Source Camera Identification Using Auto-White Balance Approximation

Zhonghai Deng
Univ.of Alabama
Tuscaloosa,AL, USA
zdeng@ua.edu

Arjan Gijsenij
Univ. of Amsterdam
Netherlands
a.gijsenij@uva.nl

Jingyuan Zhang
Univ. of Alabama
Tuscaloosa,AL, USA
zhang@cs.ua.edu

Abstract

Source camera identification finds many applications in real world. Although many identification methods have been proposed, they work with only a small set of cameras, and are weak at identifying cameras of the same model. Based on the observation that a digital image would not change if the same Auto-White Balance (AWB) algorithm is applied for the second time, this paper proposes to identify the source camera by approximating the AWB algorithm used inside the camera. To the best of our knowledge, this is the first time that a source camera identification method based on AWB has been reported.

Experiments show near perfect accuracy in identifying cameras of different brands and models. Besides, proposed method performances quite well in distinguishing among camera devices of the same model, as AWB is done at the end of imaging pipeline, any small differences induced earlier will lead to different types of AWB output. Furthermore, the performance remains stable as the number of cameras grows large.

1. Introduction

With the popularity of digital cameras and the ease of image editing, image forensics becomes indispensable. Generally, the goal of image forensics is either *authentication* or *integrity validation*. Authentication is to identify the source imaging device of a given image. Integrity validation involves determining whether the digital image has been modified, and if so, what kinds of manipulations are performed. In this paper, we focus on authentication, i.e. given an input image, identifying its source camera.

Source camera identification finds applications in many cases. For example, when digital images are used as evidence in court, it is necessary to verify the original source of such images. Further, in the case of copyright dispute over an image, identifying the source camera could help find the rightful owner of the image. An apparent simple solution is to use the EXIF (*Exchangeable Image File*) header of an image [27]. However, such information is very easy to manipulate, and therefore not usable in practice.

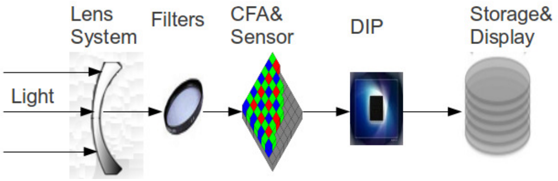

Figure 1. *Imaging pipeline in digital camera (Reproduced from [22] with permission)*

A good source camera identification solution shall rely on the image acquisition process rather than easy-to-manipulate meta-data. Although the detailed image acquisition process is kept secret by the camera manufacturers, the imaging pipeline is similar among most digital cameras (Figure 1). Light coming from the outside world passes through the camera lens, and a series of filters, including color filter array (CFA). Then it reaches the sensor (CCD/CMOS), where it is converted into digital signal, and subsequently processed by a digital image processor (DIP), where post-processing operations are performed, including gamma correction, demosaicking, image correction, white balance and JPEG compression.

One way to perform camera identification is to make use of lens distortion/aberration [2, 8, 30]. However, since lenses are often interchangeable, this approach is not reliable for practical camera identification.

Another type of approach makes use of the inherent manufacturer's imperfection, such as defective pixels [17], pattern noise [13, 25, 26], camera response functions [28] or sensor dust characteristics [10, 9]. However, such approaches are often not accurate enough for real-world applications.

The third type of approaches focuses on the Digital Image Processor (DIP). Due to the use of CFA, every camera performs a demosaicking algorithm to obtain a color image. To estimate the interpolation coefficients, Alin *et al.* [29] assume demosaicking is a linear model, while Long

978-1-4577-1102-2/11/$26.00 ©2011 IEEE

and Huang [24] assume it to be quadratic correlation model. However, the color demosaicking process is highly nonlinear [20], but all the above methods assume it is in near linear form, while having no consideration of side effects of other post-operations in DIP.

Our proposed approach falls in the third category. Rather than focusing on demosaicking algorithms, we propose to use automatic white balance (AWB) for camera identification. White balance is applied all digital cameras to correct images from the color of the light source, *e.g.* to ensure consistent color reproductions for images or to remove unnatural looking color casts. To the best of our knowledge, AWB has not been used in image forensics. Our method is inspired by the intuition that applying AWB for the second time would not change much to the image, since any color cast would have been removed by the first AWB operation. Based on this intuition, we introduce a novel source camera identification method using AWB approximation.

Section 2 briefly explains the most common approaches to computational white balance. Section 3 introduces our proposed method, and section 4 describes and discusses our experiment results. Robustness analysis of the proposed method is given in Section 5. Finally, Section 6 presents the conclusion.

2. Auto-White Balance Approaches (AWB)

Various qualities of the human visual system are taken for granted without realizing it. For instance, the human visual system is known to continuously adapt to changing environments. This results in consistent perception of surface colors (i.e. *color constancy*), even when these surfaces are observed under completely different light sources (e.g. fluorescent or incandescent light bulbs or natural daylight). Digital cameras attempt to reproduce these results by applying (*Automatic*)*White Balance* to every recorded image. Would this step be omitted, all images would have a color cast, depending on the environment where the image was taken. For instance, images recorded under fluorescent lights would have a greenish cast and images under incandescent lights would have a yellowish cast [1].

During the process of color image formation, light reflected off the surfaces in a scene that reaches the sensor of the digital camera is transformed into a color space that can be interpreted by display devices such as monitors, usually RGB. This light is the product of the spectral surface reflectance and the spectral power distribution of the illuminant. After the light reaches the sensor, transformation is applied using three *camera sensitivity functions* that each respond to specific parts of the light spectrum. White balance is defined as the problem of disentangling the effects of the light source from the resulting RGB-image without changing the actual contents of the image. Generally, this problem is attacked by first estimating the color of the light source $\mathbf{e} = (e_R, e_G, e_B)^T$ (which is assumed to be spectrally constant across the scene), followed by transformation of the RGB-image to impose a canonical illuminant $\mathbf{c}$, usually a white light source (i.e. $\mathbf{c} = (1, 1, 1)^T$). In this section, several alternatives to estimate the illuminant are discussed, as well as some chromatic adaptation transforms.

2.1. Illuminant Estimation

Since estimation of the illuminant is an under constrained problem, i.e. the amount of information required to solve the problem is larger than the amount of information available, all existing algorithms are based on one or more assumptions. Most commercial cameras are doing white balance based on the best-known *Gray-World assumption*. Under this assumption, the average color of an image that is recorded under a white light source is achromatic. One algorithm based on this assumption simply sets the color of the light source to the average color of the image[6], as any deviation from gray, by assumption, is caused by the illuminant. Alternatively, rather than computing the average of *all* pixels, using only the center pixels of a segmented image has been shown to improve the accuracy [4, 5].

Another simple yet well-known algorithm is based on the *White-Patch* assumption [21]. This assumption states that the maximum response in any image is caused by a white patch (i.e. a perfect reflectance). A simple method, known as *Max-RGB*,that utilizes this assumption computes the maximum responses in either of the three channels R, G and B and sets the color of the light source to this value.

In real-world images, both assumptions are likely to fail. Therefore, Finlayson and Trezzi[15] propose to compute a weighted average of the pixel values, assigning higher weights to pixels with higher intensities. The weight-function that is proposed is based on the Minkowski-norm p, which implies that the Gray-World and the White-Patch can be generated by using $p = 1$ and $p = \infty$, respectively. Another extension of these low-level statistics-based methods is proposed by van de Weijer et al. [31], and is based on the *Gray-Edge assumption*. This method works with derivatives of images (i.e.edges) rather than with the original pixel-values.

2.2. Chromatic Adaptation

After the color of the light source is estimated, the image can be transformed. This transformation will change the appearance of all colors, so that the image appears to be recorded under a white light source (e.g. $D65$). This can be achieved by *chromatic adaptation*, e.g. [12]. Most adaptation transforms are modeled using a linear scaling of the cone responses, and the simplest form independently scales the three color channel[14, 32]:

$$\begin{pmatrix} R_c \\ G_c \\ B_c \end{pmatrix} = \begin{pmatrix} d_R & 0 & 0 \\ 0 & d_R & 0 \\ 0 & 0 & d_B \end{pmatrix} \begin{pmatrix} R_e \\ G_e \\ B_e \end{pmatrix}, \qquad (1)$$

where $d_i = \frac{e_i}{\sqrt{3 \cdot (e_R^2 + e_G^2 + e_B^2)}}$, $i \in \{R, G, B\}$.

More accurate representation includes Bradford transform, CMCCAT2000, VonKries, and XYZ model, etc.[23].

3. Methodology

Any digital camera performs some kind of white balancing inside the camera. Even if users turn off AWB, the camera would still perform some fixed color correction operation. Our proposed method is based on the observation that most white balance algorithms will have little or no effect on the image if they are applied to the image the second time.

3.1. Theoretical Basis

Above observation illustrate the 'idempotence' property of white-balance method, the theoretical basis of this paper. An operation having 'idempotence' property, will produce the same output if executed once or multiple times, i.e. given an image im, we have

$$WB(WB(im)) = WB(im) \qquad (2)$$

This promises that, if white-balance is the last operation inside camera, and we happen to choose the same AWB method as the image has undergone, the output image would not change. For example, adjusting the image so that the average color is gray, i.e. the Gray-World algorithm, will expect the same average color. Thus applying this method the second time will have no effect on the image, since the average of the image is already set to gray. Furthermore, we observe that the methods that are based on the same assumption trend to produce similar results. For example, methods based on the gray-world assumption may receive the same illuminant estimation, thus more likely gives similar results. Finally, we we observe that methods based on different assumptions tends to give much larger color changes.

There are two questions that we must address to solid our theory. First, is the AWB performed at the end of imaging pipeline? The answer is NO. Since at least, the JPEG compression happens after it. But the lucky thing is that, from analysis of imaging pipeline, it is reasonable to believe that some major operations in DIP happen ahead of white-balance, including infrared rejection, gamma correction, demosaicking, lens aberration, antialiasing, etc. Also, our experiments shows that for many images, we can find the AWB algorithm that has little effect when performed again. (Mean Square Error < 0.5). We attribute this different to the image quality degradation due to JPEG compression. What is more, sometimes, for a high quality images (compression quality $\geq 98\%$), we can find the exact AWB algorithm that will be side effect free. From above, we can reasonably assume that white-balance is performed near the end of imaging pipeline, thus the proposed method does not suffer the side effects from other processes applied inside DIP.

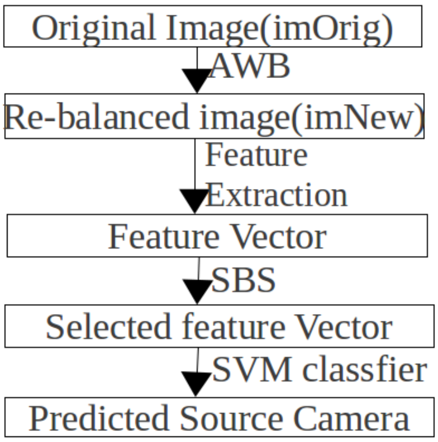

Figure 2. *Working flow of our proposed method*

The second question is, how could we assure that we have find the right AWB method? Since most digital cameras include multiple AWB methods designed for different lighting conditions. And what is worse, the actual method that is used inside each camera is unknown. Therefore, in this paper, instead of finding one algorithm for a particular image, we simply use various methods to approximate the methods that may be used inside camera. Since even the wrong algorithm would still have its own difference pattern.

The whole approximation process (Figure 2) is as follows. Given an original image *imOrig*, we first apply white balance to obtain a new image *imNew*. We then extract some Image Quality Metrics (IQM) from *imNew* as features, while outliers are eliminated using sequential backward feature selection (SBS). Finally, SVM classification is performed to identify the source camera of the given image.

3.2. Feature Extraction

In our experiments, all features come from image quality metrics (IQM) of the re-balanced images. The underlying philosophy is, for re-balanced image, the less change from re-balancing, the better quality it is , compared to the original image. This quality metrics thus could be used to identify the

Name	Number
Mean Absolute Error (MAE)	3
Mean Square Error (MSE)	3
Normalized MSE	3
Peak-Signal-to-Noise Ratio	3
Maximum Difference	3
Czekznowski Correlation	1
Angle Mean	1
Structural Content	3
Correlation Quality	3
HVS real lab distance	1
HVS similarity weight	1
Block Weighted Spectral Distance (max,mean,median)	3*3

Table 1. Image Quality Metrics used

3.2.1 AWB Methods

For each image, we apply various kinds of AWB methods based on different assumptions. Since the majority of cameras use gray-world assumption, we apply more variations of gray-world methods.

- Gray-World: with 6 color adaptation methods;
- White-Patch: with percentage of white set to be $1/255, 0.01$, and max-RGB with smoothing;
- Shades-of-gray;
- Gray-Edge (with differentiation order 1 and 2).

The six adaptation methods used in combination with the Gray-World method are the diagonal (eq.(1)), von Kries, Bradford, Sharp, CMCCAT2000 and XYZ model [23].

3.2.2 Image Quality Metrics (IQM)

Image quality metrics are used to denote the set of metrics to evaluate the image quality. To extract these features, we apply various white balance methods on the input image *imOrig* to obtain *imNew*. Using *imOrig* as baseline, various features [3, 11] are extracted from *imNew*, see Table 1 for an overview of features used. Note that the number of features indicate whether a feature is extracted from a full color image or from RGB channels separately.

To summarize, 12 re-balanced images are obtained by applying 12 white balance methods on each original image. Then, for each *imNew* we extract 34 features (Table 1), resulting in a total of 408 features per image.

3.3. Feature Selection

Feature selection is used to reduce the noise in the features and to eliminate outliers. For simplicity and computational reasons, we use the sequential backward feature selection (SBS) algorithm. This method attempts to optimize some criterion by removing features from an initial candidate feature set. In our implementation, the ensemble of features of 17 camera models is used as initial candidate set, and the SVM classifier accuracy is used as optimization criterion. Since SBS eliminates four features from the initial set of features, thus in all experiments we use a feature vector of dimension 404.

3.4. SVM Classifier

We use support vector machine (SVM) of the RBF kernel to test the effectiveness of our proposed features, with $C = 2^7, \gamma = 2^{-7.5}$. To be specific, we use LibSVM package [7], since we mainly focus on multiple class identification.

4. Experimental Results and Discussions

In our experiments, we first perform camera identification over cameras of different brand. Then we test all 17 models available in the database, with some models similar with each other. At the end, we test with images coming from different devices of the same model.

4.1. Database Used

In our experiments, images come from ‘Dresden Image Database’ [18], and we use all the camera devices available in the database, up to 29 cameras devices, with 17 models, and 8 brands (Table 2). We use only the first 169 images for each camera device, as this is the minimum number of images available per device.

The main reason that we use this database is that, almost every camera takes a picture of the same scene, and under the same lighting condition. Although it would be harder for classification, that is exactly what source camera identification is about (imagine two persons arguing the copyright of an image).

Note all images we use are in JPEG format, and each time, we randomly choose 60% images as training samples and the rest 40% for testing, unless otherwise explicitly mentioned. Further, all classification accuracies listed in this paper are the average of 250 running results.

Brand	Model	Alias	Brand	Model	Alias
Agfa	DC-504	A1	Nikon	D70	N1
	DC-733s	A2		D70S	N2
	DC-830i	A3		D200	N3
	505-x	A4		S710	$N0_i$
	530s	A5	Rollei	325XS	R1
Canon	Ixus55	C1	Olympus	μ1050SW	O1
	Ixus70	C2	Casio	EX-Z150	$S0_i$
	A640	C3	FujiFilm	J50	F1
Kodak	M1063	$K0_i$	-	-	-

Table 2. Camera Models and Resolutions (First letter as brand ID, second number as model ID and subindex as device ID)

\	C2	S0	K0	N3	N1	O1	A1	R1	F1	C3
C2	**99.0**	0.10	0.38	-	-	0.01	0.22	0.24	0.09	-
S0	0.03	**99.3**	-	0.01	0.16	0.12	0.01	0.09	0.28	-
K0	0.04	0.09	**99.1**	0.04	0.62	0.06	-	0.03	0.06	-
N3	-	0.01	-	**99.8**	0.07	0.07	0.06	0.01	-	-
N1	-	0.35	0.04	0.31	**99.2**	0.07	-	0.03	0.01	-
O1	0.53	0.06	-	0.15	0.06	**98.9**	0.25	0.04	0.01	-
A1	0.03	-	-	-	-	0.07	**99.3**	0.46	0.16	-
R1	0.12	-	-	0.03	0.01	0.12	-	**99.4**	0.32	0.01
F1	0.03	-	0.06	-	0.40	0.09	-	0.26	**99.1**	0.04
C3	-	-	-	-	-	-	-	0.13	0.15	**99.7**

Table 3. Confusion Matrix for 10 Camera Identification(for $S0_0, K0_0$, we omit subindex for simplicity)

For camera model having only one device, we omit the device ID. For ‘Kodak M1063’, ‘Nikon CoolPixS710’ and ‘Casio EX-Z150’, we have five camera devices, we use subindex $i = 0, 1, ..., 4$ to identify device 0 to 4, and omitting their device ID indicates device 0.

4.2. Cameras of Different Brands

For feature based camera model identification, Gloe *et al.* [19] did a comprehensive evaluation on existing best performing methods using the same forensic database.

Our initial design uses 8 cameras from different brands, but to be more comparable and convincing, we include two additional cameras $A4$ and $C3$, thus replicate the setting of naive test from [19]. (Note: since image data from 3 Samsung cameras are not available, we replace them by those from camera $A1, F1, C3$.)

Experiments show that the average classification accuracy is 99.26%, the lowest running as 98.38%, and the highest as 100%. Detailed results are shown in Table 3, where the left most column is the true source camera while the top row is the predicted result, and each entry is the prediction percentage. In this experiment, Gloe et al. [19] reports an overall correct model identification performance of 97.79%, where our performance is 99.26%. Further, Table 3 shows no specific camera model performs considerably worse than others.

4.3. Cameras of Different Models

In the second experiment, we evaluate the performance over all 17 models in ’Dresden Image Database’. The average predicting accuracy is 98.61%, with lowest running as 96.97%, and the highest as 99.57% (Table 4).

From these experiments, we can observe that the proposed method can distinguish among camera models as well as camera brands. Moreover, the proposed method scales well for an increasing number of different cameras.

These results can be explained as follows. First, as the feature vectors are derived from the image quality metrics, they are inherently consistent with each other, which is also reflected in the result of the SBS feature selection method, which eliminates only four features. Second, all white balancing methods are based on the gray-world assumption, which is considered to be the most often used method inside digital cameras. Finally, it is likely that white balancing is the at the end of the digital image processing (DIP) pipeline, thus the proposed method does not suffer the side effects from other processes applied inside the DIP.

Camera	**Accuracy**	**Camera**	**Accuracy**
A1	99.1529	C1	98.6588
A2	97.5059	C2	97.0765
A3	98.5118	C3	99.5529
A4	98.6706	R1	99.2882
A5	97.9059	N1	98.0824
S0	98.9353	N2	97.8647
F1	99.0941	N3	99.8353
K0	98.8000	N0	98.8118
O1	98.6529	-	-

Table 4. Prediction percentage for all available 17 cameras of different models

	$S0_0$	$S0_1$	$S0_2$	$S0_3$	$S0_4$
$S0_0$	**99.12**	0.58	-	0.31	-
$S0_1$	0.10	**98.72**	0.30	0.62	0.25
$S0_2$	0.41	0.41	**98.12**	0.44	0.62
$S0_3$	0.09	0.27	0.35	**99.25**	0.04
$S0_4$	0.55	0.39	1.12	0.30	**97.65**

Table 5. Confusion Matrix for 5 devices of ‘Casio EX-Z150’ Identification

4.4. Cameras of the Same Model

Currently, the most challenging problem of camera identification is to distinguish among camera devices of the same model. The next experiment will follow the experimental setup of [18] to evaluate the intra-camera performance over 5 devices of model ‘Casio EX-Z150’. By randomly choosing 60% images for training, we have an average accuracy of 98.57%, with the lowest running accuracy being 96.47%, and the highest being 100%. Similarly identification over 5 camera devices from ‘Kodak M1063’ gives an average accuracy of 98.47% and ‘Nikon CoolPixS710’ of 98.78%.

These results (Table 5) clearly depict that even for devices of the same model, our proposed method can correctly identify the source camera of a given image.

To explain this, we did more investigations. First, for ‘Nikon CoolPixS710‘ , if we open images ‘_1_13228.JPG’ and ‘_2_13645.JPG’, (‘_1_‘ and ‘_2_‘ are device ID), we could find that the image from device 1 is brighter than that from device 2, even their content are the same. This observation holds for all images having the same scene from

File Name	Exposure Time	AFPoint	AFPoints InFocus	Relative AF-Position
0_12830	1/400 sec	3	8	Mid-left
1_13228	1/400 sec	4	16	Mid-right
2_13645	1/400 sec	0	1	center
3_14082	1/320 sec	8	256	Lower-right
4_14500	1/320 sec	3	8	Mid-right

Table 6. Device difference among ‘Nikon CoolPixS710’ (0,1,2,3,4 are device ID, and AF is Auto-Focus)

these two devices, even though they are taken at the same time and under the same lighting condition.

Next, by checking their ‘MakerNotes’, we find that the selected auto-focus position varies between devices, even they are of the same model (Table 6). We believe that the trivial differences like this, cause the differences in scene illumination estimation process, and finally affect the output images. To better understand how these differences affect the feandatures extracted, we reduce the dimension to 2 by principle component analysis (PCA) (Figure 3).

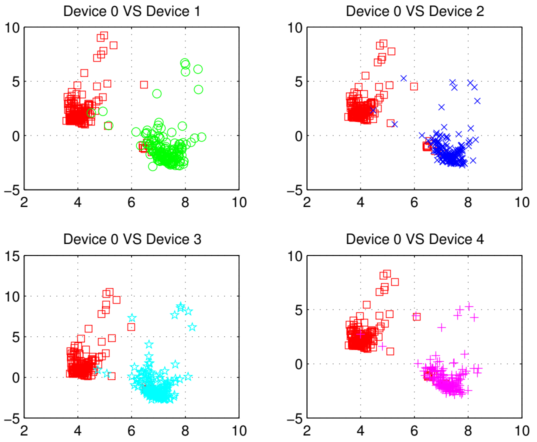

Figure 3. *Comparison of device 0 against other 4 devices from model ‘Casio EX-Z150’ (Red squares represent images from device 0)*

To be more general, we also did the test over all 29 camera devices listed in Table 2, with 60% images for training, experiment gives an average accuracy of 98.10%, with the lowest running as 96.55%, and the highest as 99.04%. Due to the use of a huge number of camera devices including 15 devices out of 3 camera models, the performance degrades a little, but it is still acceptable.

5. Robustness Analysis

To test the robustness of our method, we expose the images to several most common manipulations.

5.1. Robustness over Double JPEG Compression

Practically almost all digital cameras on market use JPEG compression to store the images. When someone modifies an image afterwards, the result is saved with JPEG compression again, resulting in double JPEG compression.

To test whether our proposed method would be resistant to double JPEG compression, we compressed the original JPEG image again with 75% quality metric. To analyze the robustness of the proposed approach, we used the same experimental setup as the first experiment in section 4.2. Training is performed on 60% of the original images; while testing is performed using the rest images that are double JPEG compressed.

Experiment shows the average prediction accuracy is 89.90%. Compared with the 99.14% for original accuracy, the performance decreased a lot. When only 5 cameras are used, the prediction accuracy is still 98.40%, while its original accuracy is 99.24%.

This experiment proved that double JPEG compression disturbs the consistency of image quality characters. When the number of camera devices grows larger, especially with cameras of close models, prediction accuracy decreases a lot.

5.2. Robustness over Noise

Every digital camera would introduce some noise to its captured images, such as noises from sensor dust, from dead pixels on sensor etc. , and these noises tend to change as time passes.

To make sure that images from a camera taken 3 years ago could be classified into the same class with those taken today, we redo the first experiment. First, we add Gaussian white noise of mean zero and variance 0.01 to the original images. Similarly, 60% original images are used for training, and when testing, we use the rest images with noise added.

The average prediction accuracy is 97.79%, with the lowest accuracy as 94.12%, and the highest accuracy as 99.41% . From the experiment, we can assert that our method is robust to white Gaussian noise, and thus, images taken 3 years ago would still be classified into the same category with those taken today.

5.3. Resistance over Resizing

Resizing an image is a common operation in image manipulation. To test the resistance over resizing, we also redo the first experiment.

At the beginning, we resize original images to its $1/2$ images with ‘bi-cubic’ interpolation. When testing, we use features from their resized images. *(Note: our images ‘original‘ image are already* $1/4$ *resized.)* Experiment gives only 75.61% prediction correctness, indicating much more disturbance in features.

But this low accuracy could be overcomed by using the resized image for training. In that way, the average accuracy turns to be 98.59%, which means that re-sampling operation, not simply disturbing the feature, but transfers features into the same direction, thus features still consistent with each other. For example, if we perform the resizing on any image in database, and compare their max and min value on each channel, we will found that resized image tends to have significantly larger min and max values.

Therefore, given target images been re-sampled, using their same sized images would give acceptable authentification result.

6. Conclusions

In this paper, we propose a novel method to identify the source camera by using the AWB residue pattern. Experimental results on a large-scale data set show the proposed method is very effective. Moreover, the prediction accuracy almost does not degrade as the number of different cameras increases, demonstrating the scalability of the proposed method. Finally, we show that even for different devices of the same model and brand, the proposed method is still able to distinguish among them.

Although we only do the source camera identification, the same idea could be applied to various applications, including but not limited to copy-move detection and steganalysis, as well as reverse engineering.

Acknowledgments

We should thank Dr. Nasir Memon for sharing their codes for feature extraction, Thomas Gloe for his explanation on 'Dresden Image Database'. Thanks also goes to the anonymous reviewer for his insightful comments and valuable suggestion.

Appendix

In this section, we give brief descriptions of some image quality features in Table 1. Given an original RGB image $C(i, j, k)$ of size M-by-N, where $i = 1, 2, ..., M$, $j = 1, 2, ..., N$ and $k = 1, 2, 3$ indicating its color plane. And $\hat{C}(i, j, k)$ stands for its re-balanced image.

A. Minkowsky Metrics

Minkowsky metrics (M) could be used to assess the similarity of two images, by averaging the pixel-by-pixel difference.

$$M_\gamma(k) = \left\{ \frac{1}{M * N} \sum_{i=1,j=1}^{M,N} |C(i,j,k) - \hat{C}(i,j,k)|^\gamma \right\}^{1/\gamma} \tag{1}$$

$\gamma = 1$ corresponds to Mean Absolute Error (MAE), $\gamma = 2$ corresponds to rooted Mean Square Error (RMSE), meaning that $MSE(k) = RMSE^2(k)$. When $\gamma = +\infty$, we actually get the Maximum Difference.

To get the Normalized Mean Square Error (NMSE), we normalize the square error by the quadratic of the baseline image.

$$NMSE(k) = \frac{M * N * MSE(k)}{\sum_{i=1}^{M} \sum_{j=1}^{N} C^2(i,j,k)} \tag{2}$$

Similarly, Peak Signal to Noise Ratio(PSNR) is actually the reciprocal of RMSE, and scaled by its logarithm.

$$PSNR(k) = 20 * log_{10} \frac{255}{RMSE(k)} \tag{3}$$

B. Correlation Metrics

Another types of metrics, focused on non-negative components, is the Czekanowski Correlation (CC) [3].

$$CC = 1 - \frac{2 * \sum_{i=1,j=1}^{M,N} \sum_{k=1}^{3} min\left(C(i,j,k), \hat{C}(i,j,k)\right)}{\sum_{i=1,j=1}^{M,N} \sum_{k=1}^{3} \left(C(i,j,k) + \hat{C}(i,j,k)\right)} \tag{4}$$

Another correlation based metric is the Angle Mean (AM), assuming at position (i,j), the RGB values of image C is denoted by bold character ***C***(i,j) as a 3-D vector.

$$AM = 1 - \frac{1}{M * N} \sum_{i=1,j=1}^{M,N} \frac{2}{\pi} \frac{\left\langle \mathbf{C}(i,j), \hat{\mathbf{C}}(i,j) \right\rangle}{||\mathbf{C}(i,j)|| * ||\hat{\mathbf{C}}(i,j)||} \tag{5}$$

Similarly, Correlation Qualities (CQ) and Structural Contents (SC) are

$$CQ(k) = \frac{\sum_{i=1,j=1}^{M,N} C(i,j,k) * \hat{C}(i,j,k)}{\sum_{i=1,j=1}^{M,N} C^2(i,j,k)} \tag{6}$$

$$SC(k) = \frac{\sum_{i=1,j=1}^{M,N} C^2(i,j,k)}{\sum_{i=1,j=1}^{M,N} \hat{C}^2(i,j,k)} \tag{7}$$

C. Quality Metrics in Frequency Domain

Median Block Weighted metrics are calculated by first doing discrete Fourier transform, extracting their weighted metrics for each block, and then use the max, mean, median value as potential feature[11].

Finally, to get Human Visual System (HVS) based metrics, users need first transform image data into HVS mdoel, details could be found in [16].

References

[1] J. Adams, K. Parulski, and K. Spaulding. Color processing in digital cameras. 18(6):20–30, 1998.

[2] L. Alvarez, L. Gómez, and J. Sendra. An algebraic approach to lens distortion by line rectification. *Journal of Mathematical Imaging and Vision*, 35(1):36–50, 2009.

[3] I. Avcibas, N. Memon, and B. Sankur. Steganalysis using image quality metrics. *Image Processing, IEEE Transactions on*, 12(2):221–229, 2003.

[4] K. Barnard, V. Cardei, and B. Funt. A comparison of computational color constancy algorithmspart i: Methodology and experiments with synthesized data. *IEEE transactions on Image Processing*, 11(9), 2002.

[5] K. Barnard, L. Martin, A. Coath, and B. Funt. A comparison of computational color constancy algorithmspart ii: Experiments with image data. *IEEE transactions on Image Processing*, 11(9):985, 2002.

[6] G. Buchsbaum. A spatial processor model for object colour perception. *Journal of the Franklin Institute*, 310(1):1–26, July 1980.

[7] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

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

[9] A. Dirik, H. T. Sencar, and N. Memon. Digital single lens reflex camera identification from traces of sensor dust. 3(3):539–552, Sept. 2008.

[10] A. E. Dirik, H. T. Sencar, and N. Memon. Source camera identification based on sensor dust characteristics. In *Proc. IEEE Workshop on Signal Processing Applications for Public Security and Forensics SAFE '07*, pages 1–6, 2007.

[11] A. Eskicioglu and P. Fisher. Image quality measures and their performance. *Communications, IEEE Transactions on*, 43(12):2959–2965, 1995.

[12] M. Fairchild. *Color appearance models*. Wiley, 2005.

[13] T. Filler, J. Fridrich, and M. Goljan. Using sensor pattern noise for camera model identification. In *Proc. 15th IEEE International Conference on Image Processing ICIP 2008*, pages 1296–1299, Oct. 12–15, 2008.

[14] G. Finlayson, M. Drew, and B. Funt. Color constancy: generalized diagonal transforms suffice. *Journal of the Optical Society of America A*, 11(11):3011–3019, 1994.

[15] G. Finlayson and E. Trezzi. Shades of gray and colour constancy. In *IST&SID's Color Imaging Conference*, pages 37–41. IST - The Society for Imaging Science and Technology, 2004.

[16] T. Frese, C. Bouman, and J. Allebach. A methodology for designing image similarity metrics based on human visual system models. In *Proceedings of SPIE/IS&T Conference on Human Vision and Electronic Imaging II*, volume 3016, pages 472–483. Citeseer, 1997.

[17] Z. Geradts, J. Bijhold, M. Kieft, K. Kurosawa, K. Kuroki, and N. Saitoh. Methods for identification of images acquired with digital cameras. *Proc. of SPIE, Enabling Technologies for Law Enforcement and Security*, 4232:505–512, 2001.

[18] T. Gloe and R. Böhme. The 'dresden image database' for benchmarking digital image forensics. In *SAC '10: Proceedings of the 2010 ACM Symposium on Applied Computing*, pages 1584–1590, New York, NY, USA, 2010. ACM.

[19] T. Gloe, K. Borowka, and A. Winkler. Feature-based camera model identification works in practice. In *Information Hiding*, pages 262–276. Springer, 2009.

[20] B. Gunturk, J. Glotzbach, Y. Altunbasak, R. Schafer, and R. Mersereau. Demosaicking: Color filter array interpolation in single chip digital cameras. *IEEE Signal Processing Magazine*, 22(1):44–54, 2005.

[21] E. Land. The retinex theory of color vision. *Scientific American*, 237(6):108–128, December 1977.

[22] T. V. Lanh, K.-S. Chong, S. Emmanuel, and M. Kankanhalli. A survey on digital camera image forensic methods. In *Proc. IEEE International Conference on Multimedia and Expo*, pages 16–19, 2–5 July 2007.

[23] B. Lindbloom. Chromatic adaptation. *Bruce J. Lindbloom, Tech. Rep*, 2007.

[24] Y. Long and Y. Huang. Image based source camera identification using demosaicking. In *Proc. IEEE 8th Workshop on Multimedia Signal Processing*, pages 419–424, 2006.

[25] J. Lukáš, J. Fridrich, and M. Goljan. Detecting digital image forgeries using sensor pattern noise. In *Proceedings of the SPIE*, volume 6072, page 15. Citeseer, 2006.

[26] J. Lukas, J. Fridrich, and M. Goljan. Digital camera identification from sensor pattern noise. 1(2):205–214, June 2006.

[27] P. F. Luo Weiqi, Qu Zhenhua and H. Jiwu. A survey of passive technology for digital image forensics. *Frontiers of Computer Science in China*, pages 166–179, 2007.

[28] T.-T. Ng and M.-P. Tsui. Camera response function signature for digital forensics - part i: Theory and data selection. pages 156 –160, dec. 2009.

[29] A. C. Popescu and H. Farid. Exposing digital forgeries in color filter array interpolated images. *IEEE Trans On Signal Processing*, 53 no.10:3948–3959, 2005.

[30] K. San Choi, E. Lam, and K. Wong. Source camera identification using footprints from lens aberration. *Digital Photography II SPIE*, 6069(1):172–179, 2006.

[31] J. van de Weijer, T. Gevers, and A. Gijsenij. Edge-based color constancy. *IEEE Transactions on Image Processing*, 16(9):2207–2214, 2007.

[32] G. West and M. Brill. Necessary and sufficient conditions for von kries chromatic adaptation to give color constancy. *Journal of Mathematical Biology*, 15(2):249–258, 1982.