Using Multi-step Transition Matrices for Camera Model Identification

Shang Gao, *Rui-Min Hu and Gang Tian

NationalEngineering Research Center for Multimedia Software, Wuhan, China
email.nancy.g@gmail.com, hrm1964@public.wh.hb.cn,
tiangang_computer@163.com

Abstract

Recently, camera model identification becomes one of the most popular research topics in digital forensics field. Since every camera imaging processing left artifacts on its final output image, and some of them can be considered as model-specific 'traces' of its source camera, camera model can be classified only with a single image by catching these 'traces'. This paper presents a camera model identification method based on multi-step transition matrices. We firstly model JPEG image coefficients by Markov process. Then, one-step and two-step transition matrices along different directions are extracted respectively. Finally, 58 statistics calculated from these matrices are used to perform camera model identification as features. In our experiment, we chose images from seven camera models in Dresden Image Database as our experiment samples. Experiments results show that the average detection accuracy of this method can reach to 99.27%. Compared with previous Markov method, our approach can perform better only using 58-D features.

Keywords: *Camera-model classification, Multi-step Transition Matrices, SVM*

1. Introduction

Today is information age, along with the rhythm of life change and visual information of development, design of digital penetration of life are everywhere. Analog multi-media format is replaced by digital counterpart rapidly because of its low-cost, real-time and easy to store, design, transmit, duplicate, and transfer. At the same time, it also accelerates communication and transmission of knowledge and information. These excellent characteristics make more and more digital multi-media come into our life, such as digital camera, video surveillance, etc. These applications entertain us and also make it possible for us to collect digital copy version record of our history and life. However, every coin has its two sides. Except for its convenience, they also can be easily changed, even faked. As the easily obtained record of our life and behavior, digital multi-media should be a proper evidence for legal dispute. But since people can distort, attack and pirate them maliciously, these digital records become less credible. For the purpose of court usage, evidence is needed to be proved as authentic; the source and content of it needs to be credible. Hence, source camera model identification is developed to address the problem of image source verification, and it is also the topic we will discuss in this paper.

In today's digital forensic field, there are two approaches of camera source identification. One is to identify various camera brands and models; the other is to recognize individual properties in each camera.

For camera model identification, the classification always is achieved by finding out the difference of hardware component or digital image processing (DIP) technologies which vary from camera-model to model. To find out these differences, the detection works often focus on some special processing procedures, such as optical distortions in lens systems, sensor resolutions, CFA patterns and interpolation algorithms, quantization process, and other post processing, etc.

Some associated works have been published in recent years. For instance, M. Kharrazi et al. [1] did camera model identification by machine learning. They first extract three kinds of statistics from images, such as color features, wavelet statistics and image quality metrics, and then combine them as the input of SVM classifier for camera model identification. It can achieve 97% correct detection rate on four different cameras. But for different modes in one brand, like 5 cameras with 3 same brands, result drops to 88%. K. S. Choi et al. [2] also use machine learning to do model detection. They use radial distortion parameters as input of SVM for classification. But it is highly affected by focal length of lens. There are many proposed detection methods focus on CFA and interpolation in camera image processing. J. Lucas et al. [3], M. Chen et al. [4] and Chang-Tsun Li [5] average multiply noise images, which is extracted from images by wavelet de-noising filter, to build a reference fixed noise pattern. The identification is achieved by calculating correlation between test noise image and reference pattern. This method is sensitive to geometrical transformation, such as re-sample, crop, etc. In [6-9], the demosaicing algorithm in camera is modeled as a linear filter. The weights of the linear filters are estimated by EM algorithm after considering various CFA patterns, the estimated weights and estimated error are combined to be features. The result is good but also costs lots of time for computation. Guan-Shuo Xu et al. [10] proposed a camera model identification algorithm by extracting the Markov transition probability matrix as features, and the method can achieve 92.5% correct detection rate under multi-models. Beyond the above mentioned approaches, some specific applications are presented in this field in recent years, such as O. Celiktutan et al. [11] combined several previous forensics features as input of SVM classifier to identify source cell phones. E. Dirik et al. [12] used dust model to discriminate individual DSLR cameras. In this paper, a camera-model detection method is proposed. We present 58-D features to capture the artifacts introduced by the whole camera inside imaging processing. The 'Dresden Image Database' is used as our experiment database. The results show that our method can perform camera-model detection well with high detection accuracy even using small size of features.

This paper is organized as follow: Section 2 describes camera model identification based on Markov model. Section 3 introduces the features we propose in this paper, and also includes feature extraction steps. The experiment and results are presented in section 4. Section 5 concludes this paper. Section 6 is acknowledgements.

2. Camera Model Identification based on Markov Model

Camera-model identification can be achieved by finding out the difference of hard ware component and digital image processing (DIP) technologies which vary from camera-model to model. To find out these differences, the detection works often focus on some special processing procedures, such as optical distortions in lens systems, sensor resolutions, CFA patterns and interpolation algorithms, quantization process, and other post processing, etc. As can be seem from above section, there are two popular approaches of camera model identification in today's digital forensics field. One approach consider difference which only caused by single hardware component or single DIP technology; the other consider the impact brought by whole camera-imaging. For the second approach, the statistics, which are sensitive to camera-model type, always be extracted as features for input of classification. And this approach is also concluded as method of nature image statistics. For camera-model identification, several kinds of statistics were already proposed, such as color features, binary similarity measures, image quality metrics, wavelet domain statistics, moments in high frequency, Markov matrix, etc. Among these statistics, statistics based on Markov matrix performances good in classification. Previous work only uses one-step Markov processing and four difference 2-D JPEG arrays along four directions [10]. Considering more hypothesis possibility will be more comprehensive for classification issue, it can be expected that the detection accuracy can also increase by more hypothesis conditions.

In this paper, we will propose a camera-model identification method based on multi-step Markov process under different direction conditions. To reduce algorithm complexity, a threshold-ing and averaging technology will be used. The experiment results will show that our method can do camera-model detection with lower dimension features and higher detection accuracy than previous method.

The following section will firstly define two kinds of difference JPEG 2-D array along different directions, and then proposed to model the difference JPEG 2-D array using Markov random process. According to the theory of random process, the Markov process can be modeled by transition probability matrix, which will be combined to generate our proposed features. To get better identification performance, we extract features from the composition of one-step and two-step transition probability matrices derived from difference JPEG 2-D arrays along multi-directions. Simultaneously, threshold-ing technology and averaging processing will be used here to rapidly decrease feature size, and carefully keep balance between detection accuracy and computational complexity.

2.1. Difference JPEG 2-D Array

As the final output of JPEG image, the quantized block DCT coefficient of image carries all the related information we can get, include ‘digital scene content’, and the ‘potential detail information referred to model identification’. Therefore, our feature generation will work on JPEG quantized block DCT coefficient without any de-quantization. However, scene content varies from image to image, and it’s also the most ‘strongest’ information within an image. If we extract statistical features directly from quantized block DCT coefficient of an image, it is more possible for us to get the statistics of ‘scene content’ rather than ‘potential model information’. To weaken the impact of ‘scene content’, the common solution is extracting statistics from the difference between original image and its approximate version. The approximate image can be ‘de-noising’ image or slight ‘manipulated’ image. But in our work, the difference image is obtained by getting the difference between elements and its corresponding neighbors in JPEG 2-D array. Here, JPEG 2-D array is the absolute value of quantized block DCT coefficients in image, the reason why we get the difference from JPEG 2-D array instead of quantized block DCT coefficients directly has three aspects[13], more obvious zig-zag scanning characteristic, dynamic range reduction, less outline and edge information of image content. For convenience, we name the difference image as difference JPEG 2-D array [13].

There are two kinds of difference JPEG 2-D array we discuss here; one is defined as the difference between elements and their one-step neighbors in the JPEG 2-D array; the other is defined as the difference between elements and their two-step neighbors in the JPEG 2-D array. For simplicity, we named them as one-step difference JPEG 2-D array and two-step difference JPEG 2-D array respectively. By calculating difference, we should consider the direction of step. As can be seemed in Figure 1(a), we assume there have four basic directions from an element to one of its neighbors, horizontal, vertical, diagonal, and minor diagonal. If the position of initial element is marked as a star, the considered positions of one-step element neighbors can be marked as hollow circles along basic directions; and so forth, seven solid circles can be listed to mark the positions of element neighbors which are at least need two steps to reach. The vectors from star to solid circles can represent seven directions respectively, which will be considered to get two-step different JPEG 2-D arrays in our work. Figure 1(b) shows the four directions by one-step and Figure 1(c) shows the seven directions by two-step.

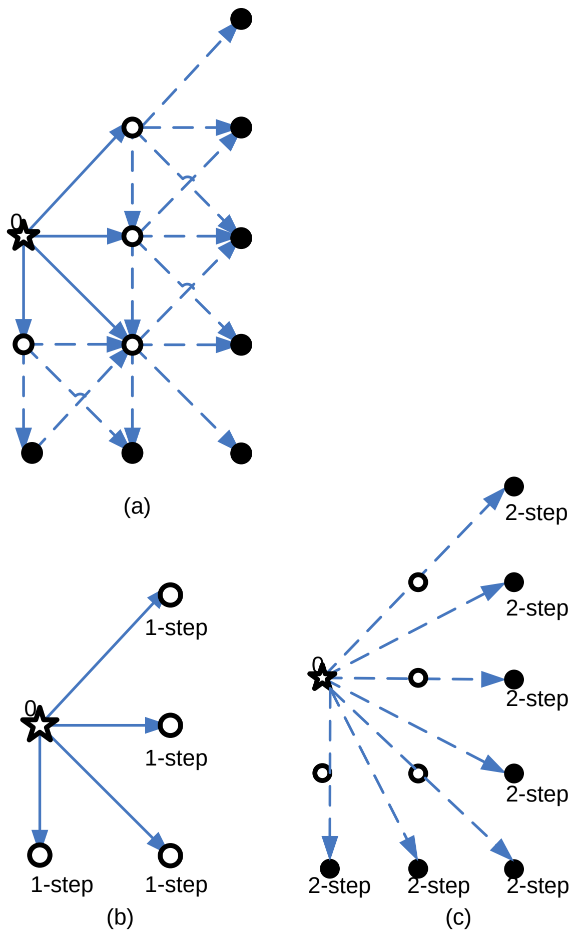

Figure 1. Different directions derivate from basic step; (a) Several possible directions; (b) Four directions by one-step; (c) Seven directions by two-step

According to the four directions we describe above, every 2-D JPEG array can derive four difference 2-D JPEG arrays. Figure 2 shows the generation method of these one-step difference 2-D JPEG arrays. In Figure 2, the small black boxes represent element positions of 2-D JPEG array. One-step difference 2-D JPEG array is generated by subtracting the array inside blue dotted box area from the array inside red real box area by elements. Figure 2 (a), (b), (c), (d) correspond to different generation ways along four directions, horizontal, vertical, main diagonal, and minor diagonal.

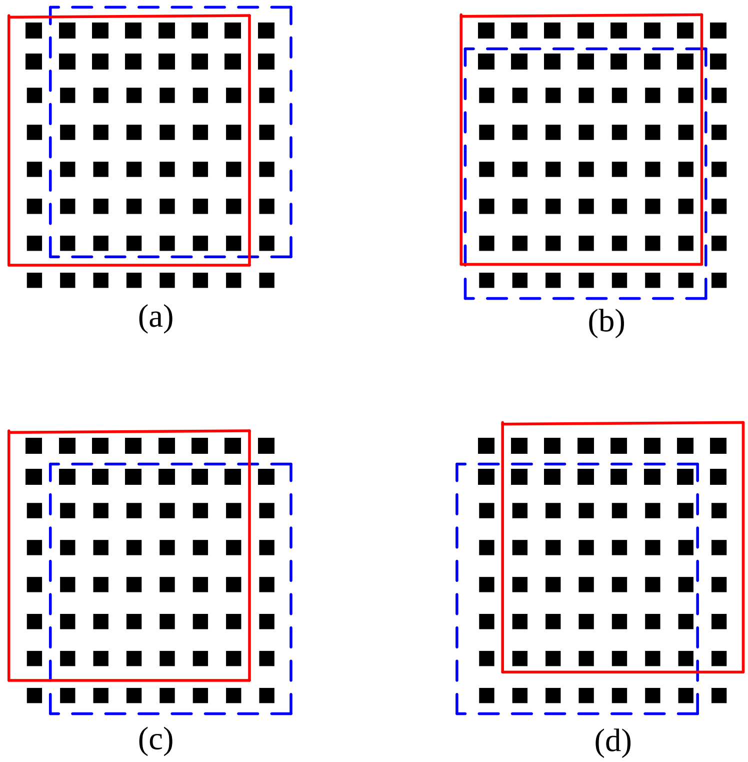

Figure 2. Generation method of one-step difference 2-D JPEG array; sub-fig. (a), (b), (c), (d) describe the generation along four different directions

As it's shown in Figure 2, the one-step difference JPEG 2-D arrays along four directions can be calculated as equal (1) to equal (4) respectively, which the JPEG 2-D array of image is denoted by $F(u,v)(u \in [1,S_u], v \in [1,S_v])$.

$$F_{D1}(u,v) = F(u,v) - F(u+1,v) \tag{1}$$

$$F_{D2}(u,v) = F(u,v) - F(u,v+1) \tag{2}$$

$$F_{D3}(u,v) = F(u,v) - F(u+1,v+1) \tag{3}$$

$$F_{D4}(u,v) = F(u+1,v) - F(u,v+1) \tag{4}$$

Where S_u is the size of the JPEG 2-D array in horizontal direction S_v is the size of it in vertical direction; for one-step difference JPEG 2-D array generation, $u \in [1, S_u - 1], v \in [1, S_v - 1]$ and the expression from F_{D1} to F_{D4} denote difference along four directions respectively.

According to the seven directions we describe above, every 2-D JPEG array can derive seven difference 2-D JPEG arrays. Figure 3 shows the generation method of these two-step difference 2-D JPEG arrays. In Figure 3, the small black boxes represent element positions of 2-D JPEG array. Two-step difference 2-D JPEG array is generated by subtracting the array inside blue dotted box area from the array inside red real box area by elements. Figure 3 (a), (b), (c), (d), (e), (f), (g) correspond to different generation ways along seven different directions.

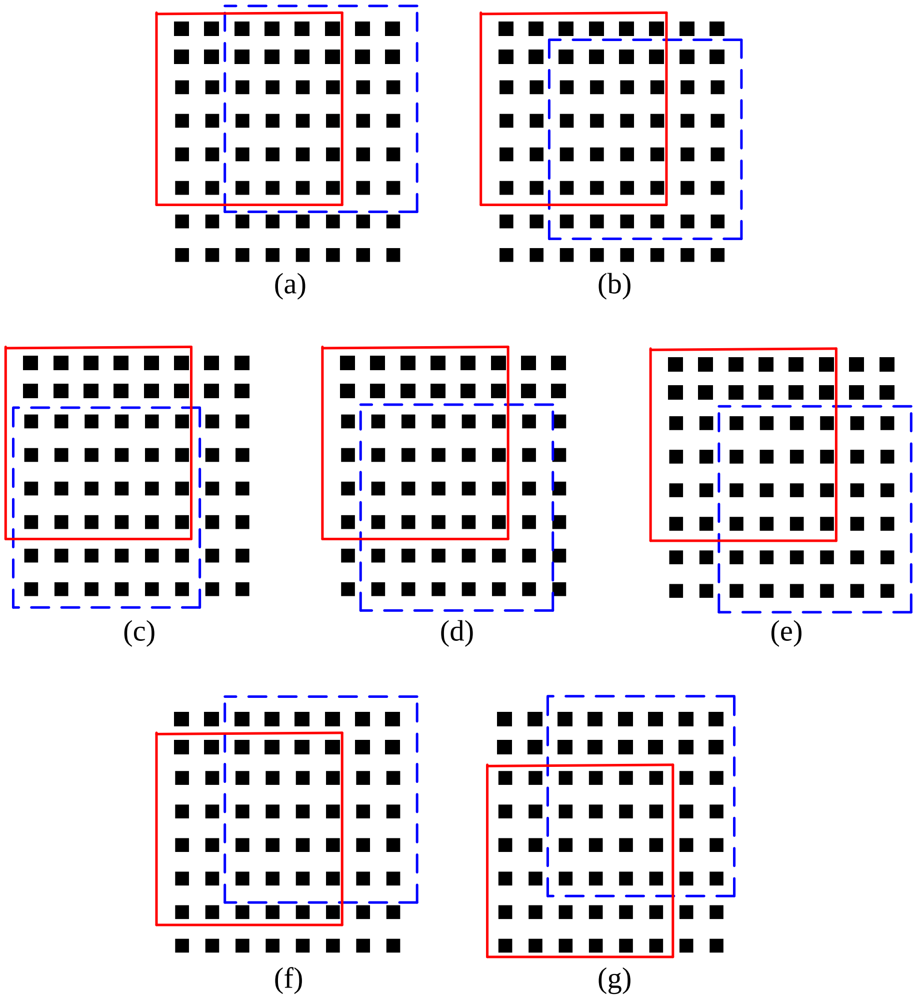

Figure 3. Generation method of two-step difference 2-D JPEG array; sub-fig. (a), (b), (c), (d) , (e), (f), (g) describe the generation along seven different directions

As it's shown in Figure 3, the two-step difference JPEG 2-D arrays along seven directions can be calculated as equal (5) to equal (11) respectively, which the JPEG 2-D array of image is denoted by $F(u,v)(u\in[1,S_u],v\in[1,S_v])$.

$$F_{d1}(u,v)=F(u,v)-F(u,v+2) \quad (5)$$

$$F_{d2}(u,v)=F(u,v)-F(u+1,v+2) \quad (6)$$

$$F_{d3}(u,v)=F(u,v)-F(u+2,v) \quad (7)$$

$$F_{d4}(u,v)=F(u,v)-F(u+2,v+1) \quad (8)$$

$$F_{d5}(u,v)=F(u,v)-F(u+2,v+2) \quad (9)$$

$$F_{d6}(u,v)=F(u+1,v)-F(u,v+2) \quad (10)$$

$$F_{d7}(u,v)=F(u+2,v)-F(u,v+2) \quad (11)$$

Where S_u is the size of the JPEG 2-D array in horizontal direction S_v is the size of it in vertical direction; for two-step difference JPEG 2-D array generation, $u\in[1,S_u-2],v\in[1,S_v-2]$ and the expression from F_{d1} to F_{d7} denote difference along seven directions respectively.

2.2. Multi-step Transition Probability Matrices

The difference JPEG 2-D array is better for detection accuracy (discussed in above section); therefore, we propose to model the difference JPEG 2-D array by Markov random process. According to the theory of random process, the transition probability matrix can be used to characterize the Markov process. Here, we extract one-step transition probability matrices from one-step difference JPEG 2-D arrays and two-step transition probability matrices from two-step difference JPEG 2-D arrays.

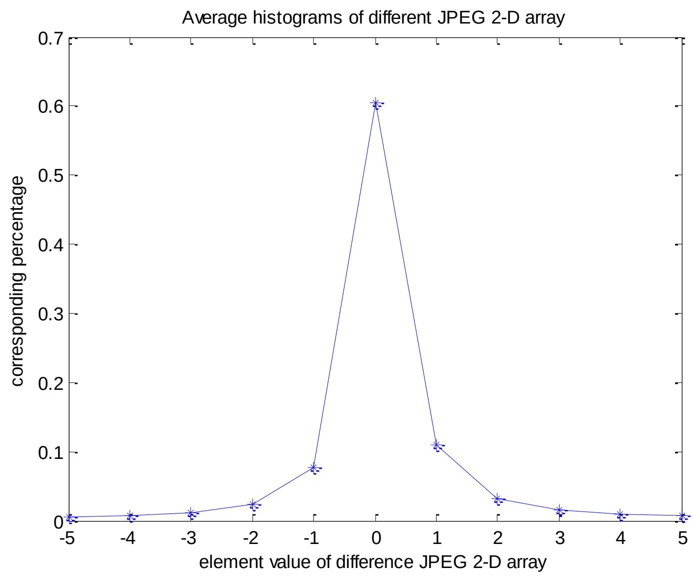

Figure 4. Average Histogram of one-step different JPEG 2-D Arrays along Horizontal Direction

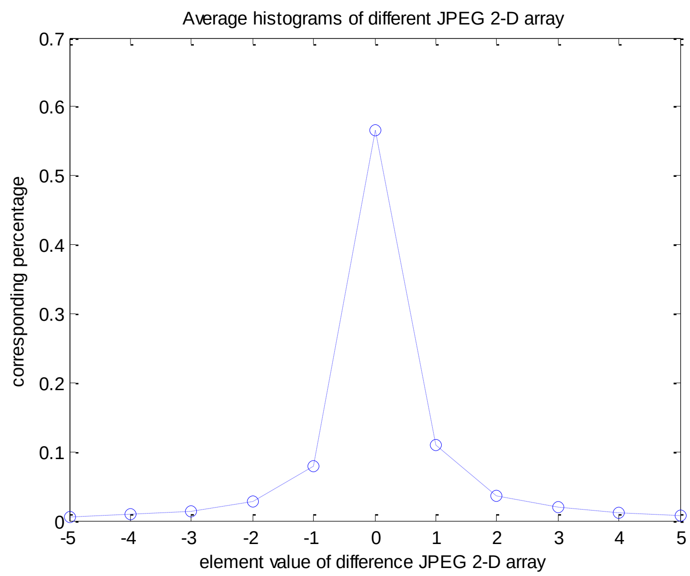

Figure 5. Average Histogram of two-step different JPEG 2-D arrays shown in Figure 3(a)

Though Markov process is a good choice to model the different JPEG 2-D array, the dynamic range of element value on difference JPEG 2-D array can be large theoretically. On this situation, extracting the transition probability matrix on difference JPEG 2-D array directly might end up large size. Figure 4 and Figure 5 show the average of histograms of one-step difference JPEG 2-D arrays along horizontal direction and two-step difference JPEG 2-D arrays shown in fig. 3(a) respectively. The images involved are randomly chosen from our experiment database, which will be described in experiment section later. As shown in Figure 4 and Figure 5, the average histogram seems like Laplacian distribution. It also means even we use a threshold-ing technique to limit value range from –T to T, most information can still be left with proper threshold choice. To reduce feature size and computational complexity, a threshold-ing technique is introduced. Firstly, we select a threshold value T; Secondly, we only consider these elements of difference JPEG 2-D arrays whose value falls into {-T, -T+1, …,-1, 0, 1, …, T-1, T}.If an element whose value is either large than T or smaller than –T, it will be represented by T or –T. This step results in a transition probability matrix of dimensionality (2T+1)× (2T+1).

The four one-step transition probability matrices are generated from the four one-step difference JPEG 2-D arrays. It can be calculated by following equals (12) to equals (15).

$$p\{F_{D1}(u+1,v)=n \mid F_{D1}(u,v)=m\}=\frac{\sum_{v=1}^{S_v-1}\sum_{u=1}^{S_u-1}\delta(F_{D1}(u,v)=m,F_{D1}(u+1,v)=n)}{\sum_{v=1}^{S_v-1}\sum_{u=1}^{S_u-1}\delta(F_{D1}(u,v)=m)} \tag{12}$$

$$p\{F_{D2}(u,v+1)=n \mid F_{D2}(u,v)=m\}=\frac{\sum_{v=1}^{S_v-1}\sum_{u=1}^{S_u-1}\delta(F_{D2}(u,v)=m,F_{D2}(u,v+1)=n)}{\sum_{v=1}^{S_v-1}\sum_{u=1}^{S_u-1}\delta(F_{D2}(u,v)=m)} \tag{13}$$

$$p\{F_{D3}(u+1,v+1)=n \mid F_{D3}(u,v)=m\}=\frac{\sum_{v=1}^{S_v-1}\sum_{u=1}^{S_u-1}\delta(F_{D3}(u,v)=m,F_{D3}(u+1,v+1)=n)}{\sum_{v=1}^{S_v-1}\sum_{u=1}^{S_u-1}\delta(F_{D3}(u,v)=m)} \tag{14}$$

$$p\{F_{D4}(u,v+1)=n \mid F_{D4}(u+1,v)=m\}=\frac{\sum_{v=1}^{S_v-1}\sum_{u=1}^{S_u-1}\delta(F_{D4}(u+1,v)=m,F_{D4}(u,v+1)=n)}{\sum_{v=1}^{S_v-1}\sum_{u=1}^{S_u-1}\delta(F_{D4}(u+1,v)=m)} \tag{15}$$

The seven two-step transition probability matrices are generated from the seven two-step difference JPEG 2-D arrays. It can be calculated by following equals (16) to equals (22).

$$p\{F_{d1}(u,v+2)=n \mid F_{d1}(u,v)=m\}=\frac{\sum_{v=1}^{S_v-2}\sum_{u=1}^{S_u-2}\delta(F_{d1}(u,v)=m,F_{d1}(u,v+2)=n)}{\sum_{v=1}^{S_v-2}\sum_{u=1}^{S_u-2}\delta(F_{d1}(u,v)=m)} \tag{16}$$

$$p\{F_{d2}(u+1,v+2)=n \mid F_{d2}(u,v)=m\}=\frac{\sum_{v=1}^{S_v-2}\sum_{u=1}^{S_u-2}\delta(F_{d2}(u,v)=m,F_{d2}(u+1,v+2)=n)}{\sum_{v=1}^{S_v-2}\sum_{u=1}^{S_u-2}\delta(F_{d2}(u,v)=m)} \tag{17}$$

$$p\{F_{d3}(u+2,v)=n \mid F_{d3}(u,v)=m\}=\frac{\sum_{v=1}^{S_v-2}\sum_{u=1}^{S_u-2}\delta(F_{d3}(u,v)=m,F_{d3}(u+2,v)=n)}{\sum_{v=1}^{S_v-2}\sum_{u=1}^{S_u-2}\delta(F_{d3}(u,v)=m)} \tag{18}$$

$$p\{F_{d4}(u+2,v+1)=n \mid F_{d4}(u,v)=m\}=\frac{\sum_{v=1}^{S_v-2}\sum_{u=1}^{S_u-2}\delta(F_{d4}(u,v)=m,F_{d4}(u+2,v+1)=n)}{\sum_{v=1}^{S_v-2}\sum_{u=1}^{S_u-2}\delta(F_{d4}(u,v)=m)} \tag{19}$$

$$p\{F_{d5}(u+2,v+2)=n \mid F_{d5}(u,v)=m\}=\frac{\sum_{v=1}^{S_v-2}\sum_{u=1}^{S_u-2}\delta(F_{d5}(u,v)=m,F_{d5}(u+2,v+2)=n)}{\sum_{v=1}^{S_v-2}\sum_{u=1}^{S_u-2}\delta(F_{d5}(u,v)=m)} \tag{20}$$

$$p\{F_{d6}(u,v+2)=n \mid F_{d6}(u+1,v)=m\}=\frac{\sum_{v=1}^{S_v-2}\sum_{u=1}^{S_u-2}\delta(F_{d6}(u+1,v)=m,F_{d6}(u,v+2)=n)}{\sum_{v=1}^{S_v-2}\sum_{u=1}^{S_u-2}\delta(F_{d6}(u+1,v)=m)} \tag{21}$$

$$p\{F_{d7}(u,v+2)=n \mid F_{d7}(u+2,v)=m\}=\frac{\sum_{v=1}^{S_v-2}\sum_{u=1}^{S_u-2}\delta(F_{d7}(u+2,v)=m,F_{d7}(u,v+2)=n)}{\sum_{v=1}^{S_v-2}\sum_{u=1}^{S_u-2}\delta(F_{d7}(u+2,v)=m)} \tag{22}$$

We denote T as threshold for one-step transition matrix and T' for two-step transition matrix. It means we have (2T+1)× (2T+1) elements for each of one-step transition probability matrix and (2T'+1)× (2T'+1) for each of two-step transition probability matrix. In total, we have 4×(2T+1)×(2T+1) plus 7×(2T'+1)×(2T'+1) elements. All of them are prepared to generate features for detection. It is clear that we should choose proper value of T and T' for good detection capability with manageable computational complexity.

3. Features

In this paper, camera-model identification is considered as a task of multi-class pattern recognition. That is, a given test image needs to be classified as which kind of camera-model took within our several known camera model types. Obviously, feature extracting method is the key issue of detection accuracy for classification.

We model difference JPEG 2-D array by transition probability matrix. To increase the accuracy of detection, a composition of multi-step transition matrices will be calculated from one-step and two-step difference JPEG 2-D arrays respectively. If we use these multi-step transition matrices as our features directly, we already have 4×(2T+1)×(2T+1) plus 7×(2T'+1)×(2T'+1) elements, mentioned in Section 2. To reduce the feature size and get balance between detection accuracy and computational complexity, we will average matrices by elements instead. Therefore, the size of final features will be (2T+1)×(2T+1) plus (2T'+1)×(2T'+1). We can see that threshold value is still key issue for feature size. In our experimental works, we set the threshold T, equal to 3; and the threshold T', equal to 1. The size of averaged one-step transition probability matrix is 7×7, and that of averaged two-step transition probability matrix is 3×3. Combining them together, we have 58 statistics as our features in total.

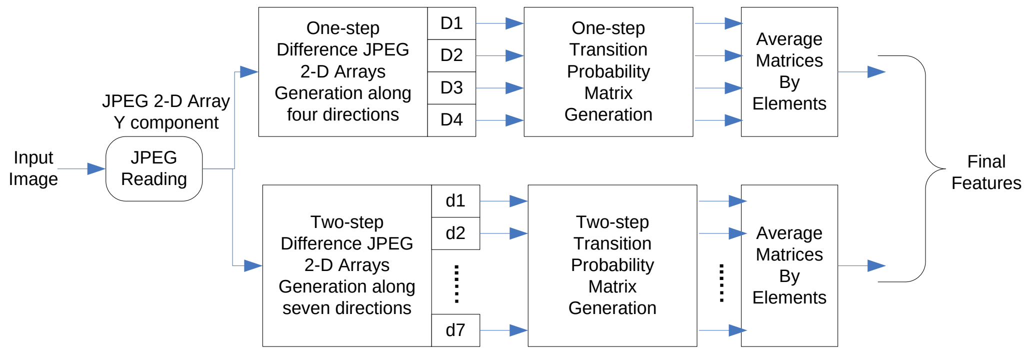

Figure 6. Feature Generation Flow Diagram

Figure 6 shows our feature generation flow diagram.

Firstly, we read JPEG 2-D array of Y component from input image.

Secondly, four one-step difference JPEG 2-D arrays along four directions are calculated by equation (12) to equation (15) respectively; and seven two-step difference JPEG 2-D arrays along seven directions are calculated by equation (16) to equation (22) respectively.

Thirdly, we set up the threshold T and T', calculate four one-step transition probability matrices from four one-step difference JPEG 2-D arrays respectively; and seven two-step transition probability matrices from seven two-step difference JPEG 2-D arrays respectively.

Fourthly, we average all one-step transition matrices by elements to get an averaged one-step transition matrix, and average all two-step transition matrices by elements to get an averaged two-step transition matrix.

Fifthly, combining the elements of averaged one-step transition matrix and two-step transition matrix, we can get a set of element values. This set is our final feature set we proposed here.

4. Experiment

In this section, we will introduce the experiment detail of our work. We firstly give the brief description on our image database. Then, the experiment method will be presented. And the last part of section will give the experiment result.

4.1. Image Database

In our experiment, we use the 'Dresden Image Database' as our image database. The Dresden image database is a public database designed for benchmarking algorithms in the area of digital image forensics. Most images are taken under same or similar acquisition procedure, such as at the same scenes, same taken positions, and same up to two motives with tripods in Dresden, and photographed with each camera of one set with systematically varying camera setting (flash, focal length and interchanging lens, if possible) [14]. The cameras used to build the dataset are different, which can be categorized in term of devices, models, bands. Therefore, the Dresden image database is collection of same or similar scene images taken by different camera, which can be categorized to do manufactory, model, or device detection or etc. Only the images taken from more than one device per model are suitable for model detection. Hence, images from seven models have been chosen. Table 1 shows some details of our experiment images.

Table 1. Camera Model

No.	model	Device num/model	Image num/model	Image resolution	Image format
1	CanonIxus70	3	567	3072×2304	JPEG
2	CasioEXZ150	5	925	3264×2448	JPEG
3	FujiFirmFinePixJ50	3	630	3264×2448	JPEG
4	NikonCoolPixS710	5	925	4352×3264	JPEG
5	NikonD70s	2	367	3008×2000	JPEG
6	NikonD200	2	752	3872×2592	JPEG
7	KodakM1063	5	2391	3664×2748	JPEG

4.2. Experiment Method

Most feature based classification method includes two parts, feature extraction and model classification. But our experiment has an extra processing step at first beginning, image blocking. To increase the number of classifying samples and unify sample size, for each image, we extract four 512x512 sub-blocks from center of difference JPEG 2-D array after JPEG reading. The sub-block position is showed in Figure 7.

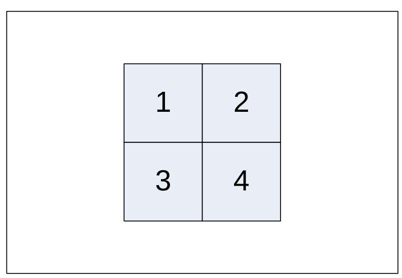

Fig. 7. Sub-block Array per each Image for Detection

According to the feature generation process shown in section 3, 58-D features are extracted from each sub-block. We use SVM classifier [15] to do model detection. 90% of the images are randomly chosen for training the classifier and the rest of them are used for testing. The random choosing is controlled to make sure the sub-blocks in training part and testing part are not from same image.

4.3. Experiment Result

In our experiment, we use images from seven models mentioned above to do model detection. The random selection of training and testing image sets and classification is performed 20 times, Table 2 shows the average detection rate, which only use averaged one-step transition matrix as 49-D features. The diagonal values are correct detection percentage rate for each model. According to Table 2, 49-D feature based method can identify seven models with high detection rate from 95.88% to 99.85%. The average detection accuracy of our proposed method is 98.73%.

Table 2. Model Detection Percentage Rate of 49-D Feature based Method (values below 0.05 are denoted as * for instead, blank denotes 0)

model	1	2	3	4	5	6	7
CanonIxus70	99.80		0.13		0.07		
CasioEXZ150	0.11	99.30	0.12	0.46	*		
FujiFirmFinePixJ50			99.48	0.40	0.06		0.06
NikonCoolPixS710	0.42	0.05	0.45	98.95	0.09	*	*
NikonD70s	*		1.80		95.88	0.92	1.39
NikonD200	0.71	0.07		0.08	1.28	97.86	
KodakM1063		*	*	*	*		99.85

Table 3 shows the average detection rate, which use the composition of averaged one-step and two-step transition matrices as 58-D features. The diagonal values are correct detection percentage rate for each model. According to Table 3, 58-D feature based method can identify seven models with high detection rate from 97.11% to 99.9%. The average detection accuracy of our proposed method is 99.27%.

Table 3. Model Detection Percentage Rate by 58-D Feature based Method (values below 0.05 are denoted as * for instead, blank denotes 0)

model	1	2	3	4	5	6	7
CanonIxus70	99.76			0.15		0.09	
CasioEXZ150	*	99.73		0.26			
FujiFirmFinePixJ50			99.84	0.08	0.08		
NikonCoolPixS710	0.16	*	0.12	99.64	0.05		
NikonD70s	*	0.07	1.29	0.14	97.11	1.26	0.10
NikonD200	0.07	0.15		0.08	0.85	98.85	
KodakM1063		*					99.99

In our comparison experiment, we use 324-D one-step Markov features extracted from Y component [10] to do model detection using exactly same image datasets for training and testing. Figure 8. shows detection accuracy of two algorithms. For each model, right side bars denote correct detection rate of our method, left side bars denote that of previous Markov method. From model 1 to model 7 denotes seven camera models respectively (shown in Table 1). We can see that the six models correct detection rates are all higher than that of one-step Markov method, except for one model. The average detection accuracy of our proposed method is 99.27%; for previous Markov method, the average detection accuracy is 98.78%.

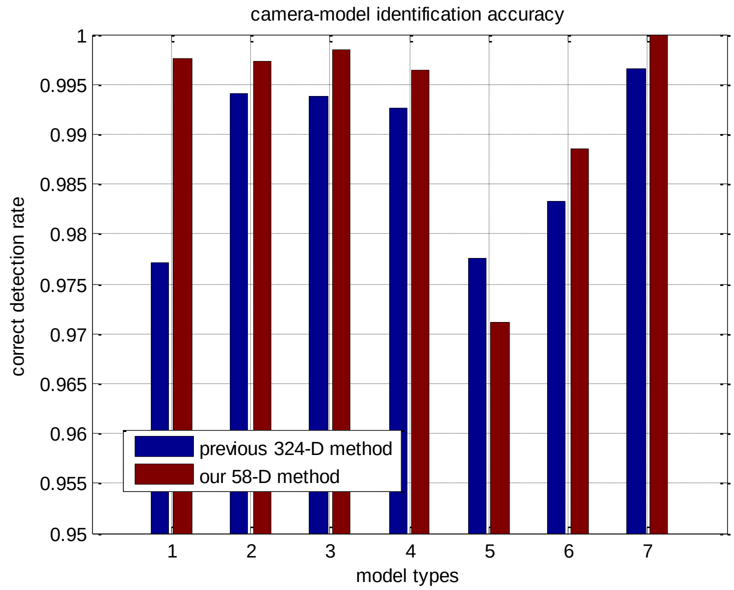

Figure 8. Model Detection Accuracy using our 58-D Method and Previous 324-D Markov Method [10]

In our experiment, samples are from the Dresden image database. According to the above experiment result, both our proposed method and Markov method works well on seven camera models classification. The detection accuracy of our proposed method is from 97.11% to 99.9%. The average correct detection rate is slight better than that of one-step Markov method, even our feature set is obvious smaller than that of one-step Markov method.

5. Conclusion

An algorithm for camera-model identification is presented in this paper. Multi-step transition probability matrices are used to model difference JPEG 2-D arrays along seven directions. The statistics of these matrices are combined to build detection features to catch the artifacts, which introduced by the whole imaging units. Feature size is also considered to be reduced by threshold-ing and averaging technology. We firstly calculate four one-step transition matrices and seven two-step transition matrices. Then, we set up two thresholds to generate one-step and two-step transition matrices. After that, we average these matrices by elements to reduce the size of features rapidly. Finally, combining the elements of two averaged transition matrices, 58-D features are obtained to do model detection. We use images from seven models of the Dresden image database as our samples. The experiment result shows that the detection accuracy of our proposed method works well on seven camera models. The average detection accuracy is 99.27%. Compared with previous Markov method, even with smaller feature size, our method can have good performance.

Acknowledgements

Thanks for Gloe, T. and Böhme, R, to build and maintain the Dresden image database is a huge load of work. They offer it to all of us. This work is supported in part by: The major national science and technology special projects (2010ZX03004-003-03), National Nature Science Foundation of China (No. 60832002), Nature Science

Foundation of Hubei Province (No. 2009CDB222, No. 2010CDB08602), and Doctoral Fund of Ministry of Education of China (Grant No.200910141110054).

References

[1] M. Kharrazi, H. T. Sencar and N. Memon, “Blind source camera identification”, In: *Proc. Int. Conf. Image Processing*, vol. 1, pp. 709–712, **(2004)** October 24-27; Singapore.

[2] K. S. Choi, E. Y. Lam and K. Y. Wong, “Automatic source identification using the intrinsic lens radial distortion”, *Opt. Express*, vol. 14, no. 24, pp. 11551–11565, **(2006)** November.

[3] J. Lucas, J. Fridrich and M. Goljan, “Digital camera identification from sensor pattern noise”, *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 2, pp. 205–214, **(2006)** June.

[4] M. Chen, J. Fridrich, M. Goljan and J. Lukáˇs, “Determining image origin and integrity using sensor noise”, *IEEE Trans. Inf. Security Forensics*, vol. 3, no. 1, pp. 74–90, **(2008)** March.

[5] Chang-Tsun Li, “Source Camera Identification Using Enhanced Sensor Pattern Noise”, *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 2, pp. 280–287, **(2010)** June.

[6] S. Bayram, H. T. Sencar and N. Memon, “Improvements on source camera-model identification based on CFA interpolation”, In: *Proc. Working Group 11.9 Int. Conf. Digital Forensics, FL* **(2006)** January 29 - February 1; Orlando, Florida, USA.

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

[8] 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, **(2009)**.

[9] M. Kirchner, “Efficient Estimation of CFA Pattern Configuration in Digital Camera Images”, Media Forensics and Security II, Proc. SPIE, Vol. 754110, **(2010)** January 18-20, San Jose, CA, USA.

[10] G. Xu, S. Gao, Y. Q. Shi, R. Hu and W. Su, “Camera-model identification using Markovian transition probability matrix”, 8th International Workshop on Digital Watermarking, , **(2009)** August 24-26, Springer Verlag, Guildford, UK.

[11] O. Celiktutan, B. Sankur and I. Avcibas, “Blind identification of source cell-phone model”, IEEE Trans. Inf. Forensics Security, vol. 3, no. 3, pp. 553-566, **(2008)** September.

[12] E. Dirik, H. T. Sencar and N. Memon, “Source camera identification based on sensor dust characteristics”, In: Proc. Signal Processing Applications Public Security Forensics, pp. 1–6, **(2007)** Apr. 11–13; Washington D.C. US.

[13] Y. Q. Shi, C. Chen, W. Chen, “A Markov Process Based Approach to Effective Attacking JPEG Steganography”, Information Hiding 2006, pp. 249-264, **(2006)** July 10-12, Alexandria, VA, USA.

[14] T. Gloe and R. Böhme, “The 'Dresden Image Database' for benchmarking digital image forensics”, In Proceedings of SAC, pp. 1584-1590, **(2010)** March 22-26, Sierre, Switzerland.

[15] C.-C. Chang and C.-J. Lin, “LIBSVM: a library for support vector machines”, **(2001)**, http://www.csie.ntu.edu.tw/˜cjlin/libsvm (2001).