) and (b), we show the tampered picture and its individual parts marked with different colors. The region displayed in white in Fig. 3(b) was obtained from an image taken with the Canon S410 digital camera, and the black parts were cropped and pasted from a picture shot using the Sony P72 model. The combined image was then JPEG compressed with quality factor 80%.

To identify the source of the individual components in the picture, we examine the image using a sliding window of 256×256 with step size 64, and apply to each 256×256 block our proposed camera identification algorithm based on CFA interpolation. The detection results from our 16-camera classifier are shown in Fig. 3(c). In this figure, the regions marked black denotes those classified as the Sony P72 model and the white areas correspond to the parts correctly classified as the Canon S410 model. The remaining regions represented in grey correspond to the blocks that were misclassified as one of the remaining 14 camera models. As shown in Fig. 3(c), our results indicate that we can identify the correct camera with a very high confidence in most of the regions in the tampered picture using the data obtained from each 256×256 macro-block. In this particular case, we notice that the manipulated picture has distinct traces from two different cameras and is therefore tampered.

Table 3. Divergence scores for different cameras. The set of values below the threshold of 0.04 have been shaded. The $*$ indicates zero divergence between the same camera models by definition. The camera index numbers are according to Table 1.

	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
01	*	0.06	0.22	0.50	0.65	0.96	0.82	1.93	1.23	1.59	0.53	0.36	0.16	0.68	0.02	1.03
02	0.06	*	0.02	0.10	0.16	0.38	0.46	0.98	0.56	0.76	0.18	0.11	0.36	0.18	0.23	0.36
03	0.22	0.02	*	0.13	0.20	0.32	0.39	0.98	0.57	0.81	0.21	0.18	0.38	0.12	0.17	0.43
04	0.50	0.10	0.13	*	0.02	0.09	0.07	0.50	0.18	0.31	0.02	0.04	0.37	0.18	0.37	0.10
05	0.65	0.16	0.20	0.02	*	0.04	0.06	0.33	0.10	0.21	0.03	0.09	0.46	0.18	0.47	0.05
06	0.96	0.38	0.32	0.09	0.04	*	0.12	0.18	0.06	0.13	0.11	0.23	0.76	0.16	0.76	0.03
07	0.82	0.46	0.39	0.07	0.06	0.12	*	0.37	0.07	0.15	0.03	0.09	0.47	0.42	0.62	0.05
08	1.93	0.98	0.98	0.50	0.33	0.18	0.37	*	0.11	0.06	0.48	0.71	1.46	0.61	1.59	0.16
09	1.23	0.56	0.57	0.18	0.10	0.06	0.07	0.11	*	0.02	0.14	0.28	0.82	0.43	0.97	0.01
10	1.59	0.76	0.81	0.31	0.21	0.13	0.15	0.06	0.02	*	0.27	0.44	1.09	0.58	1.29	0.06
11	0.53	0.18	0.21	0.02	0.03	0.11	0.03	0.48	0.14	0.27	*	0.02	0.31	0.29	0.38	0.08
12	0.36	0.11	0.18	0.04	0.09	0.23	0.09	0.71	0.28	0.44	0.02	*	0.17	0.37	0.24	0.20
13	0.16	0.36	0.38	0.37	0.46	0.76	0.47	1.46	0.82	1.09	0.31	0.17	*	0.84	0.10	0.69
14	0.68	0.18	0.12	0.18	0.18	0.16	0.42	0.61	0.43	0.58	0.29	0.37	0.84	*	0.57	0.32
15	0.02	0.23	0.17	0.37	0.47	0.76	0.62	1.59	0.97	1.29	0.38	0.24	0.10	0.57	*	0.80
16	1.03	0.36	0.43	0.10	0.05	0.03	0.05	0.16	0.01	0.06	0.08	0.20	0.69	0.32	0.80	*

A closer observation of the misclassified blocks (shown in grey) also indicates that most of these regions are clustered either around the tampering boundaries from two cameras or in very smooth areas of the image. Blocks around tampered regions would contain traces of both the camera models and thus might lead to incorrect classifications. The misclassification around the smooth regions of the image can be attributed to the fact that while most cameras may differ in their interpolation around edge regions, they employ similar techniques such as bicubic interpolation around the smooth regions. This suggests that tampering detection based on interpolation component would rely mostly on edge and texture regions rather than smooth regions.

5. COMPONENT FORENSICS METHODOLOGY

In this section, we extend the proposed non-intrusive forensic analysis to a methodology applicable to a more general family of digital devices. Let $O_1, O_2, \ldots, O_{N_o}$ be the sample outputs obtained from the test device (which we will model as a black box). Let $C_1, C_2, \ldots, C_{N_c}$ be the individual components of the black box and define $A_1^{(C_i)}, A_2^{(C_i)}, \ldots, A_{N_i}^{(C_i)}$ as the set of all possible algorithms/ techniques that be used in the component C_i. Forensic analysis is concerned with a set of tools that would help identify the individual al-

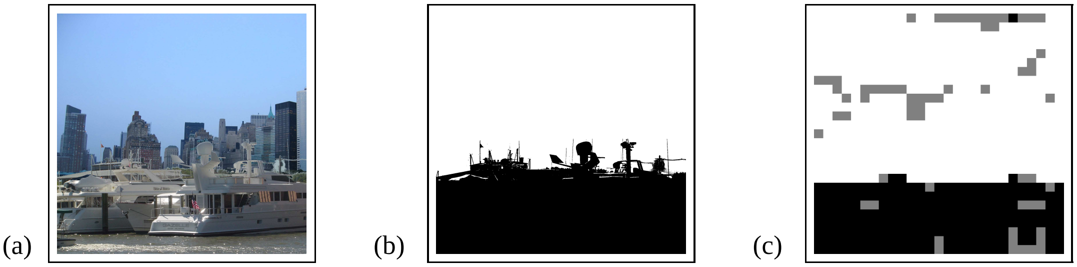

Fig. 3. Applications to source authentication showing (a) Sample tampered image; (b) Regions obtained from the two cameras; (c) CFA interpolation identification results (black: Sony P72; white: Canon S410; grey: regions classified as other cameras).

gorithms $A_x^{(C_y)}$ used in each of the processing blocks C_y. The proposed forensic analysis framework is composed of the following processing steps:

(1) Modeling of the Test Device: This is the first step of the proposed forensic analysis methodology. In this phase, a model is constructed for the object under study. The modeling helps break down the test device into a set of individual processing components $C_1, C_2, \ldots, C_{N_c}$ as seen in Section 2, and enables a systematic procedure to study the effect of each block on the final output obtained with the test device.

(2) Feature Selection: In this phase, a forensic analyst identifies a set of unique features $F_1, F_2, \ldots, F_{N_f}$ that can best differentiate the algorithms $A_x^{(C_y)}$ used in the component C_y. These features are based on the final output images and are chosen to uniquely represent each of the algorithms used. This would enable the forensic analyst to focus on a selected set of features to differentiate the algorithms. For example, in our work with digital cameras, we have used the CFA interpolation coefficients from multiple linear approximations as features.

(3) Feature Analysis and Information Fusion: We analyze the features obtained from the previous stage to obtain forensic evidence to meet specific application's needs. The appropriate analysis technique depend on the application scenario. In some cases, the features by themselves (without further processing) can be used to estimate the parameters of the model and provide forensic evidence. In other situations, the features might be an intermediate step and further processing (e.g. via classification etc.) would be required to extract valuable information.

(4) Testing and Validation Process: The validation stage uses test data with known ground truth to quantify the accuracy and performance of the forensic analysis system. It reflects the degree of success of each of the above processing stages and their combinations. Representative synthetic data obtained using the model of the test object can help provide ground truth to validate the forensic analysis systems and provide confidence levels on estimation. The results of this stage can also facilitate a further refinement of the other stages in the framework.

6. CONCLUSIONS

In this paper, we propose a non-intrusive framework of component forensics for digital camera. The proposed framework uses only the sample data acquired by a test camera to find the algorithms and parameters employed in several processing modules, such as the CFA sampling and the CFA interpolation blocks, inside the digital camera. We show through simulations that the proposed methods are robust to various kinds of postprocessing that might occur in the camera. The proposed techniques are then used to gather forensic evidence on real world data-sets captured with 16 different cameras. Measures for similarity are defined and elaborate case-studies are presented to elucidate the similarities and differences among several digital cameras. We show that the proposed methods can be used to identify camera model that is used to acquire a given image and to detect image tampering.

7. REFERENCES

[1] D. F. McGahn, "Copyright Infringement of Protected Computer Software: An Analytical Method to Determine Substantial Similarity," *Rutgers Comp. & Technical Law Journal*, Vol. 21, No. 1, pp. 88–142, 1995.

[2] A. C. Popescu and H. Farid, "Statistical Tools for Digital Forensics," *International Workshop on Info. Hiding*, 2004.

[3] J. Lukas, J. Fridrich, and M. Goljan, "Determining Digital Image Origin Using Sensor Imperfections," *Proc. of the SPIE, Security, Steganography & Watermarking of Multimedia Contents VII*, Vol. 5681, pp. 249–260, Jan 2005.

[4] M. Kharrazi, H. T. Sencar, N. Memon, and I. Avcibas, "Blind Camera Identification Based on CFA Interpolation," *Proc. of the IEEE Intl. Conference on Image Processing (ICIP)*, Vol. 3, pp. 69–72, Sep 2005.

[5] J. Adams, K. Parulski, and K. Spaulding, "Color Processing in Digital Cameras," *Proc. of IEEE*, Vol. 18, No. 6, pp. 20–30, Nov 1998.

[6] J. Adams, "Interaction between Color Plane Interpolation and Other Image Processing Functions in Electronic Photography," *Proc. of the SPIE, Cameras and Systems for Electronic Photography and Scientific Imaging*, Vol. 2416, pp. 144–151, Feb 1995.

[7] A. C. Popescu and H. Farid, "Exposing Digital Forgeries in Color Filter Array Interpolated Images," *IEEE Transactions on Signal Processing*, Vol. 53, No. 10, part 2, pp. 3948–3959, Oct 2005.

[8] A. Swaminathan, M. Wu, and K. J. R. Liu, "Non-Intrusive Forensic Analysis of Visual Sensors Using Output Images," *Proc. of the IEEE Intl. Conf. on Acoustic, Speech and Signal Processing (ICASSP)*, May 2006 (to appear).