Digital Image Analysis for Evidence: A MATLAB toolbox

Susan Welford, Dr. Stuart Gibson, Andrew Payne
Forensic Imaging Group, University of Kent,
Canterbury, Kent, CT2 7NZ

July 2011

Abstract

In the last decade, affordable digital camera technology has become widely available, resulting in the proliferation of digital images. The creation, modification and distribution of certain photographic materials is controlled by law in the UK and in many other countries. For example, the production and possession of pornographic images of under 18s is prohibited in the UK by the Protection of Children Act 1978. It is similarly an offense to produce, modify and distribute any image that would be considered useful to a person committing, or preparing to commit, an act of terrorism under the Counter-Terrorism Act 2008. Digital image forensics is the science of determining the source of digital images and detecting the presence of image tampering (e.g. photo forgery). The majority of research in this area has been conducted in the last decade by a small number of experts in the field of digital image processing. An understanding of these methods requires in-depth knowledge of image processing algorithms which most researchers and educators in the broader field of computer forensics do not possess. In this paper we describe our Digital Image Analysis for Evidence (DIAnE) toolbox, written in the MATLAB programming language. Our approach is to utilise the inherent imperfections in image sensors that have previously been shown to produce consistent and unique noise patterns. The toolbox contains code libraries for generating device 'fingerprints' that enable evidential images to be matched to their source cameras and graphical plots to facilitate easy understanding of the resulting correlation data. We believe DIAnE to be the only available MATLAB toolbox that performs this rôle.

1 Introduction

With the enormous variety of camera enabled devices such as video cameras, iPods, phones that are now available, digital images are being captured, stored and transferred across platforms with comparative ease. As a result, forensic examiners are more likely to be asked to provide evidence pertaining to the authenticity and integrity of digital images in a court of law. However, digital image forensics is still a relatively new and rapidly changing field of research and the number of available software implementations of digital forensic techniques is limited. One notable exception is steganalysis; the detection of covertly embedded data (especially in digital images). A good overview of this subject and relevant software is provided by Provos and Honeyman [14]

In this paper we use the term camera verification when referring to the process of matching an evidential image to its source camera and the term forgery detection to mean detecting regions of an image that have been altered with the intent to deceive. Many of the methods proposed offer 'weak' verification in the sense that they are able to determine the make and possibly the model of a source camera from the content of an image file. One approach is to look for image artefacts that result from demosaicing the sensor output to obtain red, green and blue values for each pixel [1]. A variation on this method is described by Çeliktutan et al. [2] who record correlations that occur between adjacent bit planes of an image that result from demosaicing. Proprietary demosaicing algorithms may be common to a number of different camera models produced by a single manufacturer which limits the effectiveness of the technique, as can the presence of JPEG compression artefacts. Choi et al. [3] utilise camera lens aberrations to determine the camera model. Aberrations may be chromatic and caused by inconsistent focusing of different wavelengths, or radial in which case the sharpness of the image is a function of the distance from the centre of the lens. Kharrazi et al. [7] were among the first to suggest a feature extraction based method for verification. The features used belonged to three categories; wavelet domain statistics, image quality metrics and colour features.

An alternative and appealing method for camera verification is the use of image sensor noise. Unlike the methods described above, sensor noise can be used to differentiate between cameras of identical make and model. Kurosawa [8] used sensor imperfections to determine a 'fingerprint' for identifying a source video camera from video film. Fridrich et al. [11] used the inconsistencies with which individual photo-sites in a camera sensor record light intensity. This property is referred to as Photo-Response Non-Uniformity (PRNU) and is the main contributing factor in sensor pattern noise. A

wavelet denoising filter [13] was used to extract the pattern noise from images. Li [10] performed additional image processing on wavelet coefficients to suppress image content that was contaminating the fingerprints. In their later work, Fridrich et al. derived the Maximum Likelihood (ML) estimate of a camera's PRNU from an imaging model that includes multiplicative and additive noise terms.

The techniques described so far are all based on properties of the pixel values. The content of file headers can also contain useful information for identifying the source of an image. Every JPEG file contains a set of instructions for decompression. Farid [4] used JPEG Discrete Quantisation Tables (DQT) for this purpose. Combining DQTs with additional header content was shown to provide greatly improved discrimination between different camera models. Using file header content has the advantage of being robust to benign image processing operations.

Since many of the techniques for verification rely on detecting noise or regular patterns that extend across a whole image, the absence of such features in a localised image region would suggest that an alteration has been made. An image forgery is created in one or more different ways; image content could be copied from one region and moved to another to conceal and object, a person or object from one image might be copied and pasted into another or artistic enhancements/alterations such as localised warping or airbrushing may take place. A method for detecting forged regions based on inconsistencies in an image's pattern noise has been reported [12]. Similarly inconsistencies in chromatic aberration effects may be a sign of image tampering [6]. Other methods include tracing light from a source to objects within a scene to determine contradictions in reflected light, and detecting when an image file's original DQT has been overwritten by DQT that is characteristic of photo-editing software. An analysis of the relative advantages of the methods described in this section is provided by Van Lanh et al. [9].

In this paper we introduce a toolbox for camera verification and image forgery detection based on sensor pattern noise. The following section describes in brief the source and characteristics of image sensor noise. Section 2 describes the functionality of DIAnE from a users perspective with reference to the graphical user interface. Researchers who are interested in using the toolbox for benchmarking their own algorithms will find the list of our MATLAB files in Section 3 useful. We validate our toolbox in Section 4 and conclude with a summary and discussion of our work.

1.1 Sensor Pattern Noise

Inside every digital camera and mobile phone is an image sensor which captures the light intensity of a scene as presented to it by its user. This sensor, commonly formed from either CCD or CMOS technology, comprises an array of light sensitive photosites that each produce a digital signal which is proportional to the number of photons incident upon its surface. The digital signal is passed to an on-board digital image processor in which several operations can take place including white balance adjustment and image compression. The digital image is then written to internal memory or removable media such as Secure Digital (SD) card.

There are a number of potential noise sources within the imaging pipeline of a digital camera. Shot noise is a random noise component that occurs in different pixel locations in each image that is captured. Conversely pattern noise is random only in the sense that there is no spatial structure to its distribution over a single image. It is found consistently in same pixel locations in every image that is captured (by a single camera). Pattern noise consists of fixed pattern noise (caused by dark currents) which is additive and can be observed when no light enters the camera and a multiplicative component, PRNU noise, that dominates when the sensor is illuminated. Since the pattern noise is consistent between images it can be extracted by applying a filter to a sample of images from the same camera to suppress scene content. Averaging the resulting noise residuals removes shot noise and remaining traces of image objects. The distilled pattern noise which is mostly attributable to PRNU can used as a fingerprint that uniquely identifies the camera from which it originated.

2 Using DIAnE Version 1

DIAnE v1 is a MATLAB toolbox that is freely available under the terms of the GNU General Public License. It was developed to illustrate the use of sensor pattern noise in digital image forensic applications. A graphical user interface has been included to enhance the educational benefits of the toolbox, making it easily accessible to teachers and students with minimal previous experience of the MATLAB environment. DIAnE was written using MATLAB Version 7.10.0.499 (release 2010a) on a Windows platform. The MATLAB Image Processing and Wavelet Toolboxes must be installed as our code depends on them. In the remainder of this paper we use the term fingerprint when referring to average noise residual extracted from a sample of training images using the wavelet denoising filter described by Mihcak.

PRNU pattern will mean the noise residual extracted from a *single* evidential (out-of-sample) image using the same denoising filter. In this section we provide an overview of DIAnE from a user's perspective.

2.1 Property Viewer

For a selected evidential image, the property viewer displays Exchangeable image file format (Exif) data. The toolbox saves PRNU patterns with the file extension .evi and camera fingerprints with a .fpt file extension. Both file types contain MATLAB structures for the properties of PRNU patterns and fingerprints respectively. These properties are also accessible in the viewer.

2.2 Building a Camera Fingerprint

DIAnE contains fingerprints for 15 cameras in its standard database. Additional fingerprints may created as described in this section. When building a new fingerprint for a camera, sample images are selected using a standard dialog box. A minimum of 50 sample images are usually required to produce a fingerprint and the advantage of using more than 300 images is negligible (see Section 4). Fingerprints are constructed from 512x512 pixel blocks cropped from the top left hand corner of each sample image. There is some evidence to suggest that a more reliable fingerprint can be obtained by using a central image block. We believe this is attributable to image saturation, and hence the loss of pattern noise, in sample images that contain sky. For camera verification we use the PRNU pattern extracted from the green colour plane only. There are twice as many photosites in the sensor that record green light as there are for either red or blue light. For a specific camera the fingerprint is calculated as the mean noise residual over a sample of images. We found that the maximum likelihood estimate [5] produced only a small improvement in the fingerprint and this step is not included in our code.

2.3 Camera Verification

The first step in verifying the source camera for an evidential image is to extract the image's PRNU pattern from the top left 512x512 image block. The fingerprint for the suspected source camera is selected from the fingerprint database. DIAnE returns the product moment correlation coefficient between the fingerprint and the PRNU pattern of the evidential image. If the evidential image originated has from the suspected source camera, we expect the correlation to be high. Conversely if the selected device is not the source of the evidential image the correlation coefficient should be low. We quantify

this statistically by fitting the correlation coefficients for *non-matches* to a normal distribution and defining a threshold for correlation which if exceeded indicates a match. By default this threshold is set for a correlation coefficient for which the cumulative distribution function is equal to 95% of the area under the normal curve. This is demonstrated graphically when the stem marker on plot lies within the dark shaded region of the normal plot as seen in Figure 1

Multiple evidential images may be compared simultaneously. In this case there are two options for displaying the data; the normal distribution plot with multiple image markers or a correlation plot of the nominal data in which the correlation is plotted against image number.

2.4 Forgery Detection

The objective of forgery detection is to identify region(s) of an evidential image that have been altered (e.g. pasted from another image). The toolbox achieves this by utilising the PRNU pattern to identify regions of interest within the evidential image. As before, we assume that a sample of images from the source camera is available to build a fingerprint. Here a fingerprint is required for the whole sensor (i.e. not only the top left hand corner as in camera verification). Under the assumption that we have no prior knowledge of the region in which the forgery has taken place, the PRNU pattern must be determined for the entire evidential image. The PRNU pattern is then split into 128x128 image blocks each of which is correlated with the corresponding region from the device fingerprint. Our empirical studies suggest that this block size offers a good compromise between localising the altered region(s) and having a sufficiently large image area to obtain a reliable correlation coefficient. The significance of the resulting correlation coefficient is obtained by considering the correlation between the 128x128 fingerprint block and 15 128x128 PRNU pattern blocks the neighbour the region on interest and collectively form a larger 512x512 image block. Neighbouring bocks are assumed to have similar image statistics to the region of interest (although this assumption does not always hold). Correlation coefficients are fitted to a normal distribution as before. Here a large correlation coefficient suggests that forgery has *not* taken place in the 128x128 region of interest. A null hypothesis is postulated which states that no forgery has taken place. By default, the alternative hypothesis (region of interest altered) is accepted if $Pr(r) < 0.05$. Alternatively the 0.05 threshold can be adjusted manually to reduce the False Acceptance Rate (FAR). For visualisation purposes DI-AnE plots a fine grid with 128 pixel spacing to indicate the image blocks and a course grid with 512 pixel spacing to indicate a the membership of a block

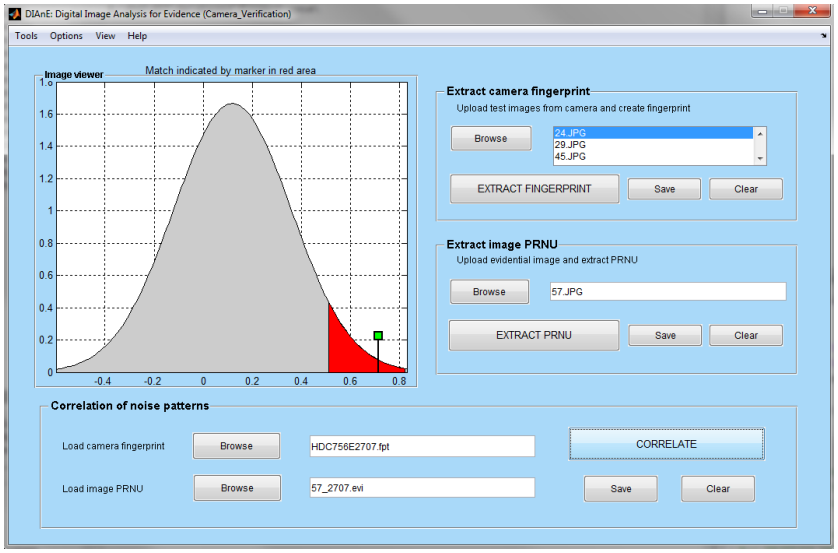

Figure 1: Camera verification interface.

to a neighbourhood (Figures 5&Figure 6). Regions of interest (suspected altered regions) are highlighted with semi-opaque shading. Fingerprints obtained from pattern noise have only modest robustness to re-compression at lower JPEG quality. This is an unavoidable limitation of the technique and to reduce the FAR we recommend saving image forgeries at 100% JPEG quality or in non-lossy format such as png, tiff or bmp.

3 Toolbox contents

DIAnE v1 contains the following MATLAB M-files that are available for use and modification by researchers who wish to compare the pattern noise fingerprinting technique with their own methods for verification and forgery detection. The project URL is:
http://www.kent.ac.uk/physical-sciences/research/fig/diane.html

4 Validation of Toolbox

To validate our toolbox we replicated previous studies and compared the results. The graph in Figure 2 shows the correlation between camera fingerprint and PRNU pattern (both from the same camera) as a function of sample size. To minimise the influence of scene content, the mean correlation coefficient was calculated using PRNU patterns extracted from 15 out-of-sample

DIAnE v1 file	Description
DIAnE_GUI.m	Graphical user interface for camera verification and forgery detection.
blockCorrelation.m	Split image (or image block) into sub-blocks. Correlate each block from the evidential image with all blocks in the camera's fingerprint on a block by block basis - used in forgery detection.
blockVisualisation.m	Overlay grid and shade modified image regions.
createFPT.m	Estimate camera fingerprint from sample of images known to originate from the source device.
waveletDenoising.m	Wavelet denoising filter based on the procedure outlined by Lukac, Goljan and Fridrich in their paper on forensic camera identification. Utilizes Mihcak's method for wavelet denoising.
im2blocks.m	Convert image into a cell array where each cell is an MxN image block of the input image A. Cell indexing uses block matrix convention.
normalPlot.m	Normal plot for camera verification tool.

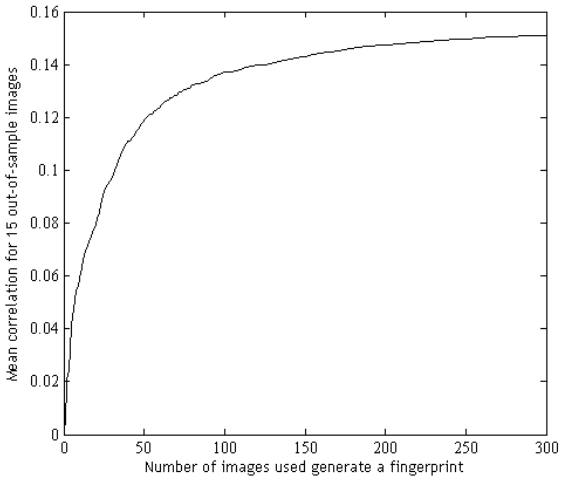

Figure 2: Improvement in fingerprint clarity achieved by increasing the number of sample images used to generate a fingerprint for a Sony Cybershot.

images. In this study the top left hand 512x512 pixel image region was used. A compromise is sought between clarity of the fingerprint and the number of sample images used. As expected the clarity of the fingerprint improves logarithmically with sample size and there is negligible improvement beyond 300 images. In Figure 2 the maximum correlation coefficient achieved after 300 sample images for our Sony Cybershot is approximately 0.15. Lukáš et al. [11] reported a maximum value of 0.09 for an Olympus C765 using the same number of sample images but in TIFF format rather than JPEG. Their results were obtained by averaging the results of 20 out-of-sample images. Reasons for the difference in performance could be due to differences in scene content between the two image sample groups or different noise characteristics of the sensors.

Obtaining the image noise residual using the wavelet denoising filter is computationally intensive with processing time increasing linearly as a function of image area. This is particularly apparent when generating whole image fingerprints for forgery detection. For example, it took 140 minutes to generate a 3888x2592 fingerprint using 100 sample images from the Canon EOS40D on a desktop PC with a 3.07GHz CPU and 3.24GB of RAM. Fortunately this calculation can be made off line.

The ability to discriminate between a source camera and other devices was demonstrated by a simple experiment. Fingerprints were generated for a Canon 40D Digital SLR, Fuji FinePix BigJob, Apple iPhone 4, Sony Cybershot and a Sony Ericsson phone camera using a sample of 100 natural scene

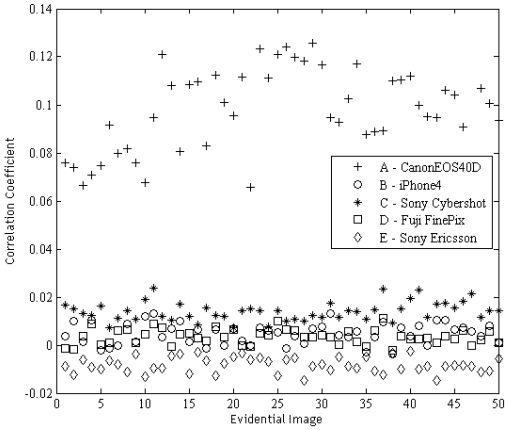

Figure 3: Verification of source camera using 50 evidential images from the Canon EOS 40D.

images in each case. These where correlated with PRNU patterns extracted from 50 evidential (out-of-sample) images known to have originated from the Canon 40D. As can be seen from Figure 3, the resulting correlation coefficients are higher for the source camera than for any other device. These are similar to previously published results [11]. We found that our toolbox was also able to verify the correct source when a comparison is made between two cameras of identical make and model (not shown here).

The plot in Figure 3 indicates a weak correlation of the evidential images with the Sony Cybershot and a weak negative correlation with the Sony Ericsson. We attribute this to stripe artefacts in the fingerprint caused by demosaicing and image compression that takes place within the camera. These artefacts were suppressed by subtracting the pixel row and pixel column means from the noise residual images as proposed by (Chen, Fridrich, & Goljan, 2007). Recalculating the correlation coefficients with this modification improves the ability to discriminate between the correct source camera and other cameras as observed by comparing Figure 3 with Figure 4.

Figure 5 shows a forgery in which a subject has been added to the image scene and Figure 6 shows a duplicate of this image in which the modified image blocks have been detected and shaded by DIAnE's forgery detection tool.

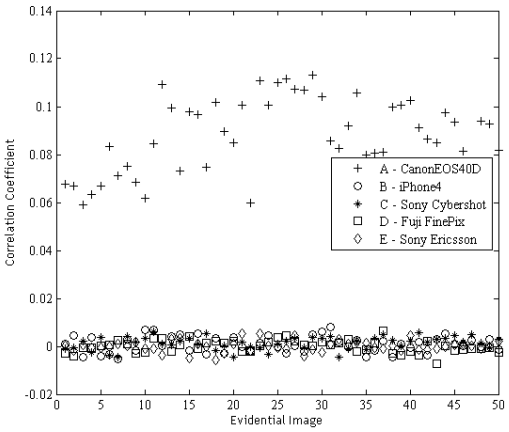

Figure 4: Data from Figure 3 reprocessed to remove stripe artefacts which results in better discrimination between the true source and other cameras.

Figure 5: Image forgery.

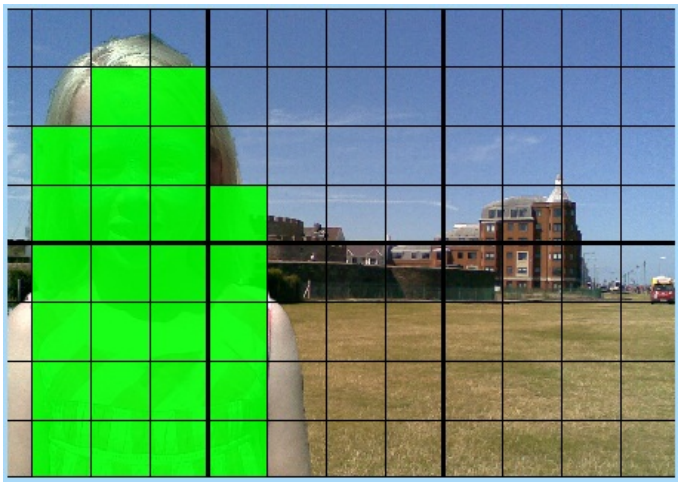

Figure 6: Altered region detected by inconsistency in PRNU pattern.

5 Conclusion

We have described a MATLAB toolbox for digital image forensics that utilises sensor pattern noise. This technique has the advantage of distinguishing between cameras of identical make and model but is only applicable when a sample of training images are available that are known to have originated from the source camera (or equivalently if source camera is available). The toolbox has been validated by showing the repeatability of results obtained in previous studies. Using DIAnE we observed that method for forgery detection was prone to high FAR in regions of an evidential image in which the intensity was low or regains that contained a lot of high spatial frequency content. Of the cameras we tested, models fitted with CMOS sensors produced more distinctive fingerprints than cameras fitted with CCD sensors although this may not be a statistically significant result. It is our intension to extend the functionality of DIAnE in the future to accommodate a broader range of techniques that reflect current interests in the rapidly advancing subject of digital image forensics. We would like to thank the Nuffield Foundation for their support during the very early stages of this work (grant ref: URB/38565).

References

[1] S. Bayram, H. T. Sencar, and N. Memom. Classification of digital camera-models based on demosaicing artifacts. *Digital Investigation*, 5:49–59, 2008.

[2] O. Celiktutan, B. Sankur, and I. Avcibas. Blind identification of source cell-phone model. *IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY*, 3(3):553–566, September 2008.

[3] K. Choi, E. Lam, and K. Wong. Source camera identification using footprints from lens aberration. In *Proceedings of the SPIE*, volume 6069, pages 205–214, 2006.

[4] H. Farid. Digital image ballistics from jpeg quantization. Technical Report TR2006-583, Dartmouth College, Computer Science, 2006.

[5] J. Fridrich. Digital image forensics using sensor noise. *IEEE Signal Processing Magazine*, 26(2):26–37, 2009.

[6] M. K. Johnson and H. Farid. Exposing digital forgeries through chromatic aberration. In *ACM Multimedia and Security Workshop*, Geneva, Switzerland, 2006.

[7] M. Kharrazi, H. T. Sencar, and N. Memon. Blind source camera identification. In *IEEE International Conference on Image Processing*, volume 1, pages 709–712, 2004.

[8] K. Kurosawa, K. Kuroki, and N. Saitoh. Ccd fingerprint method - identification of a video camera from videotaped images. In *Int. Conf. Image Processing*, pages 537–540, Kobe, Japan, October 1999.

[9] Tran Van Lanh, Kai-Sen Chong, Sabu Emmanuel, and Mohan S. Kankanhalli. A survey on digital camera image forensic methods. In *ICME'07*, pages 16–19, 2007.

[10] C-T. Li. Source camera identification using enhanced sensor pattern noise. In *IEEE International Conference on Image Processing*, pages 1493–1496, 2009.

[11] J. Lukas, J. Fridrich, and M. Goljan. Determining digital image origin using sensor imperfections. In *SPIE Electronic Imaging*, pages 249–260, San Jose, CA, January 2005.

[12] J. Lukas, J. Fridrich, and M. Goljan. Detecting digital image forgeries using sensor pattern noise. In *SPIE Conference on Electronic Imaging, Security, Stegaanography and Watermarking of Multimedia Contents*, volume 6072, pages 362–372, January 2006.

[13] M. K. Mihcak, I. Kozintsev, K. Ramchandran, and P. Moulin. Low-complexity image denoising based on statistical modeling of wavelet coefficients. *IEEE Trans. Sig. Proc.*, 6(12):300–303, December 1999.

[14] N. Provos and P. Honeyman. Hide and seek: An introduction to steganography. *IEEE Security and Privacy Magazine*, 1(3):3244, May-June 2003.