COMPUTATIONALLY EFFICIENT DEMOSAICING FILTER ESTIMATION FOR FORENSIC CAMERA MODEL IDENTIFICATION

Xinwei Zhao and Matthew C. Stamm

Dept. of Electrical and Computer Engineering, Drexel University, Philadelphia, PA 19104, USA

ABSTRACT

Determining the make and model of an image's source camera is an important forensic problem. While significant research has been conducted towards developing new camera model identification algorithms, little research has focused on controlling the computational cost of these algorithms. This becomes an important issue if forensic algorithms are to be used in "big data" scenarios. In this paper, we propose a new approach for controlling the computational cost associated with the algorithm proposed by Swaminathan et al. that identifies an image's source camera using least squares estimates of its demosaicing filter. Through a set of experiments, we show that our algorithm is able to achieve a higher classification accuracy at a fixed computational cost than the existing method. Similarly, our algorithm is able to reach a target classification accuracy at a lower computational cost.

Index Terms— Information Forensics, Camera Model Identification, Demosaicing, Computational Efficiency

1. INTRODUCTION

Technological advances over the past several decades, such as the widespread availability of high-speed Internet service, has made it easy to share digital images throughout the world. However, since digital images are used in several settings such as news reporting, criminal proceedings, and military intelligence, determining or verifying their origin is an important task. While this information can be contained in an image's metadata, metadata is often missing from an image and can easily be falsified.

In response to this problem, information forensics researchers have developed a variety of techniques to determine the manufacturer and model of the camera that captured an image [1]. These techniques operate by extracting a set of forensic traces from an image that were left by it's source camera, then providing these traces as features to a machine learning algorithm trained to determine the model of the image's source camera. Forensic techniques have been developed to make use of a variety of traces left by an image's source camera such as information contained in JPEG headers [2], sensor noise statistics [3], and color processing traces left by an image's demosaicing algorithm [4, 5, 6, 7], as well as other statistical features [8, 9]. Other forensic techniques have been developed to determine if an image was created using a scanner [10, 11], computer generated [12, 10, 13], or acquired using compressive sensing [14, 15].

While significant research effort has been devoted to discovering new traces for performing camera model identification, little research has focused on the controlling the computational cost of these algorithms. By contrast, the number of digital images, and possible source cameras, has increased dramatically. Since virtually all of the aforementioned forensic techniques rely upon machine learning algorithms to perform source identification, the computational cost of training these algorithms grows dramatically as the number of possible camera sources increases. In order for camera model identification algorithms to process the large volume of images and possible source camera models associated with today's "big data" environment, it is necessary to control the computational costs associated with training and testing these forensic algorithms.

In this paper, we focus on reducing the computational cost associated with the camera model identification algorithm proposed by Swaminathan et al. [5]. This algorithm operates by modeling a camera's demosaicing algorithm as linear interpolation, obtaining a set of least squares estimates of the demosaicing filter, then using the estimated filter coefficients as features for a support vector machine (SVM) trained to distinguish between different source camera models. Since the computational cost of using all pixels in an image to estimate the demosaicing filter quickly becomes impractical, Swaminthan et al. proposed performing estimation using only pixels in a small window of the image. While this approach helps to some degree, the algorithm's overall performance varies significantly depending on which window is chosen, and results in a highly suboptimal trade-off between the computational cost and overall accuracy.

In this paper, we propose a new approach for controlling the computational cost associated with Swaminathan et al.'s camera model identification algorithm. In our approach, we identify a set of n pixels throughout the *entire image* that results in a better estimate (as measured by the Frobenius norm of the estimation error covariance matrix) of the demosaicing filter. By obtaining demosaicing filter estimates in this manner, we are able to greatly improve the trade-off between the classification accuracy and computational cost of Swaminathan et al.'s algorithm. Specificially, at a fixed computational cost, our algorithm results in a significantly higher camera model identification accuracy. Similarly, our algorithm significantly reduces the computational cost needed to achieve a particular minimum camera model identification accuracy.

2. BACKGROUND

A digital camera operates by measuring the intensity of light reflected from a real world scene onto an electronic sensor. As shown in Fig. 1, light enters a digital camera by first passing through its lens. Since most sensors are capable of capturing only one color component at one pixel location, the light next passes through an optical fil-

Email: { Xinwei Zhao, xz355@drexel.edu}

Email: { Matthew C. Stamm, mcstamm@drexel.coe,edu}

Research was sponsored by the U.S. Army Research Office and the Defense Forensics and Biometrics Agency and was accomplished under Cooperative Agreement Number W911NF-15-2-0013. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office, DFBA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

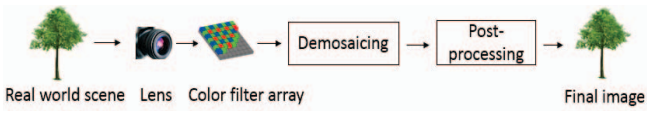

Fig. 1. Image processing pipeline in digital cameras

ter known as a color filter array (CFA), before hitting the sensor. The CFA consists of a repeating fixed pattern and allows only one color band of light (red, green, or blue) to fall incident on the sensor at a particular pixel location. The sensor then measures the light intensity of the color component corresponding to the CFA pattern at each pixel location, and yields an image constructed with three partially sampled color layers. Next, the unobserved color component values are demosaiced using nearby directly captured pixel values. After demosaicing, images may undergo further internal post-processing such as white balancing and compression before the final output image is produced.

2.1. Camera Model Identification

Since the performance of the demosaicing algorithm is critical for the quality of the final image, most camera manufactures employ proprietary demosacing algorithms. As a result, several forensic algorithms have been developed to determine the make and model of an image source camera using features pertaining to the demosaicing algorithm used to produce the image [16, 17].

One of the commonly used methods was proposed by Swaminathan et al. [16]. Since the ground truth demosaicing filters are adaptive to common gradient orientation and color bands, they approximate demosaicing filters for each color band and gradient orientaion to be linear.

First, they guess a CFA pattern, and reverse the demosacing process to obtain an image which is formed by retaining only directly captured color channel values. Next, they group pixel locations of the resulting image into horizontal, vertical, or smooth regions by comparing the absolute difference between horizontal and vertical gradient magnitudes of each pixel to a well-selected threshold. Next, for a particular color channel of the image, they model the unobserved pixel values as a linear combination of directly captured pixel values in its local neighborhood. As a result, they obtain 9 overdetermined systems of linear equations pertaining to each particular pairing of color channel and a gradient region. To form each overdetermined system of linear equations, let n be the number of demosaiced pixels collected from a particular color and gradient region. Let m be the number of directly captured pixels in a local neighborhood centered at one demosaiced pixel. The overdetermined system of equations can be expressed as,

$$\mathbf{y} = \mathbf{X}\boldsymbol{\theta}, \tag{1}$$

where $\mathbf{y} = [y_1, ..., y_n]^\top$ is a $n \times 1$ vector with each scalar entry representing a demosaiced pixel value, and $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_n]^\top$ is a $n \times m$ matrix with each vector entry $\mathbf{x}_i^\top = [x_{i,1}, ..., x_{i,m}]$ representing the directly captured pixels in a local neighborhood surrounding $y_i, i \in [1, n]$

The demosaicing filter estimates $\hat{\boldsymbol{\theta}}$ are obtained by solving (1) using least squares estimation. Using this approach, they estimate the demosaicing filter coefficients for each of the 9 sets of linear equations, and use the demosaicing filter estimates as features for a multiclass support vector machine trained to identify the model of the image's source camera.

3. TRADE-OFF BETWEEN COMPUTATIONAL COST AND ESTIMATION ACCURACY

The computational complexity of obtaining each least squares estimate of the $m \times 1$ demosaicing filter $\boldsymbol{\theta}$ using an $n \times 1$ observation vector $\mathbf{y}$ and data matrix $\mathbf{X}$ of size $n \times m$ is $O(nm^2)$ [18]. As a result, the computational cost of is directly proportional to the number of pixels used to perform estimation. For large images, the computational cost of using all pixels in an image to obtain these estimates quickly becomes impractical. To address this problem, Swaminathan et al. proposed using only pixels in a small window of the full image to obtain the set of demosaicing filter estimates. The window is heuristically chosen by dividing the image into a set of overlapping windows, then selecting the one with the highest total variation (i.e. the window with the largest sum of the magnitude of pixel gradients).

This problem brings to light a fundamental trade-off between classification accuracy and computational cost experienced by their algorithm. It is well known that the least squares estimator is a consistent estimator. As a result, increasing the number of pixels used to perform estimation tends to increase the accuracy of the demosaicing filter estimate. Since the demosaicing filter estimates are used as classification features by the SVM, better filter estimates tend to result in a higher camera model identification accuracy. Using more pixels to perform estimation, however, increases the size of the observation vector $\mathbf{y}$ and the data matrix $\mathbf{X}$, therefore increasing the computational cost.

While the window based approach allows this trade-off to be balanced by adjusting the window size, it has several important shortcomings. First, the performance of the classification algorithm is highly dependent upon the image window chosen for demosaicing filter estimation. While the total variation heuristic can produce good results, we have experimentally found its performance to be highly variable. Second, the set of pixels that yield the best estimate of θ may be spread throughout the entire image. The requirement that all pixels used for estimation fall within a particular window likely results in a suboptimal estimate. Third, the window-based approach does not allow for direct control over the computational cost of each filter estimate. This is because the pixels in each gradient region are chosen by comparing the magnitude of their horizontal and vertical gradients to a fixed threshold. For a given window size, the number of pixels in each gradient region can vary significantly from image to image depending on its content. As a result, the size of the data matrix, and thus computational cost, can vary significantly. This is discussed in further detail in Section 5.

4. PROPOSED APPROACH

We propose an alternative approach for obtaining the least squares estimate of the demosaicing filter that is designed to improve the trade-off between computational cost and estimation accuracy, and to address the other issues described above. Instead of performing estimation using a small window of the image, we propose using a subset of n pixels throughout the *entire image* that will yield a significantly better estimate of the demosaicing filter.

The accuracy of $\hat{\boldsymbol{\theta}}$, and thus the overall camera model identification accuracy, varies significantly based on which subset of N pixels are used as observations during estimation. We propose an algorithm to quickly identify a set of pixels that will achieve a much better estimate of the demosaicing filter at a fixed computational cost. To accomplish this, we formulate a stochastic model of an image's pixel values, then use this model to derive a lower bound on the Frobenius norm of the estimator's error covariance matrix. We then identify

the set of pixels that minimizes this lower bound, and use this to determine the set of pixels used during demosaicing filter estimation.

We begin by establishing a model relating each demosaiced pixel value to its local neighborhood of directly observed pixel values. Without loss of generality, we assume that we are examining only one color layer of an image so that each pixel has a one dimenisional value associated with it. Let $\mathcal{X}$ be the set of directly observed pixel values and $\mathcal{Y}$ be the set of interpolated pixel values. For every $y_i \in \mathcal{Y}$, we can write

$$y_i = \mathbf{x}_i^\top \boldsymbol{\theta} + \epsilon_i, \tag{2}$$

where $\mathbf{x}_i^\top = [x_{i,1}, \ldots, x_{i,m}]$ is the $w \times w$ neighborhood of directly observed pixels surrounding y_i and ϵ_i is modeling noise. We model $x_{i,j}$ as independent and identically distributed (i.i.d.) random variables distributed $\mathcal{N}(0, \sigma_i^2)$, and ϵ_i as independent Gaussian noise distributed $\mathcal{N}(0, \sigma_e^2)$. While in reality the pixel values in $\mathbf{x}_i$ are not zero mean, they can easily be made to be zero mean by simply subtracting the mean pixel value from both $\mathbf{x}_i$ and y_i. Furthermore, we assume that σ_i^2 may not equal σ_j^2 for two different interpolated pixels y_i and y_j.

For an arbitrary set of n pixels in $\mathcal{Y}$, we can group the equations relating the values of these pixels to their local neighborhoods together to form the overdetermined set of equations

$$\mathbf{y} = \mathbf{X}\boldsymbol{\theta} + \boldsymbol{\epsilon}, \tag{3}$$

where $\mathbf{y} = [y_1, ..., y_n]^\top$, $\mathbf{X} = [\mathbf{x}_1, \mathbf{x_2}, ..., \mathbf{x}_n]^\top$ and $\boldsymbol{\epsilon} = [\epsilon_1, .., \epsilon_n]^\top$. The least squares estimate $\hat{\boldsymbol{\theta}}$ of the demosaicing filter $\boldsymbol{\theta}$ can then be obtained using the pseudoinverse of the data matrix $\mathbf{X}$

$$\hat{\boldsymbol{\theta}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}. \tag{4}$$

To improve the trade-off between computational cost and estimation accuracy, we would like to form the observation vector $\mathbf{y}$ and its associated data matrix $\mathbf{X}$ by using the set of n pixels that yield the "best" estimate $\hat{\boldsymbol{\theta}}$ of the demosaicing filter. We define this estimate as the one which minimizes $||\boldsymbol{\Sigma}_{\hat{\boldsymbol{\theta}}}||$, where $\boldsymbol{\Sigma}_{\hat{\boldsymbol{\theta}}}$ is the covariance matrix of the estimation error and $||\cdot||$ denotes the Frobenius norm. The least squares estimator's error covariance matrix conditioned on the data matrix is given by the well known formula

$$\boldsymbol{\Sigma}_{\hat{\boldsymbol{\theta}}|\boldsymbol{X}} = \mathbf{E}[(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^\top \mid \mathbf{X}] = \sigma_e^2 (\mathbf{X}^T \mathbf{X})^{-1}. \tag{5}$$

The error covariance matrix can then be found by taking the expected value of (5)

$$\boldsymbol{\Sigma}_{\hat{\boldsymbol{\theta}}} = \mathbf{E}[\boldsymbol{\Sigma}_{\hat{\boldsymbol{\theta}}|\mathbf{X}}] = \sigma_e^2 \mathbf{E}[(\mathbf{X}^\top \mathbf{X})^{-1}], \tag{6}$$

Deriving an explicit formula for $\mathbf{E}[(\mathbf{X}^\top \mathbf{X})^{-1}]$ is very difficult due to the presence of the matrix inverse. Despite this, we are able to derive the following lower bound on the Frobenius norm of the covariance matrix

$$\|\boldsymbol{\Sigma}_{\hat{\boldsymbol{\theta}}}\| \geq \frac{m\sigma_e^2}{\sum_{i=1}^{n} \sigma_i^2}. \tag{7}$$

Our derivation of this lower bound is provided in the Appendix to this paper.

Using (7), we can see that this lower bound is inversely proportional to the sum of the variances of each pixel neighborhood. As a result, we can minimize this lower bound by forming the observation vector and its accompanying data matrix by choosing the y_i's whose local neighborhood of pixel values have the n largest variances σ_i^2.

While this approach results in a lower demosaicing filter estimation error and a higher camera model identification accuracy, estimating the variance of the directly observed pixels in the neighborhood of each demosaiced pixel adds a nontrivial computational cost. We note, however, that the variance of pixel values in the local neighborhood of pixels located on strong edges tends to be significantly higher than for other pixels. One reason for this is that the pixel values on either side of the edge tend to correspond to different colors. To avoid the cost of computing these local variances, we propose the heuristic of forming the observation vector from the set of demosaiced pixels with the largest edge strength as measured by the magnitude of their gradient. Furthermore, since different demosaicing filters estimates must be computed for each gradient region, this approach has the added benefit of making it easy to separate pixels into different regions according to their edge direction.

We can summarize our approach as follows. For each color channel:

1. Compute the magnitude and angular direction of the gradient of each pixel demosaiced pixel.
2. Divide the set of pixels into different gradient regions according to their angular orientation.
3. Obtain a least squares estimate of the demosaicing filter for each gradient region using the set of n pixels throughout the entire image with the largest gradient magnitudes.

   The resulting demosaicing filter estimates can then be used as classification

5. EXPERIMENTAL RESULTS

We performed a set of experiments to evaluate the performance of our proposed method for obtaining demosaicing filter estimates, and to demonstrate its benefits over the window-based approach. In order to conduct these experiments, we first built an experimental database of images captured using 13 different camera models. These models consisted of 9 cell phone cameras, 2 point-and-shoot cameras, and 2 digital SLRs. We manually captured 100 images using each camera to create an experimental database of 1300 images in total.

In our first experiment, we examined the ability of Swaminathan et al.'s window-based method to directly control the computational cost of estimating the demosaicing filter coefficients. As we noted in Section 3, the computational cost of obtaining the least squares estimate of the demosaicing filter is directly proportional to the length n of the data matrix (i.e. the number of pixels used during estimation). As a result, we use the length of the data matrix as a proxy for the computational cost. We began by identifying the 512×512 pixel window in each of the 1300 images with the highest total variation. Next, we chose a variety of different thresholds for dividing the window into different gradient regions, and recorded the size of the data matrix for each image at each threshold. Table 1 shows the mean length of each data matrix along with its standard deviation when the gradient threshold is chosen to be 0.5. As we can see from this table, the length of the data matrix, and hence computational cost, varies significantly from image to image. By contrast, our proposed approach allows the size of the data matrix to be directly specified.

In our second experiment, we characterized the trade-off of between the computational cost and camera identification accuracy of our proposed approach, and compared it to that of Swaminathan et al.'s window based method. We began by experimentally characterizing this trade-off for the window based method. Using Swaminathan et al.'s window-based method, we estimated the demosaicing filter coefficients for the 7 gradient thresholds shown in Table 2. We

Color, Gradient Direction	R, H	G, H	B, H	R, V	G, V	B, V
Mean	697	475	712	499	340	507
Standard Deviation	2358	1620	2410	1597	1105	1660

Table 1. Mean and standard deviation of the length of the data matrix using Swaminathan et al.'s window-based method using a gradient threshold of 0.5. R, G ,B represent the red, green, and blue channels respectively. H, V represent horizontal and vertical gradient regions respectively.

Threshold	0.5	0.3	0.1	0.05	0.01	0.005	0.001
Mean	538	2515	14738	29682	69796	75843	79270

Table 2. Mean of the total length of data matrix for each image using Swaminathan et al.'s method for 7 thresholds.

recorded the mean length of the data matrix (computed across all gradient regions) and used this to represent the computational cost. Next, we trained a support vector machine to perform camera model identification using these demosaicing filter estimates, and measured the classification accuracy using 5-fold cross validation. The classification accuracies achieved using the window-based approach for each data matrix length are shown in blue in Fig. 2.

Next, we repeated this experiment using our proposed method. To provide a fair comparison in terms of the computational cost, we used our method to estimate the demosaicing filter cofficients using each of the mean lengths of data matrix obtained in Table 2. We then trained a support vector machine to perform camera model identification using these demosaicing filter estimates, and measured the camera classification accuracy using 5-fold cross validation. The classification accuracy obtained using our method for each data matrix size is shown as the magenta line in Fig. 2.

As we can see from Fig. 2, our proposed method results in a significantly greater classification accuracy at a fixed computational cost. For example, for a data matrix length of $n = 2515$ per gradient region, our method achieves a classification accuracy of 71.3% accuracy. By contrast, Swaminathan et al.'s method achieves a classification accuracy 47.3%. This 24.0% gain in classification accuracy is marked using the green double arrow.

Similarly, Fig. 2 also shows that a given classification accuracy can be achieved by our algorithm at a much lower computational cost. For example, our proposed method needs a data matrix length of approximately $n = 15000$ to acheive a classification accuracy of 80%. By contrast, the window-based method requires a data matrix length of approximately $n = 30000$ equations to achieve the same classification accuracy. The decrease in computational cost acheived using our proposed method is marked using the red double arrow.

6. CONCLUSION

In this paper, we proposed a new method to improve the trade-off between computational complexity and accuracy for demosaicing filter coefficient estimation. Compared to Swainathan *et al.*'s method, our proposed method is able to achieve higher estimation demosaicing filter estimation accuracy, and thus higher camera model identification accuracy. Experimentally we demonstrated that at fixed computational cost, our proposed method is able to achieve a higher camera identification accuracy. Similarly, our algorithm is able to achieve a desired identification accuracy at a lower computational cost.

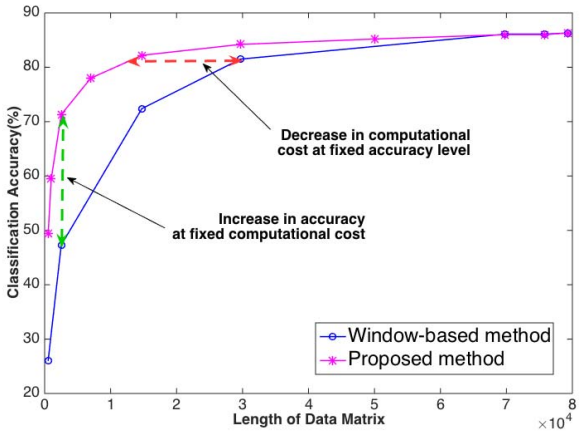

Fig. 2. Data matrix length vs. classification accuracy for our proposed method and Swaminathan et al.'s window-based.

7. APPENDIX

In this appendix, we provide our derivation of the lower bound on the Frobenius norm of the covariance matrix given in (7) in Section 4. We begin by noting that the matrix $\mathbf{X}^\top\mathbf{X}$ is positive semidefinite since it can be expressed as the following sum of positive semidefinite matrices

$$\mathbf{X}^\top\mathbf{X} = \sum_{i=1}^{n} \mathbf{x}_i\mathbf{x}_i^\top. \tag{8}$$

Because the inverse of a positive semidefinite matrix is a convex function, we can use Jensen's inequality to write the following lower bound on the expectation in the right hand side of (6)

$$\mathbf{E}[(\mathbf{X}^\top\mathbf{X})^{-1}] \geq (\mathbf{E}[\mathbf{X}^\top\mathbf{X}])^{-1}. \tag{9}$$

Next, we let $(\mathbf{X}^\top\mathbf{X})_{kl}$ denote the element at the k^{th} row and l^{th} column of $\mathbf{X}^\top\mathbf{X}$. This element can be expressed as

$$(\mathbf{X}^\top\mathbf{X})_{kl} = \sum_{i=1}^{n} x_{i,k}x_{i,l}. \tag{10}$$

Using this equation along with the fact the $x_{i,j}$'s are zero mean i.i.d. random variables, the entries of $\mathbf{E}[(\mathbf{X}^\top\mathbf{X})^{-1}]$ are given by

$$(\mathbf{E}[\mathbf{X}^\top\mathbf{X}])_{kl} = \begin{cases} 0 & \text{if } k \neq l, \\ \sum_{i=1}^{n}\sigma_i^2 & \text{if } k = l. \end{cases} \tag{11}$$

As a result, the diagonal elements of $(\mathbf{E}[\mathbf{X}^\top\mathbf{X}])^{-1}$ are all equal to $(\sum_{i=1}^{n}\sigma_i^2)^{-1}$ while the off-diagonal elements are all zero. Since the Frobenius norm of a matrix is equal to its trace (i.e. the sum of its diagonal elements), we can write the expression

$$\|(\mathbf{E}[\mathbf{X}^T\mathbf{X}])^{-1}\| = \frac{m}{\sum_{i=1}^{n}\sigma_i^2} \tag{12}$$

Finally, combining (6), (9), and (12), we can arrive at the following expression for the lower bound on the Frobenius norm of the error covariance matrix

$$\|\mathbf{\Sigma}_{\hat{\boldsymbol{\theta}}}\| \geq \frac{m\sigma_e^2}{\sum_{i=1}^{n}\sigma_i^2}. \tag{13}$$

8. REFERENCES

[1] M. C. Stamm, M. Wu, and K. R. Liu, "Information forensics: An overview of the first decade," *Access, IEEE*, vol. 1, pp. 167–200, 2013.

[2] E. Kee, M. K. Johnson, and H. Farid, "Digital image authentication from jpeg headers," *Information Forensics and Security, IEEE Transactions on*, vol. 6, no. 3, pp. 1066–1075, 2011.

[3] T. Filler, J. Fridrich, and M. Goljan, "Using sensor pattern noise for camera model identification," in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*. IEEE, 2008, pp. 1296–1299.

[4] S. Bayram, H. Sencar, N. Memon, and I. Avcibas, "Source camera identification based on cfa interpolation," in *IEEE International Conference on Image Processing 2005*, vol. 3, Sep. 2005, pp. III–69–72.

[5] A. Swaminathan, M. Wu, and K. Liu, "Non-intrusive forensic analysis of visual sensors using output images," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 5. IEEE, 2006, pp. V–V.

[6] H. Cao and A. C. Kot, "Accurate detection of demosaicing regularity for digital image forensics," *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 4, pp. 899–910, Dec. 2009.

[7] C. Chen and M. C. Stamm, "Camera model identification framework using an ensemble of demosaicing features," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, Nov. 2015, pp. 1–6.

[8] M. Kharrazi, H. Sencar, and N. Memon, "Blind source camera identification," in *Image Processing, 2004. ICIP '04. 2004 International Conference on*, vol. 1, Oct. 2004, pp. 709–712.

[9] F. Marra, G. Poggi, C. Sansone, and L. Verdoliva, "Evaluation of residual-based local features for camera model identification," in *New Trends in Image Analysis and Processing–ICIAP 2015 Workshops*. Springer, 2015, pp. 11–18.

[10] N. Khanna, G. T. C. Chiu, J. P. Allebach, and E. J. Delp, "Forensic techniques for classifying scanner, computer generated and digital camera images," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Mar. 2008, pp. 1653–1656.

[11] H. Gou, A. Swaminathan, and M. Wu, "Intrinsic sensor noise features for forensic analysis on scanners and scanned images," *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 3, pp. 476–491, Sep. 2009.

[12] T.-T. Ng, S.-F. Chang, J. Hsu, L. Xie, and M.-P. Tsui, "Physics-motivated features for distinguishing photographic images and computer graphics," in *Proceedings of the 13th annual ACM International Conference on Multimedia*. ACM, 2005, pp. 239–248.

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

[14] X. Chu, M. C. Stamm, and K. J. R. Liu, "Compressive sensing forensics," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 7, pp. 1416–1431, Jul. 2015.

[15] X. Chu, M. C. Stamm, W. S. Lin, and K. J. R. Liu, "Forensic identification of compressively sensed images," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Kyoto, Japan, Mar. 2012, pp. 1837–1840.

[16] A. Swaminathan, M. Wu, and K. Liu, "Nonintrusive component forensics of visual sensors using output images," *Information Forensics and Security, IEEE Transactions on*, vol. 2, no. 1, pp. 91–106, 2007.

[17] H. Cao and A. C. Kot, "Mobile camera identification using demosaicing features," in *Circuits and systems (ISCAS), Proceedings of 2010 IEEE international symposium on*. IEEE, 2010, pp. 1683–1686.

[18] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Johns Hopkins University Press, 1996, vol. 3.