Chapter 8

SOURCE CAMERA IDENTIFICATION USING SUPPORT VECTOR MACHINES

Bo Wang, Xiangwei Kong and Xingang You

Abstract Source camera identification is an important branch of image forensics. This paper describes a novel method for determining image origin based on color filter array (CFA) interpolation coefficient estimation. To reduce the perturbations introduced by a double JPEG compression, a covariance matrix is used to estimate the CFA interpolation coefficients. The classifier incorporates a combination of one-class and multi-class support vector machines to identify camera models as well as outliers that are not in the training set. Classification experiments demonstrate that the method is both accurate and robust for double-compressed JPEG images.

Keywords: Camera identification, CFA interpolation, support vector machine

1. Introduction

Sophisticated digital cameras and image editing software increase the difficulty of verifying the integrity and authenticity of digital images. This can undermine the credibility of digital images presented as evidence in court. Two solutions exist, watermarking and digital image forensics. Compared with the active approach of digital watermarking, digital image forensics [7, 12] is a more practical, albeit more challenging, approach. In a digital forensics scenario, an analyst is provided with digital images and has to gather clues and evidence from the images without access to the device that created them [15]. An important piece of evidence is the identity of the source camera.

Previous research on source camera identification has focused on detecting defective sensor points [5] and generating reference noise patterns for digital cameras [10]. The reference noise pattern for a digital camera is obtained by averaging over a number of unprocessed images. The

G. Peterson and S. Shenoi (Eds.): Advances in Digital Forensics V, IFIP AICT 306, pp. 107–118, 2009.

source camera corresponding to an image is identified using a correlator between the reference pattern noise and the noise extracted from the image. These methods suffer from the limitation that the analyst needs access to the digital camera to construct the reference pattern. Moreover, the reference pattern is camera-specific instead of model-specific.

Several methods have been proposed for identifying the source camera model. These methods primarily extract features from the digital image and use a classifier to determine image origin. The method of Kharrazi, *et al.* [8] uses image color characteristics, image quality metrics and the mean of wavelet coefficients as features for classification. Although this method has an average classification accuracy of nearly 92% for six different cameras, it cannot easily distinguish between cameras of the same brand but different models. The classification accuracy can be improved by combining the feature vector in [8] with the lens radial distortion coefficients of digital cameras [4]. However, extracting distorted line segments to estimate the distortion parameters limits the application of this method to images that contain distorted line segments. Meanwhile, good performance has been obtained by combining bi-coherence and wavelet features in a classifier [11].

Recently, several algorithms that use color filter array (CFA) interpolation coefficients have been developed. Most digital cameras use a number of sensors to capture a mosaic image, where each sensor senses only one color – red (R), green (G) or blue ((B). Consequently, a CFA interpolation operation called "demosaicking" is necessary to obtain an RGB color image. A variety of CFA interpolation patterns are used; the most common is the Bayer pattern. Bayram, *et al.* [1] employ an expectation-maximization algorithm to extract the spectral relationship introduced by interpolation to build a camera-brand classifier. Long and Huang [9] and Swaminathan, *et al.* [14, 15] have developed CFA interpolation coefficient estimation methods based on the quadratic pixel correlation model and the minimization problem. The best experimental results were obtained by Swaminathan, *et al.* [15], who achieved an average classification accuracy of 86% for nineteen camera models.

Most of the methods discussed above use Fisher's linear discriminant or support vector machine (SVM) classifiers. But these classifiers only distinguish between classes included in their training model – a false classification occurs when an item belonging to a new class is presented. Another problem is that the images for source camera identification often have double JPEG compression, which usually causes the methods discussed above to incorrectly classify the images.

This paper focuses on the important problem of identifying the camera source of double-compressed JPEG images. The method addresses the

difficulties posed by outlier camera model detection and identification. A classifier that combines one-class and multi-class SVMs is used to distinguish between outlier camera models. The image features use the covariance matrix to estimate the CFA interpolation coefficients used to accurately identify the source camera model. Experimental results based on sixteen different camera models demonstrate the robustness of the approach.

2. CFA Coefficient Features

A CFA interpolation algorithm, which is an important component of the imaging pipeline, leaves a unique pattern on a digital image. Such an algorithm is brand- and often model-specific. Consequently, CFA coefficients derived from an image can be used to determine image origin. An accurate estimation of the CFA coefficients improves classification accuracy. Our method applies the covariance matrix to reduce the negative impact of JPEG compression in the linear CFA interpolation model when the coefficients are estimated.

Practically every CFA interpolation algorithm interpolates missing RGB pixels in a mosaic image from a small local neighborhood. Thus, the interpolation operation can be modeled as a weighted linear combination of neighbor pixels in RGB channels [1, 14, 15]. For example, a missing G pixel $g_{x,y}$ is interpolated using an $n \times n$ neighborhood as:

$$g_{x,y} = \sum_{i=-n}^{n} \sum_{j=-n}^{n} w_g g_{x+i,y+j} \Big|_{\substack{except \\ i=0\&j=0}} + \sum_{i=-n}^{n} \sum_{j=-n}^{n} w_r r_{x+i,y+j} + \sum_{i=-n}^{n} \sum_{j=-n}^{n} w_b b_{x+i,y+j}$$

where w_g, w_r and w_b are the weighted coefficients in the interpolation.

The linear model can be expressed in vector form as:

$$p = [\vec{W}_r \quad \vec{W}_g \quad \vec{W}_b] * \begin{bmatrix} \vec{R} \\ \vec{G} \\ \vec{B} \end{bmatrix}$$

where p is the interpolated value and $\vec{R}$, $\vec{G}$ and $\vec{B}$ refer to the R, G and B pixel values, respectively, whose center is the interpolated pixel.

For an image with $M \times N$ resolution, each interpolation operation can be described as:

$$p_k = \sum_{l=1}^{3n^2-1} w_l s_{l,k}, \quad k \in [1, M \times N]$$

where s_{lk} denotes the $n \times n$ pixel values of the three channels except the k^{th} interpolated value, and w_l is the corresponding interpolation coefficient weight.

Equivalently, the vector expression $\vec{P} = \vec{W} * \vec{S}$ can be written as:

$$\begin{aligned}
\vec{P} &= \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_{M\times N} \end{bmatrix} \\
&= \begin{bmatrix} w_1 s_{1,1} + w_2 s_{2,1} + \ldots + w_{3n^2-1} s_{3n^2-1,1} \\ w_2 s_{1,2} + w_2 s_{2,2} + \ldots + w_{3n^2-1} s_{3n^2-1,2} \\ \vdots \\ w_1 s_{1,M\times N} + w_2 s_{2,M\times N} + \ldots + w_{3n^2-1} s_{3n^2-1,M\times N} \end{bmatrix} \\
&= \vec{W} * \vec{S}
\end{aligned}$$

JPEG compression is a common post-processing operation used in image storage that follows CFA interpolation in the imaging pipeline. An additional JPEG compression to reduce file size is commonly performed when an image is intended to be distributed over the Internet. A JPEG compression is lossy and alters the pixel values from the CFA interpolated results. To counter this, we introduce a term in each interpolation to model the perturbation introduced by single and double JPEG compressions:

$$\begin{aligned}
\vec{P}' &= \vec{P} + \vec{\delta} = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_{M\times N} \end{bmatrix} + \begin{bmatrix} \delta_1 \\ \delta_2 \\ \vdots \\ \delta_{M\times N} \end{bmatrix} \\
&= \begin{bmatrix} w_1 s_{1,1} + w_2 s_{2,1} + \ldots + w_{3n^2-1} s_{3n^2-1,1} + \delta_1 \\ w_2 s_{1,2} + w_2 s_{2,2} + \ldots + w_{3n^2-1} s_{3n^2-1,2} + \delta_2 \\ \vdots \\ w_1 s_{1,M\times N} + w_2 s_{2,M\times N} + \ldots + w_{3n^2-1} s_{3n^2-1,M\times N} + \delta_{M\times N} \end{bmatrix}
\end{aligned}$$

This can be written as:

$$\vec{P}' = w_1 \vec{S}_{1,k} + w_2 \vec{S}_{2,k} + \ldots + w_{3n^2-1} \vec{S}_{3n^2-1,k} + \vec{\delta}$$

where $\vec{S}_{l,k} = [s_{l,1} \quad s_{l,2} \quad \cdots \quad s_{l,M\times N}]'$, $l \in [1, 3n^2 - 1]$ is the vector of pixel values in the neighborhood of the interpolated location. In this formulation, we attempt to estimate all the interpolation coefficients w_l using the covariance between $\vec{P}'$ and $\vec{S}_{l,k}$:

$$\begin{aligned} cov(\vec{P}', \vec{S}_{l,k}) &= cov(w_1\vec{S}_{1,k} + w_2\vec{S}_{2,k} + \ldots + w_{3n^2-1}\vec{S}_{3n^2-1,k} + \delta, \vec{S}_{l,k}) \\ &= w_1 cov(\vec{S}_{1,k}, \vec{S}_{l,k}) + \ldots + w_{3n^2-1} cov(\vec{S}_{3n^2-1,k}, \vec{S}_{l,k}) \\ &\quad + cov(\vec{\delta}, \vec{S}_{l,k}) \end{aligned}$$

The JPEG compression is a non-adaptive method that is independent of the pixel values. Therefore, the perturbing term $\vec{\delta}$ is assumed to be independent of the coefficient vector $\vec{S}_{l,k}$ and, consequently, $cov(\vec{\delta}, \vec{S}_{l,k}) = 0$. The covariance reduces the negative impact of the JPEG and double JPEG compression. When l varies from 1 to $3n^2 - 1$, we construct the covariance matrix containing $3n^2 - 1$ linear equations, and the interpolation coefficients w_l are computed as:

$$\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_{3n^2-1} \end{bmatrix} = \begin{bmatrix} cov(\vec{S}_{1,k}, \vec{S}_{1,k}) & \ldots & cov(\vec{S}_{3n^2-1,k}, \vec{S}_{1,k}) \\ cov(\vec{S}_{1,k}, \vec{S}_{2,k}) & \ldots & cov(\vec{S}_{3n^2-1,k}, \vec{S}_{2,k}) \\ \vdots & \ddots & \vdots \\ cov(\vec{S}_{1,k}, \vec{S}_{3n^2-1,k}) & \ldots & cov(\vec{S}_{3n^2-1,k}, \vec{S}_{3n^2-1,k}) \end{bmatrix}^{-1} * \begin{bmatrix} cov(\vec{P}', \vec{S}_{1,k}) \\ cov(\vec{P}', \vec{S}_{2,k}) \\ \vdots \\ cov(\vec{P}', \vec{S}_{3n^2-1,k}) \end{bmatrix} \quad (1)$$

In the interpolation operation, pixels at different interpolated locations usually have different interpolation coefficients. Therefore, it is necessary to obtain the interpolation coefficients separately for the different pixel categories. In the case of the commonly used Bayer pattern, the eight missing color components in a 2×2 Bayer CFA unit are placed in seven categories. The two missing G components are grouped together in one category because of their symmetric interpolation pattern. The remaining six color components are placed in separate categories.

In each category, the interpolation coefficients are assumed to be the same and are computed using Equation (1). The interpolation neighborhood size is $n = 7$ in order to collect more information introduced by the interpolation algorithm while keeping the computation complexity reasonable. Each of the seven categories has $(3 \times 7^2 - 1) = 146$ interpolation

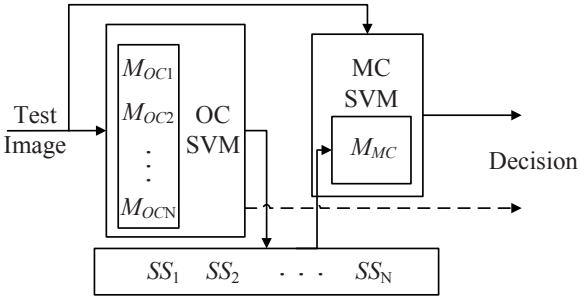

Figure 1. Combined classification framework.

coefficients. Therefore, the total number of interpolation coefficients is $(3 \times 7^2 - 1) \times 7 = 1,022$.

2.1 Combined Classification Framework

Several researchers have employed multi-class classifiers for camera identification [1, 4, 8, 11, 14]. The methodology involves extracting feature vectors from several image samples created by various camera models. The multi-class classifier is then trained by inputting feature vectors from sample images along with their class labels. After training is complete, the classifier is provided with the feature vector corresponding to a test image; classification is performed by assigning to the test image, the class label corresponding to the class that is the best match. The problem with this approach is that a multi-class classifier cannot identify outliers that do not belong to any of the classes in the original training set.

To address this issue we combine a one-class SVM [13] and a multi-class SVM [2]. The one-class SVM distinguishes outliers that do not correspond to any of the training camera models. If the one-class SVM associates a test image with multiple camera models, the multi-class SVM is used to determine the camera model that is the best match.

Figure 1 presents the combined classification framework. $M_{oc1}, M_{oc2}, \cdots, M_{ocN}$ denote the one-class models and $SS_1, SS_2, \cdots, SS_N$ denote the sets of image samples captured by N cameras. When the feature vector of a test image is extracted, all of the one-class models are first used to classify the test image. Each one-class SVM identifies the image as either being from the camera model it recognizes or an outlier to the model. Each positive result of a one-class SVM indicates that the test image may belong to one of the camera models. In general, there are three possible one-class SVM outputs for a test image:

Table 1. Camera models used in the experiments.

Camera Model	ID	Number of Images
Canon PowerShot A700	1	35
Canon EOS 30D	2	40
Canon PowerShot G5	3	33
Sony DSC-H5	4	35
Nikon E7900	5	35
Kodak Z740	6	35
Kodak DX7590	7	35
Samsung Pro 185	8	35
Olympus Stylus 800	9	35
Fuji FinePix F30	10	37
Fuji FinePix S9500	11	35
Panasonic DMC-FZ8	12	38
Casio EX-Z750	13	35
Minolta Dimage EX 1500	14	44
Canon PowerShot G6	15	37
Olympus E-10	16	31

1. **Outlier:** The test image has been created by an unknown camera outside the data set. In this case, the outlier camera can be exposed by the one-class SVM.

2. **One Positive Result:** The test image has been created by the camera model corresponding to the positive result.

3. **Multiple Positive Results:** The test image has been created by one of the camera models with a positive result.

For Cases 1 and 2, the final decision about image origin is made as indicated by the dashed line in Figure 1. For Case 3, the one-class SVM output is used to select image samples created by the camera models that give positive results for the test image. A new multi-class model M_{MC} is then trained using the selected image samples; this model is used to classify the test image as indicated by the solid line in Figure 1.

3. Experimental Results

Our experiments used a dataset containing images from sixteen different cameras over 10 brands (Table 1). The images were captured under a variety of uncontrolled conditions, including different image sizes, lighting conditions and compression quality. Each image was divided into 512×512 non-overlapping blocks of which eight were randomly chosen

Table 2. Confusion matrix for all sixteen cameras.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Outlier
1	**97.5**	*	*	*	*	*	*	*	*	*	*	*	*	*	*
2	1.3	**91.9**	*	*	*	*	1.3	*	2.5	*	*	*	*	*	*
3	1.9	*	**92.3**	*	*	*	*	*	*	*	*	*	*	*	*
4	*	*	*	**91.7**	*	*	*	*	*	*	1.6	*	*	*	*
5	*	*	*	*	**90.0**	*	*	1.6	*	*	*	*	2.5	*	*
6	*	1.6	*	*	*	**90.0**	2.5	*	*	*	*	*	*	*	*
7	*	1.6	*	*	*	2.5	**90.8**	*	*	1.6	*	*	*	*	*
8	*	*	*	*	1.6	*	*	**93.3**	*	*	*	*	*	*	*
9	*	1.6	*	3.3	*	*	*	*	**85.0**	2.5	*	3.3	*	*	*
10	*	*	*	*	*	*	*	*	1.5	**94.1**	2.2	*	*	*	*
11	*	*	*	*	1.6	*	*	*	*	3.3	**90.8**	*	*	*	*
12	*	*	*	1.4	*	*	*	*	1.4	*	*	**93.8**	*	*	*
13	1.6	*	*	*	1.6	*	*	*	*	*	*	*	**91.7**	*	*
14	*	*	*	*	*	*	*	*	*	*	*	*	*	**97.9**	*
15	10.8	*	11.5	*	1.4	*	*	*	*	*	3.7	*	*	*	**72.0**
16	*	2.0	*	*	*	*	5.7	*	20.2	*	*	*	1.2	*	**69.8**

for analysis. The image database consisted of 4,600 different images with 512×512 resolution. For each of the first fourteen camera models (IDs 1–14), 160 randomly chosen images were used for classifier training; the remainder were used for testing purposes. The remaining two cameras (IDs 15 and 16) provided 544 images that were used as outlier test cases.

One-class SVM and multi-class SVM implementations provided by LIBSVM [3] were used to construct the classifier. The RBF kernel was used in both SVMs. The kernel parameters were determined by a grid search as suggested in [6].

The experimental results are shown in Table 2 in the form of a confusion matrix. The fifteen columns correspond to the fourteen one-class training models and the outlier class. The sixteen rows correspond to the sixteen cameras used in the study. The (i, j^{th}) element in the confusion matrix gives the percentage of images from camera model i that are classified as belonging to camera model j. The diagonal elements indicate the classification accuracy while elements (15, 15) and (16, 15) indicate the classification performance for the outlier camera models. Values that are less than 1% are denoted by the symbol "*" in the table. The average classification accuracy is 92.2% for the fourteen cameras and 70.9% for the two outlier camera models.

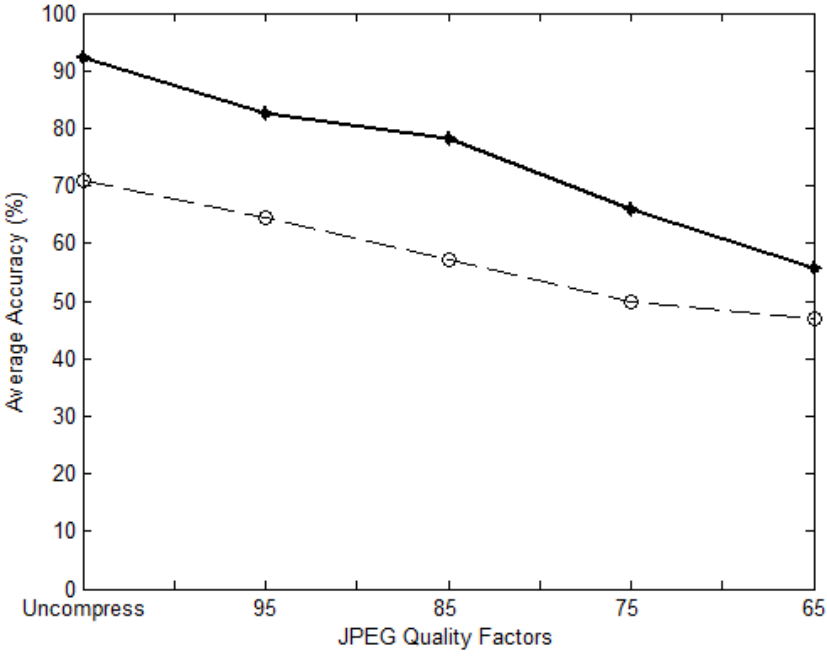

Figure 2. Average accuracy under different JPEG quality factors.

In order to test the classification method on double-compressed JPEG images, the images were re-compressed with secondary quality factors (QF) of {65, 75, 85, 95} without any other manipulations such as scaling and color reduction. Figure 2 presents the performance of the method for various quality factors for the double JPEG compression. The solid line shows the average detection accuracy for the fourteen camera models and the dashed line shows the average detection accuracy for the two outliers. For the quality factor of 95, the classification method provides an average accuracy of 82.5% for the fourteen camera models and 64.6% for the two outliers. For the quality factor of 65, the average accuracy drops to 55.7% for the fourteen cameras and 46.9% for the two outlier cameras.

Table 3 compares the results for the proposed method with those reported for the method of Meng, Kong and You [11]. The Meng-Kong-You method supposedly outperforms other source camera identification methods [11]. However, the results in Table 3 indicate that the Meng-Kong-You method cannot handle double-compressed JPEG images and is incapable of detecting outliers. On the other hand, the

Table 3. Average accuracy (double JPEG compression) for different quality factors.

	Meng-Kong-You Method					Our Method				
QF	**None**	**95**	**85**	**75**	**65**	**None**	**95**	**85**	**75**	**65**
1	91.5	61.7	58.7	49.9	50.1	97.5	94.2	91.7	73.3	63.3
2	87.9	61.3	57.2	52.3	47.2	91.9	87.5	81.9	67.5	51.9
3	89.7	60.4	59.5	52.3	50.9	92.3	79.8	74.0	63.5	57.7
4	89.1	64.0	57.8	51.0	49.0	91.7	84.2	78.3	62.5	51.7
5	91.2	62.5	58.8	50.5	47.6	90.0	78.3	73.3	59.2	48.3
6	90.5	65.0	57.4	52.5	49.9	90.0	77.5	75.8	63.3	55.8
7	89.0	62.6	56.8	52.1	48.8	90.8	79.2	75.8	60.0	54.2
8	86.8	62.4	56.2	51.2	48.1	93.3	85.8	79.2	66.7	45.8
9	90.8	64.5	56.3	52.1	49.9	85.0	73.3	69.2	57.5	49.2
10	88.9	62.9	58.7	51.3	49.3	94.1	87.5	81.6	76.5	59.6
11	89.8	61.3	56.8	49.7	50.6	90.8	78.3	75.8	70.0	55.0
12	90.7	63.7	58.0	49.4	51.7	93.8	83.3	79.2	68.1	61.1
13	91.3	64.5	56.1	49.7	48.7	91.7	75.0	71.7	62.5	58.3
14	90.4	60.4	58.8	50.7	51.1	97.9	91.1	88.0	71.9	67.2
Av.	**89.8**	**62.7**	**57.7**	**51.0**	**49.5**	**92.2**	**82.5**	**78.3**	**65.9**	**55.7**
15	-	-	-	-	-	72.0	65.5	60.5	50.3	47.0
16	-	-	-	-	-	69.8	63.7	54.0	49.6	46.8
Av.	-	-	-	-	-	**70.9**	**64.6**	**57.3**	**50.0**	**46.9**

proposed method is robust against double JPEG compression and can detect training model outliers with reasonable accuracy.

4. Conclusions

This paper has described a new method for determining the source camera for digital images. A covariance matrix is used to obtain a feature vector of 1,022 CFA interpolation coefficients. The feature vector is input to a classifier that is a combination of one-class and multi-class SVMs. The classifier can identify camera models in the training set as well as outliers. Experiments indicate that average accuracies of 92.2% and 70.9% are obtained for camera model identification and outlier camera model identification, respectively. The experiments also demonstrate that the method exhibits good robustness for double-compressed JPEG images.

Acknowledgements

This research was supported by the National High Technology Research and Development Program of China (Program 863; Grant No. 2008AA01Z418) and by the National Natural Science Foundation of China (Grant No. 60572111).

References

[1] S. Bayram, H. Sencar and N. Memon, Improvements on source camera model identification based on CFA interpolation, in *Advances in Digital Forensics II*, M. Olivier and S. Shenoi (Eds.), Springer, Boston, Massachusetts, pp. 289–299, 2006.

[2] B. Boser, I. Guyon and V. Vapnik, A training algorithm for optimal margin classifiers, *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pp. 144–152, 1992.

[3] C. Chang and C. Lin, LIBSVM: A Library for Support Vector Machines, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan (www.csie.ntu.edu.tw/∼cjlin/libsvm).

[4] K. Choi, E. Lam and K. Wong, Automatic source camera identification using intrinsic lens radial distortion, *Optics Express*, vol. 14(24), pp. 11551–11565, 2006.

[5] Z. Geradts, J. Bijhold, M. Kieft, K. Kurosawa, K. Kuroki and N. Saitoh, Methods for identification of images acquired with digital cameras, *Proceedings of the SPIE*, vol. 4232, pp. 505–512, 2001.

[6] C. Hsu, C. Chang and C. Lin, A Practical Guide to Support Vector Classification, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan (www.csie.ntu.edu.tw/∼cjlin/papers/guide/guide.pdf), 2003.

[7] N. Khanna, A. Mikkilineni, A. Martone, G. Ali, G. Chiu, J. Allebach and E. Delp, A survey of forensic characterization methods for physical devices, *Digital Investigation*, vol. 3(S1), pp. 17–18, 2006.

[8] M. Kharrazi, H. Sencar and N. Memon, Blind source camera identification, *Proceedings of the International Conference on Image Processing*, vol. 1, pp. 709–712, 2004.

[9] Y. Long and Y. Huang, Image based source camera identification using demosaicking, *Proceedings of the Eighth IEEE Workshop on Multimedia Signal Processing*, pp. 419–424, 2006.

[10] J. Lukas, J. Fridrich and M. Goljan, Detecting digital image forgeries using sensor pattern noise, *Proceedings of the SPIE*, vol. 6072, pp. 362–372, 2006.

[11] F. Meng, X. Kong and X. You, A new feature-based method for source camera identification, in *Advances in Digital Forensics IV*, I. Ray and S. Shenoi (Eds.), Springer, Boston, Massachusetts, pp. 207–218, 2008.

[12] T. Ng and S. Chang, Passive-blind image forensics, in *Multimedia Security Technologies for Digital Rights*, W. Zeng, H. Yu and C. Lin (Eds.), Academic Press, New York, pp. 383–412, 2006.

[13] B. Scholkopf, A. Smola, R. Wiliamson and P. Bartlett, New support vector algorithms, *Neural Computation*, vol. 12(5), pp. 1207–1245, 2000.

[14] A. Swaminathan, M. Wu and K. Liu, Component forensics of digital cameras: A non-intrusive approach, *Proceedings of the Fortieth Annual Conference on Information Sciences and System*, pp. 1194–1199, 2006.

[15] A. Swaminathan, M. Wu and K. Liu, Non-intrusive component forensics of visual sensors using output images, *IEEE Transactions on Information Forensics and Security*, vol. 2(1), pp. 91–106, 2007.