

DIGITAL FORENSIC TECHNIQUES FOR THE REVERSE ENGINEERING OF IMAGE ACQUISITION CHAINS

by

THIRAPIROON THONGKAMWITOON

A Thesis submitted in fulfilment of requirements for the degree of

Doctor of Philosophy of Imperial College London

Communications and Signal Processing Group
Department of Electrical and Electronic Engineering
Imperial College London
2014

Statement of Originality

I declare that the intellectual content of this thesis is the product of my own research work under the guidance of my thesis advisor, Dr. Pier Luigi Dragotti. Any ideas or quotations from the work of other people, published or otherwise, are fully acknowledged in accordance with the standard referencing practices of the discipline. The material of this thesis has not been submitted for any degree at any other academic or professional institution.

Copyright

The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work.

Abstract

In recent years a number of new methods have been developed to detect image forgery. Most forensic techniques use footprints left on images to predict the history of the images. The images, however, sometimes could have gone through a series of processing and modification through their lifetime. It is therefore difficult to detect image tampering as the footprints could be distorted or removed over a complex chain of operations. In this research we propose digital forensic techniques that allow us to reverse engineer and determine history of images that have gone through chains of image acquisition and reproduction.

This thesis presents two different approaches to address the problem. In the first part we propose a novel theoretical framework for the reverse engineering of signal acquisition chains. Based on a simplified chain model, we describe how signals have gone in the chains at different stages using the theory of sampling signals with finite rate of innovation. Under particular conditions, our technique allows to detect whether a given signal has been reacquired through the chain. It also makes possible to predict corresponding important parameters of the chain using acquisition-reconstruction artefacts left on the signal.

The second part of the thesis presents our new algorithm for image recapture detection based on edge blurriness. Two overcomplete dictionaries are trained using the K-SVD approach to learn distinctive blurring patterns from sets of single captured and recaptured images. An SVM classifier is then built using dictionary approximation errors and the mean edge spread width from the training images. The algorithm, which requires no user intervention, was tested on a database that included more than 2500 high quality recaptured images. Our results show that our method achieves a performance rate that exceeds 99% for recaptured images and 94% for single captured images.

Acknowledgement

I would like to express my gratitude to different persons who have all contributed to my memorable journey towards a PhD at Imperial College London.

First and foremost, I would like to deeply thank to my PhD supervisor, Prof. Pier Luigi Dragotti, for his constant support, guidance, and understanding. I am grateful to him for offering me the chance to undertake a PhD in London and all the opportunities over the past 4 years. Despite being very busy, he always arranges a weekly meeting for his students with enthusiasm and I have left every meeting wiser and more motivated. He is my role model in how to work professionally. Without his invaluable advices and patience I would not have been able to have come this far.

I would like to thank to my PhD examiners, Mr. Mike Brookes and Professor Anthony T.S. Ho, for valuable suggestions to improve this thesis. Several comments I received from them during my viva examination have lifted my working standard.

I also would like to thank to Dr. Hani Muammar, a post-doctoral colleague for his close collaborations, suggestions, and useful support on writing. His dedication to build our database of professionally recaptured images has great contributions to this research. I would like to acknowledge all colleagues in the REWIND project for sharing knowledge and opportunities.

In addition, I am thankful to the Office of the National Broadcasting and Telecommunications Commission (NBTC), Thailand, for a scholarship support throughout my PhD. In particular, I would like to thank to Mr Takorn Tantasith, the Secretary General of NBTC, and Assoc. Professor Sudharma Yoonaidharma, a former NBTC Commissioner, who initiated the scholarship project. I am also indebted to Narumol Roylapcharoenporn, a scholarship collaborator of the NBTC, for all her supports and understanding.

I would then like to thank all my friends and colleagues from the C&SP research group including Marco, Jon, Toni, Adam, James, Liang, and Shiqiang for suggestions and the good moments we had in London and at conferences abroad. I also wish to thank all Thai friends at Imperial, especially Khoonsake, Ohm, Chedtha, Chon, Tos, Jeep, Moo, Nong, Kendo, Moc, Toey, and Pete for helps, motivation, and constructive discussions.

Last but not least, I am thankful to Wanwisa Chaisaklert, who has accompanied me since the first day of my journey. Her support, love, and encouragement have made me keep walking forward. Finally, I am forever grateful to my Dad and my Mum, Mr Sathian and Mrs Angkana Thongkamwitoon, who have been all inspirations and role models. I deeply thank to their unconditional and incessant support and love.

Contents

Statement of Originality	3
Copyright	5
Abstract	7
Acknowledgement	9
Contents	11
List of Figures	15
List of Tables	19
Abbreviations	21
Notations and definitions	23
1 Introduction	29
1.1 Motivation	29
1.2 Problem Statement	33
1.3 Outline of the Thesis	34
1.4 Original Contribution	36
2 State of the Art in Image Forensics	39
2.1 Previous Related Work	39
2.1.1 Related Work in the General Area of Image Forensics	39
2.1.2 Footprints left on Recaptured Images	49
2.1.3 Related Work in Image Recapture Detection	53
2.2 Modelling of Blurriness Footprints	54
2.2.1 Related Work in Blurriness Measurement	57

2.3	Summary	59
3	Reverse Engineering of Signal Acquisition Chains using the Theory of Sampling Signals with FRI	61
3.1	Introduction	61
3.2	Problem Setup	62
3.3	The Theory of Sampling Signals with FRI	64
3.3.1	Introduction to Signals with Finite Rate of Innovation	64
3.3.2	Examples of FRI Signals	65
3.3.3	Sampling Signals with Finite Rate of Innovation	66
3.4	Reverse Engineering Technique using the FRI Theory	70
3.4.1	Chain Structure Identification and Recapture Detection	70
3.4.2	The Retrieval of Chain Parameters	72
3.4.3	Numerical Simulation	74
3.4.4	Sufficient Conditions for Reverse Engineering	75
3.5	Summary	78
4	Image Recapture Detection using Features from Edge Profiles	79
4.1	Overview of Our Proposed Method	80
4.2	Automatic Edge Detection and Feature Extraction	82
4.3	Dictionary Learning Algorithm	85
4.4	Classification for Recapture Detection	88
4.5	Summary	92
5	Experiments	93
5.1	Database of Recaptured Images	93
5.1.1	Image Capture and Display Equipment	94
5.1.2	Original scene capture	94
5.1.3	Recapture	96
5.2	Experimental Results	98
5.2.1	Experimental Design and Parameter Setting	98
5.2.2	Performance Evaluation of the Proposed Algorithm	99
5.2.3	Blurring Metric as a Feature for Classification	101
5.2.4	Universality of the Algorithm	105
5.3	Performance Comparison	106

5.3.1	Recaptured Detection using LBP, MSWS, and Colour Features	106
5.3.2	Recaptured Detection using Higher-order Wavelet Statistics	108
5.3.3	Comparison Results	108
5.4	Summary	110
6	Conclusions	111
6.1	Thesis Summary	111
6.2	Future Research	113
	Bibliography	116

List of Figures

1.1	Example of (a) original and (b) recaptured images. When the original image is not available it is very difficult to detect by inspection whether the image (b) was recaptured.	30
1.2	Examples of (b) a tampered image which was modified from the (a) original image. The tampered image was then recaptured and most algorithms fail to detect tampering in the recaptured version shown in (c).	32
1.3	A simple chain of image acquisition and reproduction. The original image is obtained by capturing the scene using the first digital camera. The digital image is then displayed on an LCD monitor before it is recaptured using the second digital camera. The original and recaptured images look very similar. .	33
2.1	A chain of operations that produced footprints to a digital image. The footprints might be left by ambient light of the scene, in-camera operations, and digital manipulations applied to the image.	40
2.2	Chromatic aberration is introduced when lens cannot focus all wavelengths of light to the same point on a sensor plane.	40
2.3	Colour filter array (a) in the Bayer pattern. (b) The filter array is generally placed on top of the image sensor in order to filter specific colour component for a single pixel location.	42
2.4	Examples of the histograms built from DCT coefficients of (a) a single compressed image and (b) a double compressed image.	44
2.5	Examples of (a) an original image and its enhanced versions (c) and (e) that have gone through contrast and brightness adjustment. The corresponding histograms of image intensity are shown in (b), (d), and (f) respectively. . . .	48
2.6	The Nyquist boundary for the unfiltered sensor (solid), green (dashed) and red/blue channels (dotted) of the Bayer CFA.	50

2.7	An image recaptured from an LCD monitor with aliasing patterns. Peaks in the 2D FFT response of the recaptured image noise residual are clearly visible.	50
2.8	Simplified models of (a) image acquisition which describes how a sharp edge becomes blurred through an acquisition process and (b) corresponding signal diagram.	55
2.9	Example of a point spread function	56
2.10	Examples of line spread functions measured from (a) Canon EOS 600D with Canon EF-S 18-135mm IS lens and (b) Nikon D40 with 18-55mm kit lens. While performing a slanted edge testing, the focal lengths of the cameras were fixed at 135mm and 55mm respectively.	56
2.11	Two images with different levels of blurriness. (a) original image with sharp edges and corresponding CPBD at 0.57 (b) the blurred version of (a) with CPBD at 0.13.	59
3.1	Problem statement diagram for signals with FRI in the chain of signal acquisition.	62
3.2	Examples of signals with finite rate of innovation (a) stream of Diracs (b) Piecewise Polynomials and (c) 2D Piecewise Constant	65
3.3	Comparative plots of discrete samples (b) $g[n]$ and (e) $\hat{g}[n]$ obtained from acquiring the continuous inputs (a) step input $f(t)$ and (d) the reconstructed signal $\hat{f}(t)$ respectively. The corresponding reconstruct locations based on the samples (b) $g[n]$ and (e) $\hat{g}[n]$ using the FRI sampling theory are shown in subfigures (c) and (f) respectively.	70
3.4	Counter examples when sampling $f_a(t)$ and $\hat{f}_b(t)$ with a sampling kernel $\varphi_a(t) = \varphi_b(t) = \beta_0(\frac{t}{T_2})$	77
4.1	The comparison of line spread functions obtained from slanted edge testing using single captured images and recapture simulation. The cameras used from single capture are Nikon D70s and Canon EOS 600D. The LSF for recapture is created using the recapture combination of the two cameras.	80
4.2	Diagram showing an overview of the training process for our proposed algorithm. Following the dictionary learning process, the learned dictionaries, $\mathbf{D}_{\mathbf{SC}}$ and $\mathbf{D}_{\mathbf{RC}}$, are used to compute a pair of parameters $\{E_d, \bar{\lambda}\}$ for each training image. The classifier is then trained using all pairs of parameters $\{E_d, \bar{\lambda}\}$ which are labelled according to the class of training images.	81

4.3	Overview working diagram of the classification scheme of our proposed recapture detection algorithm	82
4.4	Working diagram of the proposed automatic block-based edge detection algorithm.	82
4.5	Examples of blocks with a binary mask of edges that are detected (a) and discarded (b and c). The block in Figure (a) satisfies all our selection criteria. Blocks in Figure (b) and c) do not qualify because the majority of columns in the block shown in Figure (b) contain double edges and in Figure (c) the number of columns containing an edge is less than βW	83
4.6	The distributions (a) average width of line spread profiles $\bar{\lambda}_{m,n}$ and (b) contrast $\sigma_{m,n}$ were computed from 135 original images. The distributions (c) $\bar{\lambda}_{m,n}$ and (d) $\sigma_{m,n}$ were created using 135 images recaptured from the original images used in building (a) and (b).	84
4.7	The root-mean-squared error computed from K-SVD training with sets of single captured and recaptured images. The errors reduce over 100 training iterations for $L = 1, 3$, and 5.	87
4.8	(a) The root-mean-squared error from K-SVD training using single captured images over 160 iterations when the number of atoms used is varied from $L = 1, 2, 3, \dots, 10$. (b) The optimal number of atoms used is obtained by observing the number of atoms at which the errors begin to converge. This can be estimated from the number of atoms that correspond to the peak of the second derivative of the training error. From our experiment, the optimal value $L = 3$	88
4.9	The criteria for the calculation of the width λ of the spread function. The width is the minimum distance that allows the shape of the edge spread function to be approximated using an estimate of the energy spectral density.	89
4.10	The distributions of average width of line spread profiles computed from (a) single capture dataset and (b) recapture dataset.	89
4.11	The histograms of error E_d computed from (a) single capture dataset and (b) recapture dataset.	90

4.12	A plot of the distributions of features extracted from training images with the average width of spread function ($\bar{\lambda}$) shown on the horizontal axis and the difference of representation error (E_d) on the vertical axis. The hyperplane for recapture classification was obtained using SVM training and is defined as the line that separate the features from the recaptured (star) and single captured (circle) images with minimum classification error. A query image is classified based on the coordinate location of the feature pair, $\bar{\lambda}, E_d$, determined from the image.	91
5.1	Images from the recapture database showing examples of originally captured and recaptured scenes.	95
5.2	Examples of dictionary atoms from \mathbf{D}_{SC} (dark blue) and \mathbf{D}_{RC} (red) obtained after the dictionary learning process.	99
5.3	A plot of the distributions of features extracted from training images with CPBD metric (θ) shown on the horizontal axis and the difference of representation error (E_d) on the vertical axis. The hyperplane for recapture classification was obtained using SVM training and is defined as the line that separate the features from the recaptured (star) and single captured (circle) images with minimum classification error. A query image is classified based on the coordinate location of the feature pair, $\{\theta, E_d\}$ determined from the image. The dotted line ($E_d = 0$) shows the classification threshold when only the feature E_d is used for training.	103

List of Tables

1	Notations	23
2	Definitions	24
3	Symbols	24
4	Device Settings	27
3.1	Frequently used notations	64
3.2	Algorithm 1 Reacquisition detection	71
3.3	Algorithm 2 The retrieval of maximum order R used in interpolation	73
3.4	Algorithm 3 The estimation of the sampling period T_1	74
3.5	The retrieved locations t_k of the discrete samples $g[n]$ estimated using the annihilating filter method	75
3.6	The estimated periods with the average	75
5.1	A table of the digital cameras used in the recapture database.	94
5.2	Distance and camera settings used for recapture, listed by camera models.	97
5.3	The performance of the algorithm in detecting original captured images. The algorithm is tested with the original capture dataset and the results are listed according to the cameras used for capture.	99
5.4	The performance of the algorithm in detecting recaptured images when the algorithm is tested with the recapture dataset. The results according to the camera used for recapture are listed in the rows and the cameras used during the original capture are listed in the columns.	100
5.5	CPBD metrics of images according to cameras used for original capture. The metrics computed before and after image recapture are compared.	102
5.6	The performance of the algorithm in detecting single captured images according to the feature used for blurriness measurement.	103
5.7	The performance of the algorithm in detecting recaptured images according to the feature used for blurriness measurement.	104

5.8	Group configuration for images used to test the universality of the algorithm .	105
5.9	Recapture detection performance of the algorithm for the universality testing .	106
5.10	The comparison of performances of the algorithms in classifying original and recaptured images.	109

Abbreviations

1-D	One dimensional (signal dimension);
A/D	Analogue to digital;
CCD	Charge coupled device (type of digital camera sensor);
CFA	Colour filter array;
CMOS	Complementary metal oxide semiconductor (type of digital camera sensor);
CPBD	Cumulative probability blur detection (blurring metric);
CRF	Camera response function;
D/A	Digital to analogue;
DCT	Discrete cosine transform;
DOF	Depth of field;
DWT	Discrete wavelet transform;
EM	Expectation-Maximization;
ESF	Edge spread function;
EXIF	Exchangeable image file format;
FN	False negative rate;
FP	False positive rate;
FRI	Finite rate of innovation;
JNB	Just noticeable blur (blurring metric);
LCD	Liquid crystal display;
LED	Light-emitting diode;
LSF	Line spread function;
MOMS	Maximal-order-minimal-support (type of function);
MSE	Mean squared error;
PCA	Principal component analysis;
PRNU	Photo Response Non-Uniformity;
PSF	Point spread function;
RGB	Red, Green, Blue (colour components);
SLR	Single lens reflex (type of camera);
SVD	Singular value decomposition;
SVM	Support vector machine;
TN	True negative rate;
TP	True Positive rate;

Notations and definitions

Table 1: Notations

Continuous time signals	
$f(t) \in L_2$	Real continuously defined signals with $t \in \mathbb{R}$, typically included in $L_2(\mathbb{R})$ which is the Hilbert space of finite energy functions.
$f^{(r)}(t)$	r -th derivative of $f(t)$. We note that the zero order derivative coincides with the function itself $f^{(0)}(t)$. We may equivalently use $\frac{d^{(r)}}{dt^{(r)}} f(t)$.
$\langle f(\cdot), g(\cdot) \rangle$	Inner product in $L_2(\mathbb{R})$, defined as $\langle f(\cdot), g(\cdot) \rangle = \int_{-\infty}^{\infty} f(t)g^*(t)dt$ where $g^*(t)$ is the complex conjugate of $g(t)$.
$\ f\ _2$	L_2 -norm of $f(t)$, defined based on the inner product as $\ f\ _2 = \sqrt{\langle f, f \rangle}$.
$f(t) * g(t)$	The convolution of two continuous-time functions $f(t)$ and $g(t)$ is $\int_{-\infty}^{\infty} f(x)g^*(t-x)dx$.
Discrete time signals and vectors	
$a[n] \in \ell_2$	Real or complex-valued discrete time signals with $n \in \mathbb{Z}$, included in ℓ_2 , which is the Hilbert space of square-summable sequences.
$a[z]$	z -transform of the sequence $a[n]$ defined as $a[z] = \sum_{n \in \mathbb{Z}} a_n z^{-n}$.
\mathbf{u}, \mathbf{S}	Vector, and we also use boldface uppercase to indicate matrices, \mathbf{S} .
$\ \mathbf{A}\ _F^2$	Frobenius norm, defined as $\ \mathbf{A}\ _F^2 = \sum_{ij} A_{ij} ^2$.

Table 2: Definitions

Functions	
$\delta(t)$	The delta Dirac $\delta(t)$ is a distribution function that satisfies $\int_{-\infty}^{\infty} f(t)\delta^{(r)}(t-t_0)dt = (-1)^r f^{(r)}(t_0)$, where $f(t)$ is r times continuously differentiable.
$u(t)$	The Heaviside step function is a function defined as such: $u(t) = \begin{cases} 0 & \text{if } t < 0 \\ 1 & \text{if } t > 0 \\ 1/2 & \text{if } t = 0. \end{cases}$ <p>We also note that $\frac{d}{dt}u(t) = \delta(t)$.</p>

Table 3: Symbols

(t_k, a_k)	Innovation parameters of a train of Diracs, where t_k and a_k are the location and the corresponding amplitude of the k^{th} Dirac;
$\beta_P(t)$	1-D B-spline function of order P ;
$\tilde{\beta}_P(t)$	1-D Dual B-spline function of order P ;
η	Number of lines in the block where a desirable edge profile is detected;
θ	Value obtained from blurriness measurement using CPBD blurriness metric;
$\bar{\lambda}$	Average width of line spread profiles for a given image;
$\bar{\lambda}_{m,n}$	Average width of line spread profiles of the block $B(m, n)$;
$\lambda(t)$	Linear filter for interpolation;
ρ	Rate of innovation of an FRI signal;
$\sigma_{m,n}$	Block-based variance of the block $B(m, n)$;
τ_p	Polynomial moments;
$\varphi(t)$	Sampling filter;
$\tilde{\varphi}(t)$	Dual function of $\varphi(t)$;
$\varphi_1(t)$	First sampling kernel in the chain of signal acquisition;
$\varphi_2(t)$	Second sampling kernel in the chain of signal acquisition;
$B(m, n)$	Non-overlapping square block at the m and n vertical and horizontal indices of the block respectively;
$c_{n,p}$	Coefficients for the polynomial reproducing property;
D	Overcomplete dictionary;

\mathbf{D}_{RC}	Overcomplete dictionary trained using line spread profiles from a labelled set of recaptured images;
\mathbf{D}_{SC}	Overcomplete dictionary trained using line spread profiles from a labelled set of single captured images;
$e_t(L)$	K-SVD training error when using L dictionary atoms for sparse representation;
E_d	Difference of approximation errors given by $E_d = E_{SC} - E_{RC}$;
E_{RC}	Representation error when using the dictionary \mathbf{D}_{RC} ;
E_{SC}	Representation error when using the dictionary \mathbf{D}_{SC} ;
$f(t)$	Continuous-time input signal of the acquisition chain;
$\hat{f}(t)$	Continuous-time signal reconstructed from the original discrete samples;
f_s	Sampling frequency;
$g(t)$	Filtered input of the acquisition chain;
$\hat{g}(t)$	Filtered version of the reconstructed signal;
$g[n]$	Original discrete samples obtained from the first acquisition;
$\hat{g}[n]$	Discrete samples obtained from reacquisition;
$h(t)$	Sampling kernel;
h_p	Annihilating filter;
I_O	Original digital image;
I_Q	Query digital image;
I_R	Recaptured digital image;
I_{RC}^j	Digital image at index j in the labeled set of recaptured images;
I_{SC}^j	Digital image at index j in the labeled set of single captured images;
\mathbf{I}_{RC}	Set of recaptured images;
\mathbf{I}_{SC}	Set of single captured images;
K	Number of discontinuities, Number of Diracs;
L	Number of atoms used for K-SVD dictionary training;
L_1	Support of the first sampling kernel $\varphi_1(t)$;
L_2	Support of the second sampling kernel $\varphi_2(t)$;
N	Number of samples;
$P + 1$	Number of moments;

$q[n]$	Query discrete samples;
\mathbf{q}_i	Line spread profile at a row or column i obtained from $\mathbf{q}_i = \mathbf{y}_i^{(1)} / \ \mathbf{y}_i^{(1)}\ _2$;
\mathbf{Q}	Line spread profile matrix;
\mathbf{Q}_{RC}	Line spread profile matrix obtained from a labelled set of recaptured images;
\mathbf{Q}_{SC}	Line spread profile matrix obtained from a labelled set of single captured images;
R	Maximum order of a piecewise polynomial function;
\mathbf{S}	Toeplitz matrix (in Chapter 3);
\mathbf{S}	Line spread profile matrix extracted from the training set (in Chapter 4 and Chapter 5);
T	Sampling period;
T_1	Sampling period of the first acquisition;
\bar{T}_1	Estimated value of sampling period T_1 obtained from the purposed reverse engineering technique;
T_2	Sampling period of the second acquisition;
t_k	Locations of discontinuities;
t_s	Step location of the unit step function;
W	Block size (pixels);
$x(t)$	Continuous-time input signal;
\mathbf{x}_i	Coefficients which provide the best L -sparse representation of the profile \mathbf{q}_i based on the dictionary \mathbf{D} ;
\mathbf{X}	Coefficients matrix obtained from the concatenation of the column vectors \mathbf{x}_i ;
$y(t)$	Filtered input;
$y[n]$	Discrete samples;
$y^{(r)}[n]$	Finite difference of order r of the discrete samples $y[n]$;
\mathbf{y}	Vector of samples $\mathbf{y} = (y[0], y[1], \dots, y[N-1])^T$;
\mathbf{y}_i	Discrete signal of an edge profile at a row or column i in an image block;
\mathbf{Y}	Grey scale values of an image block which contains edge(s);

Table 4: Device Settings

γ	Gamma correction value of an LCD monitor;
f/number	Aperture setting for a digital camera. The f/number is expressed as a ratio of the lens's focal length to the diameter of the entrance pupil. For example, if a lens's focal length is 55 mm and its entrance pupil diameter is 5 mm, the aperture diameter is $f/11$;

Chapter 1

Introduction

1.1 Motivation

The boom of internet activities has contributed to an unprecedented growth of the sharing of multimedia information. We live in a time where digital images distributed over social networks have widespread impact on societies. On-site pictures of plane accidents, sport events, fatal epidemic, and warfare now can reach millions of people at almost the same time they are published online. On the bright side it helps create mutual contributions and raises global awareness on important issues. The prevalent use of digital images has, however, brought new concerns. Now it is also possible for everyone to edit digital images using simple and easy-to-use software on computers or smart phones. From time to time images are deliberately manipulated in order to mislead public perception on crucial subjects, and in some cases for fraudulent purposes. Such tampering has undermined the fidelity status of digital images as the authenticity of images is difficult to prove.

The recent development in image forensics has made it possible to validate the authenticity of digital images. Most techniques are based on *footprints*, artefacts produced and left on images by specific devices or processes. For example, unique sensor noise and distortion patterns are important traces that provide clues about the source of images. Image processing techniques such as rotation, resizing, splicing, and medial filtering also leave artefacts which are evidences of image manipulation. While digital techniques for tampering have become more sophisticated, the state-of-the-art image forensic techniques can also maintain high detection precision against image forgery.

In addition to image forgery detection, the detection of near-duplicate images is emerging as an important forensic problem. A frequent misuse of image duplication is the theft of intellectual properties. Usually a near duplication is obtained because a person who intends to copy images does not have access to the original digital files of the images. For example, the

person might want to obtain printed pictures on magazines or the images shown on screens. One way to obtain the duplicates is to recapture images using a scanner or a digital camera. Digital cameras today are capable of delivering high resolution images with pleasing colour and tone reproduction at relatively low cost to the consumer. Moreover, with the widespread availability of high quality colour ink-jet printers and liquid crystal display (LCD) devices, images can be easily reproduced by recapturing the printed or displayed image with a digital camera. If a high quality digital camera is used, such as a DSLR, and the image is recaptured from a good quality print or a high resolution LCD monitor, then a recapture with high fidelity can be obtained. An example of a duplicated image which was recaptured using a SLR camera is shown in Figure 1.1(b). When the original image, Figure 1.1(a), is not available, it is very difficult for humans to determine whether the image is a recaptured version. In addition, prior information about the image is generally limited. Only available information about the image is obtained from Exif (Exchangeable image file format) data which is the metadata embedded within the image file itself. In general it can be viewed using file property options on most operating systems. The Exif data provides information about the date the image was taken, the device used to capture the image, and camera parameters used during the capture. This data, however, provides only the information about the device used in the most recent capture. It is impossible to retrieve the history of the image before the most recent capture using given prior information. According to the experiment conducted by Cao and Kot [1], their test subjects could detect recaptured images by observation with a success rate of approximately 51%.

Digital images today are not only easy to produce and tamper, they also “live” longer. With the advances in cloud network and mass storage technologies, digital images can be

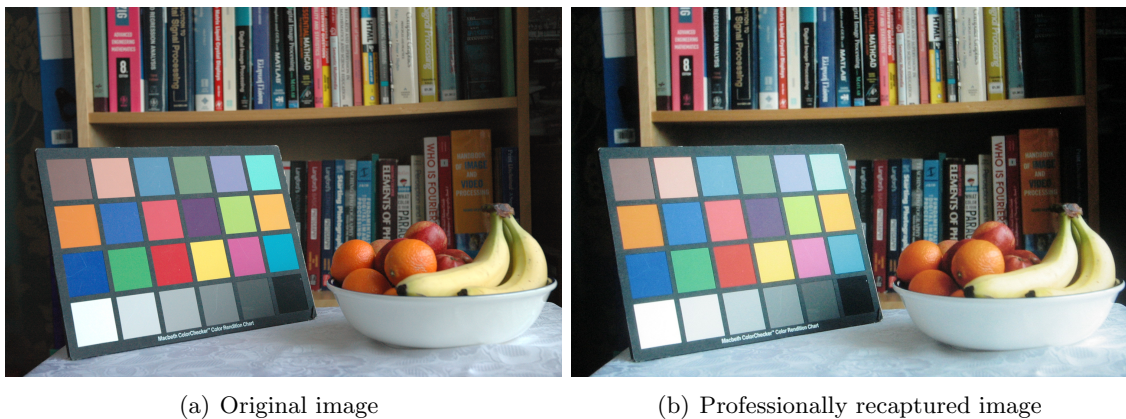


Figure 1.1: Example of (a) original and (b) recaptured images. When the original image is not available it is very difficult to detect by inspection whether the image (b) was recaptured.

stored over a longer period of time and are easily accessible online. It is therefore possible that images are used and processed more than once over their lifetime. This fact has made digital forensic problem more complicated. Given an unknown image the question on what processes the image has gone through is generally of interest. With the existing digital forensic techniques it might be possible to provide some satisfying answers based on the links between footprints left on images and the corresponding past processes. The issue, however, is that the useful artefacts are corrupted over a complex chain of signal operations. Consider, for instance, the case that an image is recaptured after manipulation. The traces such as noise and compression artefacts might be destroyed when the image is displayed on a low resolution screen before the recapture. In addition, the qualities of lens and image sensor also determine the survival of footprints over image acquisition. Noise and processing artefacts are often filtered out when images are recaptured using a low quality camera or deliberately removed by professional recapture settings. In Figure 1.2 we illustrate a counter forensic technique which exploits the fact that a tampered image can evade major forgery detection techniques when the image is recaptured. The original image in Figure 1.2(a) shows a calm beach scene on a sunny day. With the touch of image editing software, the tampered photo in Figure 1.2(b) depicts the scene of landing invasion with battle ships, paratroopers, and intimidating tanks added from other images. If the image was posted online it would be detected later as a hoax picture because footprints left on the image by editing techniques would be detected by most forgery detection algorithms.

Next we displayed the tampered image on a LCD screen and recaptured it. The resulting image is shown in Figure 1.2(c). Again the traces of recapture are not obvious in this image and one might find it difficult to detect the recapture by visual inspection alone. More importantly, the recapture has destroyed or distorted all the trace of manipulations which are typically used for forensic analysis. It also adds new consistent footprints across the image and makes the image look like an original. We later authenticated the image and found that it could spoof major forgery detection techniques. For example, we tested the image with Izitru [2], one of the most well-known and leading image authentication engines publicly available online. According to the disclaimer stated on the website, the engine at the present time cannot detect recaptured images, and in fact, the test result for manipulation detection was negative and the tool identified the test image as an original image.

This authentication result suggests that image recapture is one possible way to bypass the state-of-the-art forensic techniques from detecting tampered images. It highlights the

importance of an image recapture detection algorithm and a reverse engineering technique that can retrieve the processing history of digital images.



(a) Original image



(b) Tampered image



(c) Recaptured version of the tampered image

Figure 1.2: Examples of (b) a tampered image which was modified from the (a) original image. The tampered image was then recaptured and most algorithms fail to detect tampering in the recaptured version shown in (c).

1.2 Problem Statement

As highlighted before, in its lifetime a digital image may go through a series of processing stages and the issue is to retrieve the whole processing history of the image. In this thesis we focus only on one type of chain based on image acquisition and reconstruction. Figure 1.3 shows this type of chain. In the figure a scene is first captured using the first digital camera. This process is called *image acquisition*, where the spectral energy of light is converted into digital data. The original digital image, I_O , is obtained. After that the image is displayed on an LCD screen and viewed by observers. At this stage the image is converted to an analogue signal again, in form of light spectrum corresponding to the image signal. This process is referred to as *image reproduction*. The reproduction can be also carried out through image printing.

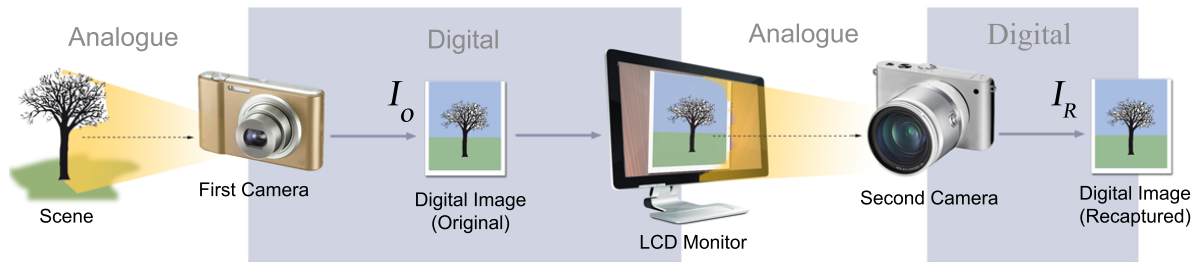


Figure 1.3: A simple chain of image acquisition and reproduction. The original image is obtained by capturing the scene using the first digital camera. The digital image is then displayed on an LCD monitor before it is recaptured using the second digital camera. The original and recaptured images look very similar.

Next we assume that the image is recaptured by one of the observers using a digital camera. The image thus is acquired to a digital format again, given by the recaptured image I_R . Now the image I_R is a near-duplicated version of the genuine image I_O . We have shown in the last section that, in practice, two images are almost identical when observed by the human eye.

Suppose we are given one of these two images without any prior information. The natural question is how can we authenticate the query image I_Q using only information present in the query image? That is, how can we verify that the query image I_Q is the original image I_O , or actually the recaptured image I_R in Figure 1.3 ?

The second problem that is related to the first question is how much can we say about the history of the query image I_Q , in particular when we have detected that the image is from a chain. As for the example in Figure 1.3, how can we retrieve key information which

is related to the properties of the LCD monitor, and the first camera? The answers lie in the traces left by the past operators which are still present in the recaptured image. It is therefore important that those footprints survive through the second acquisition. In practice several factors determine the presence of the past footprints on the recaptured image. It can be the properties of the footprints themselves, or the specific conditions under which the image was captured. Our third question then is what conditions determine the success of the reverse engineering approach?

In this thesis we present methods which address these problems. We consider blurring patterns introduced to images during acquisition as our footprints, and study how these footprints change after each processing stage in the chain in Figure 1.3. It is anticipated that the reproduction and recapture processes would leave new footprints or alter the existing footprints in some ways. We propose new methods to solve the problems in two different approaches. We first describe, in our theoretical framework, how signals are changed at different stages of the chain using a 1-D signal model. We then propose an approach for signal reacquisition detection and an algorithm to reverse the processes in the chain model. An attempt to derive sufficient conditions required for successful reverse engineering is also presented.

Next we apply the framework in the context of a practical image recapture detection problem. We focus on addressing the first statement problem with the aim to maximise the success rate of detection. In that part of the thesis it is shown how to develop the algorithm that is robust to different capture conditions and universal to different types of devices.

Note that in this research we are focusing on a problem of recapturing images from an LCD monitor. Although some of the features and techniques discussed in this thesis can be also useful for detecting printed-and-scanned images, different approaches are required. This is because the footprints left on printed materials are different from the footprints introduced by capturing images from an LCD monitor. In addition, the method to generate digital images from a scanner head is different from the method that a camera captures images. We therefore consider image acquisition chains using a printer and a scanner as a different problem.

1.3 Outline of the Thesis

This thesis has five additional chapters, which are briefly outlined below:

In the first part of Chapter 2, we provide a literature survey of image forensic techniques and image recapture detection approaches. The survey is presented according to the types of

footprints left on digital images and types of features used for recapture detection. The second part of the chapter describes how we model blurriness footprints introduced to images by an acquisition process. We discuss how these footprints are important for our forensic analysis. The literature review on the related research on blurriness metrics is also presented in this chapter.

In Chapter 3, we propose a theoretical framework for reverse engineering of signal acquisition chains. We create a 1-D chain model using the A/D and D/A operators to simplify the chain of signal acquisition and reproduction. Next the theory of sampling signals with finite rate of innovation (FRI) is introduced. We then present our proposed techniques for reacquisition detection and the retrieval of chain parameters. To the end, we derive the sufficient conditions that allow reacquisition detection and reverse engineering techniques to produce the correct results.

A novel algorithm for image recapture detection is presented in Chapter 4. We propose a more concrete technique to determine the acquisition history of digital images, based on detecting image recapture. The overview of the algorithm is first introduced. Next we describe how to extract line spread profiles using an automatic edge detection algorithm. Next a method for dictionary training using the K-SVD approach is presented. We discuss how to train two overcomplete dictionaries to learn footprint patterns left on single captured and recaptured images. Then two distinctive features and a method to build a classifier for recapture detection are presented.

The experimental results on the performance of the proposed algorithm are presented in Chapter 5. First the parameter settings and image database preparation are briefly described. Next this chapter discusses how we conducted four experiments to test different aspects of the algorithm. In the first experiment the classification precisions of our proposed algorithm in detecting original and recaptured images are shown according to different camera models and chain combinations. In the next experiment, we compare the performances of the algorithm when using our proposed distinctive feature and blurriness metric. We show how our proposed feature is more suitable for our recapture detection application than one of the state-of-the art blurriness metric. The third experiment presents how we test the universality of the algorithm. Finally, we conduct an experiment to compare the performances of our algorithm with two well-known benchmark methods.

Finally, Chapter 6 concludes the thesis. We summarise the achievement of the research and highlight possible directions for future research.

1.4 Original Contribution

The original contributions of this thesis are presented in Chapters 3, 4 and 5. In Chapter 3 we propose a novel technique for reverse engineering image acquisition chains using finite rate of innovation principles. Our main contributions are as follows:

- An analytical method to describe footprints left by sampling and interpolation operators in terms of signals with FRI.
- A theoretical framework for the reverse engineering of signal acquisition and reproduction chains. This leads to an algorithm for detecting signal reacquisition and a technique to retrieve important parameters in the chains.
- Sufficient conditions required for our theoretical framework to estimate correct answers.

In Chapter 4 we propose a method for detecting digital images that have gone through image recapture. The innovative features of the algorithm are as follows:

- A novel algorithm for image recapture detection based on blurriness patterns on edges. The algorithm has very high detection precision and is robust to different types of scenes, different types of cameras, and different brightness conditions.
- An automatic block-based algorithm to extract features from image edges. The proposed recapture detection algorithm is thus entirely automated. This helps reduce errors from human bias and allows the algorithm to work continuously with large database.
- The algorithm is universal to different type of cameras used. Therefore there is no need to obtain training features from all possible cameras in the market to maintain the detection. The features used provide high classification performance when compared to the feature based on existing blurriness metrics.

To the best of our knowledge, Chapters 3, 4, and 5 of this thesis contain the original research work which led to the following publications:

- T.Thongkamwitoon, H. Muammar, and P.L. Dragotti, "An image recapture detection algorithm based on learning dictionaries of edge profiles", *submitted to IEEE Transactions on Information Forensics and Security*, June 2014.

- T.Thongkamwitoon, H. Muammar, and P.L. Dragotti, "Robust image recapture detection using a K-SVD learning approach to train dictionaries of edge profiles", in *Proc. IEEE International Conference on Image Processing (ICIP)*, 2014.
- T.Thongkamwitoon, H. Muammar, and P.L. Dragotti, "Reverse Engineering of Signal Acquisition Chains using The Theory of Sampling Signals with Finite Rate of Innovation", in *Proc. IEEE International Conference on Acoustic, Speech, and Signal Processing (ICASSP)*, 2013.
- T.Thongkamwitoon, H. Muammar, and P.L. Dragotti, "Identification of Image Acquisition Chains Using a Dictionary of Edge Profiles", in *Proc. of European Conference on Signal Processing (EUSIPCO)*, 2012.

In addition to the publications, the following prizes have been awarded to this research project:

- The best demo/poster presentation, *IEEE SPS Italy Chapter Summer School on Signal Processing*, 2013.
- The Educational Awards for Excellence 2013, Highly Commended (Second Place) for Engineering and Technology, *Anglo-Thai Society and The Royal Thai Embassy in the United Kingdom*, 2013.

Chapter 2

State of the Art in Image Forensics

In this chapter we provide a literature survey of image forensic techniques. Given the focus of this thesis on image recapture detection, particular emphasis is given to the analysis of traces left on images during the recapture process. In this context it is noted that recapture introduces further level of blur in images and that this footprint can be very useful for recapture detection. For this reason we also present a mathematical model of the blur introduced by cameras during the acquisition process and discuss blurriness metrics. The chapter is organised as follows:

In the next section we review the state-of-the-art in image forensics and image recapture detection. In Section 2.2 we provide an in-depth analysis of the footprints left on recaptured images. Finally, given the importance of level of blur present in images, we present in Section 2.3 a model for the blur introduced during acquisition and a survey of methods developed to measure blurriness in images.

2.1 Previous Related Work

2.1.1 Related Work in the General Area of Image Forensics

A common approach in image forensics is based on the recovery of footprints left on digital images. Footprints are artefacts left on the signal after a processing stage and thus carry important information corresponding to the past processing carried on the signals. Figure 2.1 provides an example of a typical chain of operations that are applied to images and that leave different types of footprints. In this section we review how footprints are used to infer the processing history of an unknown digital image. The techniques are grouped into 5 categories according to the type of footprint involved.

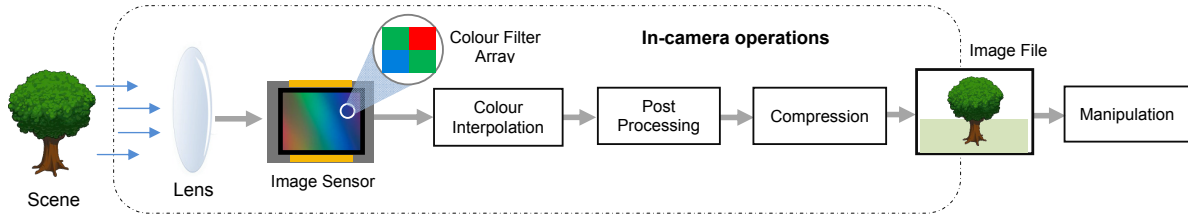


Figure 2.1: A chain of operations that produced footprints to a digital image. The footprints might be left by ambient light of the scene, in-camera operations, and digital manipulations applied to the image.

Intrinsic property of acquisition devices

Artefacts introduced during the acquisition stage are the first footprints left on digital images. This type of footprints is often used to trace sources of the images, that is, to detect the device used to capture an image. As illustrated in Figure 2.1 when an image is acquired, the properties of the image are altered by optical lens. The footprints left by chromatic aberration of lens are studied in [3] and [4]. The aberration is a form of distortion which is introduced when the lens cannot focus all light colours to the same convergence point. The reason is different colours of light propagate in the lens with different speed and are focused at different locations on the sensor as shown in Figure 2.2. Johnson and Farid [3] presents a method to detect image tampering based on inconsistency of chromatic aberration across images. The features introduced in that paper are then used in [4] to create a method to identify cell phone source of images based on the Support Vector machine (SVM) training approach.

Lens radial distortion is used as a footprint in [5] in order to identify the source cameras. When straight lines in the object space are distorted by the radial effect, they are observed as curved lines in images. In that paper the degrees of distortion introduced by specific cameras

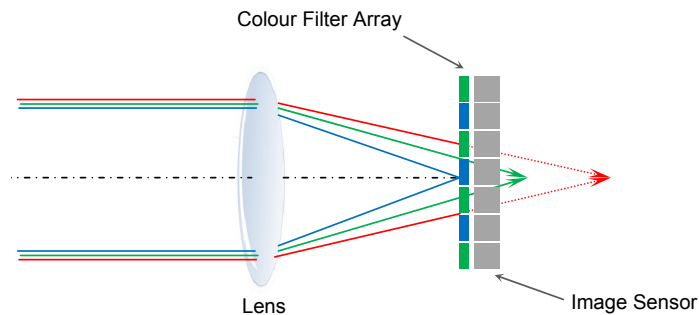


Figure 2.2: Chromatic aberration is introduced when lens cannot focus all wavelengths of light to the same point on a sensor plane.

are measured and used to build a classifier for camera identification.

The noise patterns present on digital images are important footprints left by the cameras that produced the images. Noise patterns are typically determined by a number of factors including imperfections in CMOS/CCD production and the variations of electromagnetic or thermal properties of substrate materials. These intrinsic variations cause different sensors to have different sensitivities to a uniform level of light intensity. The measurement of these variations is referred to as Photo Response Non-Uniformity (PRNU). The PRNU patterns are generally unique to image sensors and were first used to identify source of images in [6] and [7]. Footprint modelling based on noise patterns is extended in [8] and [9] in order to increase the robustness of the algorithm and reduce the sensitivity in high spatial frequency regions respectively.

Gou *et al.* [10] use features based on statistical properties of noise. The basic insight is that manipulations will change the statistics of noise in specific ways. The detection is based on an SVM classifier trained using noise features from original and tampered images. Their method has been extended in [11, 12, 13] in order to deal with larger datasets and more types of cameras and cell phones. Unique dust patterns on image sensors are employed in [14] to identify source cameras.

Camera response function (CRF) can be described by a transfer function that converts light intensity to image signals. Lin *et al.* [15] propose the estimation of a camera response using a non-linear function. The estimation technique is later used in [16] to map camera responses to specific devices, and the authors propose a tampering detection algorithm based on the inconsistencies of camera responses in images. That is, inconsistencies highlight the presence of tampering.

Footprints from Colour Filter Array (CFA) interpolation

In an acquisition process colour intensity is converted to an image signal. In practice, the capture of all red, green, and blue colour components at a single pixel requires substantial precision in manufacturing to align the three planes of colour sensors to the same location. Most consumer cameras, in fact, capture a single colour per pixel using only one plane of image sensor. Since each pixel of most CCD/CMOS sensor detects light intensity with no colour specificity, a colour filter array (CFA) is required in order to separate light to a specific colour channel. The CFA is typically placed on top of the image sensor. One of the most common patterns for the CFA is the Bayer filter, shown in Figure 2.3(a). Figure 2.3(b) illustrates how

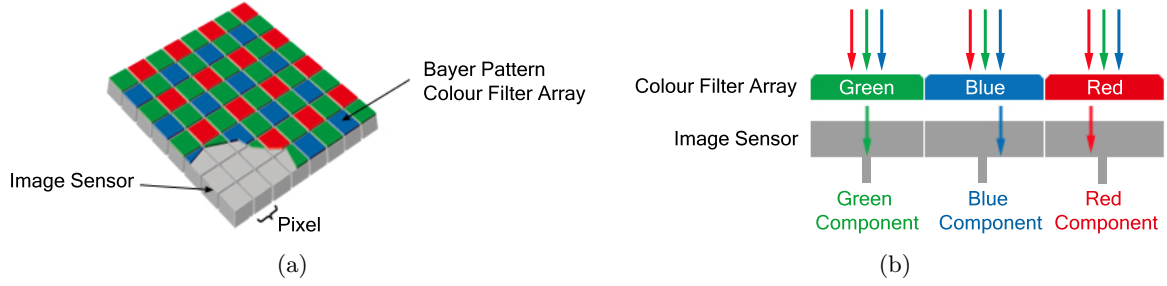


Figure 2.3: Colour filter array (a) in the Bayer pattern. (b) The filter array is generally placed on top of the image sensor in order to filter specific colour component for a single pixel location.

separated colour components are obtained using the colour filter array.

In order to reproduce the correct colour at a certain pixel, it is necessary to reconstruct the two missing colour components using the neighbouring pixels. This process is referred to as demosaicing or CFA interpolation. Such interpolation introduces statistical correlation across pixels. Moreover, the correlations are periodic because of the periodic patterns of the colour filters. Popescu and Farid [17] use an Expectation-Maximization (EM) algorithm to detect seven different types of demosaicking techniques. The detection is based on the computation of the probability that a pixel has been generated using a certain interpolation given its neighbours. The paper suggests that image tampering can be detected when the pixels or regions have low correlation of periodic patterns with their neighbouring pixels. The same principle is used in [18] to classify different types of correlations which are unique to camera models. A Support Vector Machine (SVM) classifier was trained for camera model identification.

The traces from CFA interpolation and PRNU noise patterns are used jointly in [12] in order to predict source and camera models used to capture images. In [19] an algorithm to localise tampered region based on CFA demosaicking artefacts is proposed. The research investigates the presence of fine demosaicking artefacts at 2×2 block level. The algorithm employs Bayesian statistics to generate the spatial confidence map based on the presence of the artefacts. Since the fine artefacts are likely to be removed by manipulations, the regions of the image with low probability to detect the artefacts will be classified as the tampered regions. Image tampering detection algorithm using artefacts from CFA interpolation and demosaicking have been further improved in the following recent papers [20, 21, 22, 23].

Footprints from image compression

Most digital images are stored in a compressed format, typically in JPEG. Different manufacturers often have their own different compression settings in order to balance between compression ratios and image quality. Fan and de Queiroz [24, 25] proposed a method to retrieve the compressions history of images through the estimation of the quantisation table used during JPEG compression. The papers also discuss methods to estimate JPEG quantisation step sizes through the observation of DCT coefficients of images. The key insight is that before quantisation, the histograms of DCT coefficients of images are usually continuous. Quantisation typically introduces a comb-like pattern to the histogram because the DCT coefficients are redistributed to discrete bins. The uniform space between bins corresponds to the step size of the quantiser. In [26], Qadir *et al.* show in the experiments that the distribution of the 1st digit probability of Discrete Wavelet Transform (DWT) coefficients of images follows the Benford's law [27]. One way to obtain the first digit of a coefficient is to represent the decimal floating-point coefficient as a fixed number of significant digits and scaled using an exponent. The first digit of the coefficient is the most significant digit of the fixed number. The authors use this findings to develop a technique to detect JPEG2000 compression and estimate compression rate.

Although multiple JPEG compression is not a conclusive sign of image tampering, it is an indication that images have been resaved. The resaved images are suspicious because they might have been manipulated before being saved again. In [28] Lukas and Fridrich suggest that double compression introduces peaks in the DCT histogram. Figure 2.4(a) shows a histogram of DCT coefficients of an original image which is compressed once. When the image is compressed again, the peaks or comb-like patterns would be introduced to the histogram as shown in Figure 2.4(b) due to repeated quantisation. When the step size of the first JPEG quantisation is larger than the step size of the second compression, total number of histogram bins would increase. Some bins in the histogram would be empty. In contrast, if the step size of the second quantisation is larger, total number of histogram bins would decrease. Some bins in the new DCT histogram would receive more samples from their neighbours in the original histogram. In both cases, the new histogram would have a comb-like pattern.

The comb-like pattern would introduce periodicity in the histogram of DCT coefficients of images. Popescu and Farid [30] propose a double JPEG detection algorithm based on the detection of this periodicity on the Fourier domain. The authors also claim that the characteristics of the periodicity depend on the quality parameters. On this basis it is possible

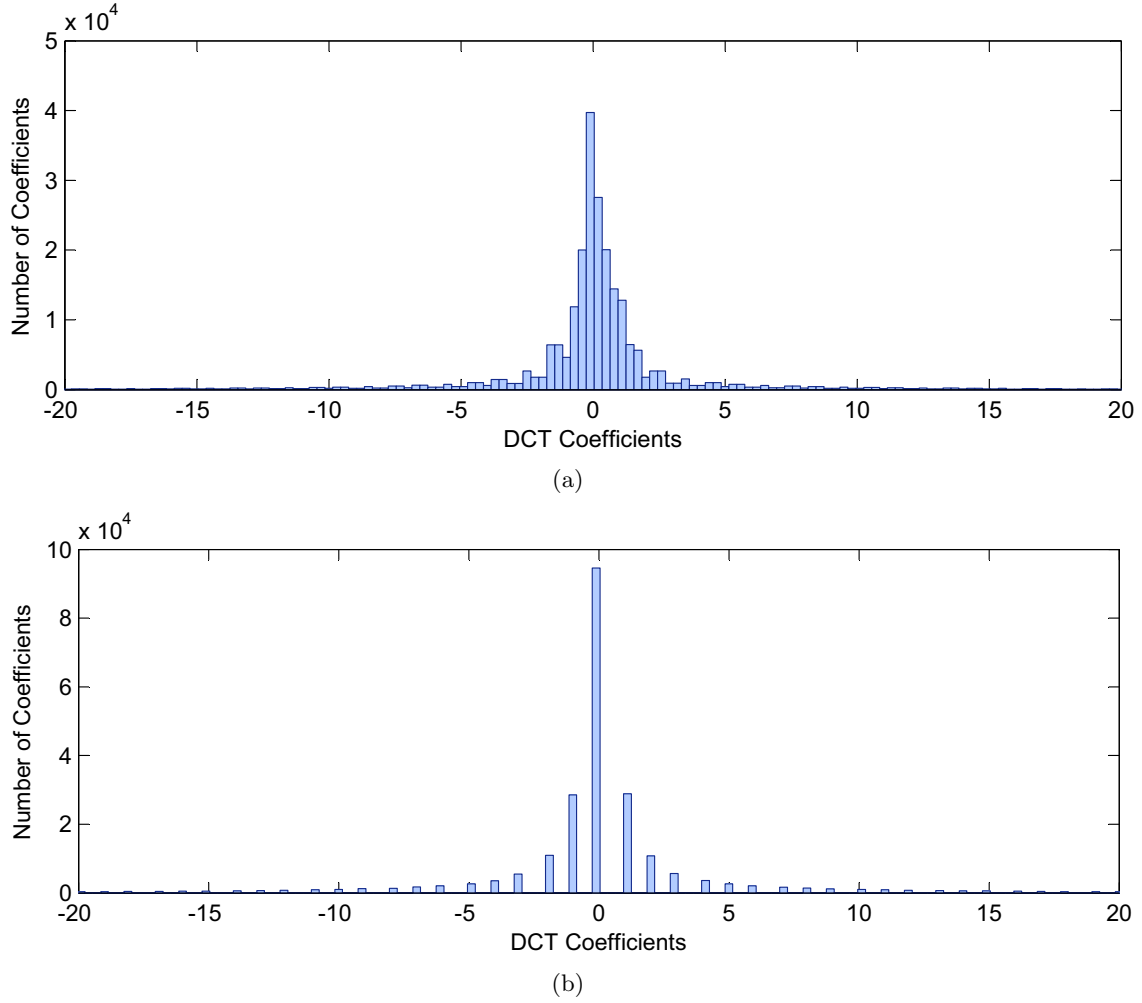


Figure 2.4: Examples of the histograms built from DCT coefficients of (a) a single compressed image and (b) a double compressed image. The histograms are created using the Steganographic software library provided in [29]

not only to detect double JPEG compression but also to estimate the quality parameters used. This principle is used by He *et al.* [31] to detect localised footprints of double JPEG compression. Further methods to detect double JPEG compression have been presented in recent literature [32, 33, 34, 35]. In [36], Milani *et al.* propose a technique to predict number of compression stages applied to images using the Benford’s law. The prediction is based on distribution of the first significant digits of DCT coefficients.

Blocking artefacts are typically introduced by lossy JPEG compression. At high compression rate, high frequency details are removed by quantisation. When images are decoded, they are reconstructed from the remaining low frequency DCT coefficients. The block patterns are left because pixels in the same block share the same average colour value. In [37], the algorithm for forgery detection based on inconsistencies of blocking artefacts is proposed. The

inconsistencies are produced because images from different sources are likely to have been encoded with different parameters and therefore have different blocking patterns. A similar principle is used in [38] by Li *et al.* According to their paper the manipulation using copy-paste, or resizing techniques might cause block grid mismatch. The algorithm they developed detects manipulation based on this inconsistency.

Geometry and physics-based footprints

Environment of image scenes such as lighting conditions, shadow, and colour temperature when images are captured are also important traces left on images. When manipulation is performed using images from different sources, the resulting tampered images often does not look very convincing because of lack of consistency of these scene conditions. Forgery detection using inconsistency of 3D lighting was first explored by Johnson and Farid [39]. The authors propose a method to predict light source directions in the scenes. The tampering is detected when lighting directions on objects in images are not consistent with those of scene context. The method is improved in later work by the same authors [40]. The paper describes how to estimate complex 3-D lighting condition using low-parameter representation to reduce computational complexity. They then extended the method to cover the case of complex lighting environment with the assistance of 3-D human head model [41] and spotlight reflections in human eyes [42].

Riess and Angelopoulou [43] introduce features based on illumination conditions such as flash light, indoor lighting, daylight lighting when the scenes are captured. Given an image, the method creates image map which comprises segmented regions according to colour similarity. The algorithm then investigates the consistency of colour temperatures of each region with the dominant illuminating colour of the image. If the inconsistency is detected, it is likely that the image was tampered. In [44, 45, 46] the inconsistencies of shadow geometry, sizes, and directions have been used to detect tampered images.

Geometrical distortion and image perspective describe how scenes were captured. When the geometrical shapes of objects are not consistent with the perspective of images, it is possible that the image of those objects were added later by manipulation. In [47], an algorithm to detect manipulation on pictures of signs and billboards is proposed. The technique is based on the detection of inconsistency of 3-D perspective mapping of characters onto a planar surface. Kakar *et al.* [48] create a model to estimate motion blur in images and image manipulation is detected when the motion blur patterns are not consistent.

Unique software-based tampering traces

Image tampering often leaves footprints of specific photo-editing techniques. Such techniques include cloning, cut and paste, shape transform, splicing, and a number of filtering operations. One of the most common and simple techniques is region duplication. The purpose of duplication is to replace a picture area containing a person or objects using a portion of pixels copied from some background region within the same image. Algorithms to detect basic pixel cloning have been proposed in the literature [49, 50, 51]. The detection is based on the correlation between the original image segment and newly created region using a cut-and-paste technique. In [49], Fridrich *et al.* first divide an image into a number of blocks and compute a block-based discrete cosine transform (DCT). The DCT coefficients of all blocks are compared in order to search for block matching. If the blocks are matched, it is likely that some regions of the image are duplicated. Popescu and Farid [50] adopt this method and propose an additional feature using a principal component analysis (PCA) on small image blocks. Both DCT and PCA features are used to create more robust algorithm. These methods, however, focus on simple duplication problems and did not cover the case that the pasted region was modified before. More recent approaches for a copy-and-move problem which are robust to rotation, lighting and geometrical distortions have been proposed in [52, 53, 54, 55].

Geometrical transformations are often used in region duplication to create a convincing forgery. The transformations include rotation, resizing, and stretching and are often applied to the cropped portion of image before it is pasted to another file. All of these transformations require a resampling process. For the upsampling or up-scaling case, some pixel values are obtained by interpolating adjacent pixels. Such interpolation introduces correlation among neighbouring pixels. Popescu and Farid [56] apply the technique previously used in their work [17] to detect correlation introduced by resampling operation based on the expectation maximization (EM) algorithm. The probability that a specific pixel has been resampled is obtained using the result of linear prediction for each pixel value and the probability that the pixel is correlated with its neighbours. A probability map is created using probability values from each pixel. From this map it is possible to identify the region with high possibility of being resampled. This method is, however, computationally expensive. Kirchner [57] improves this method using a fixed linear filter and achieves faster and more reliable algorithm. Alternative algorithms to detect image resample are presented in [58, 59]. Specifically, in [58] periodicity is detected using the average of an image's second derivative along its rows or columns, in [59] instead, the detection method is based on the variance of the signal's derivative.

A cut-and-paste forgery often uses image sources that have different brightness levels. In such a case, brightness and contrast enhancement are needed to equalise the intensities of the newly pasted region to the level of the whole forged image. Contrast adjustment however alters the histogram of the distribution of intensities. In [60], Stamm and Liu suggest that the contrast enhancement creates spike and gap patterns on the redistributed histogram. The patterns are introduced because the range of image intensities is expanded thus the newly mapped values are redistributed to a wider range of histogram bins. The expansion often leaves some empty bins or creates a peak pattern on the histogram. An example of an original image is shown in Figure 2.5 (a). Figure 2.5 (c) and Figure 2.5 (e) show the corresponding enhanced images after the original image has gone through two different degrees of contrast adjustment. The expansion of the intensity range can be observed in the histograms in Figure 2.5 (d) and Figure 2.5 (f). This is in contrast with the histogram of original image in Figure 2.5 (b) which is smooth and centred at the mid-tone area. From the histograms of adjusted images when the level of image contrast is increased, intensities of most pixels are shifted toward darker or brighter tones. The histogram bins in low-key and high-key areas would receive more samples due to the tone shift. Such redistribution creates more observable peaks at some bins in low-key or high-key areas as shown in Figure 2.5 (f). These unique histogram patterns are therefore used in [60] to detect contrast enhancement.

An improved version of the algorithm is proposed in [61] by Stamm and Liu. They explain that the spike patterns can be detected by observing the Fourier transform of image histograms. Since the histograms of unadjusted images are typically smooth, the energy of the histogram will concentrate at the low frequency range. The histograms of enhanced images, on the other hand, have a spike patterns. The patterns cause the energy of the histogram to concentrate in the high frequency regions.

Median filtering is an efficient tool to smooth signals while edge details are preserved. For this reason, median filters are often used for image tampering. In [62], Bovik observes that median filtering leaves streaking artefacts in images. The artefacts can be described by the fact that pixels in adjacent rows or columns share the same value. Kirchner and Fridrich [63] present a method to detect images that are altered by median filtering using streaking artefacts. The detection is based on the ratio of the number of difference that equals to zero to the number of difference that equals to one. Using these criteria, the filtered images are likely to have the value of this ratio greater than one. Similarly in [64] the detection of median filtered images is based on the probability of zero values on the first order difference map in textured

regions. These methods are highly efficient in detecting streaking artefacts in uncompressed images. Pevný *et al.* [65] create an alternative approach to the method of Kirchner and Fridrich [63] to detect streaking artefacts in the images that have been compressed with JPEG. The approach is based on subtractive pixel adjacency matrix (SPAM) features. Further methods to detect traces from median filtering can be found in [66, 67].

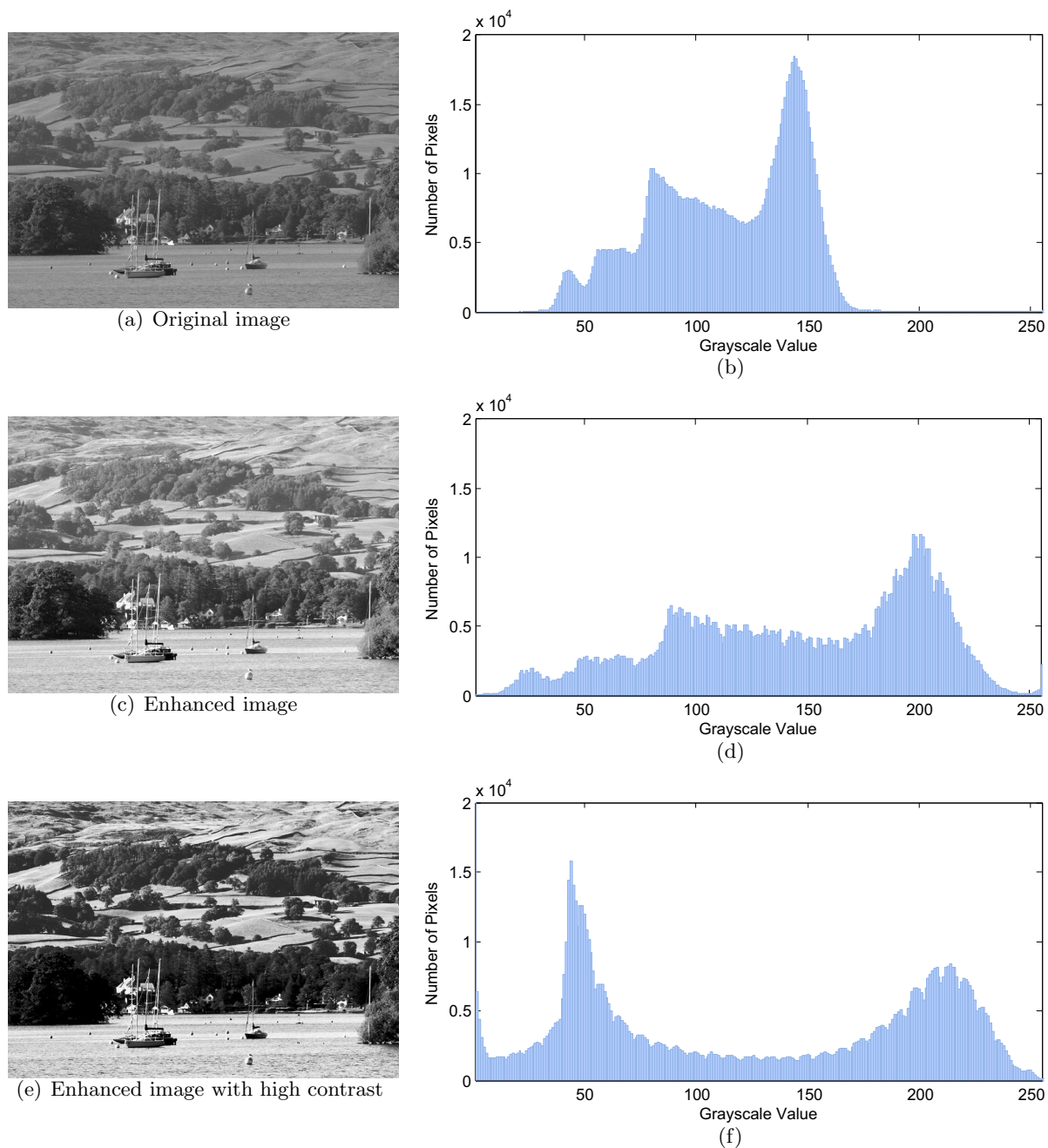


Figure 2.5: Examples of (a) an original image and its enhanced versions (c) and (e) that have gone through contrast and brightness adjustment. The corresponding histograms of image intensity are shown in (b), (d), and (f) respectively.

2.1.2 Footprints left on Recaptured Images

In this section we provide an overview of some of the more common features found in images that have been recaptured from LCD monitors. Some of these footprints were previously used for recapture detection in the literature.

Aliasing

Aliasing is sometimes introduced in digital camera images when the scene is insufficiently band-limited or contains detail with very high spatial frequencies [68]. It can be introduced to images at different stages of image recapture. Aliasing can occur in an initial photograph when capturing a scene with high detail using a camera with low pixel resolution. In particular, spatial aliasing might be visible when the spatial resolution of the image sensor is lower than the spatial Nyquist frequency of the input image. Aliasing can also occur during a reconstruction process. It is created because an output device, the monitor or printer, doesn't have sufficiently high resolution to represent a smooth line, or detail in the image. This type of aliasing is often referred to as jaggies.

Aliasing patterns are introduced during a recapture process because images are obtained from capturing photographs displayed on an LCD monitor which has a fine pattern of the pixel grid structure. This pattern is typically periodic and has high spatial frequency. In cameras that are equipped with a Colour Filter Array [69] the colour channels are normally sampled at frequencies that are lower than the native frequency of the image sensor. The green channel of a Bayer CFA can be described by a quincunx lattice arrangement and has a frequency response equivalent to the native 'unfiltered' sensor only in the horizontal and vertical directions. The red and blue channels are sampled on a rectangular lattice at one half the frequency of the native sensor. The diagram in Fig 2.6 shows the Nyquist boundaries and replication points for the red (R), green (G) and blue (B) colour channels of the Bayer CFA.

Most camera manufacturers fit optical anti-aliasing filters [71] to band-limit the high frequency components in the scene and prevent aliasing. However, the cut-off frequency of the filter is normally set above the Nyquist frequency to preserve the camera response at frequencies in the range 30-80% of the Nyquist frequency. The recapture of an image displayed on the screen of an LCD monitor is, therefore, highly likely to introduce aliasing due to the high frequency periodic pattern of the monitor pixel grid structure. Indeed, casually recaptured still images or videos of LCDs are often characterised by the presence of aliasing artefacts, also referred to as colour moiré, over the visible region of the display. These artefacts are very

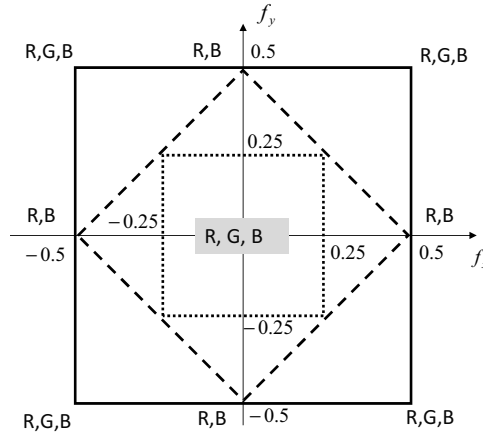
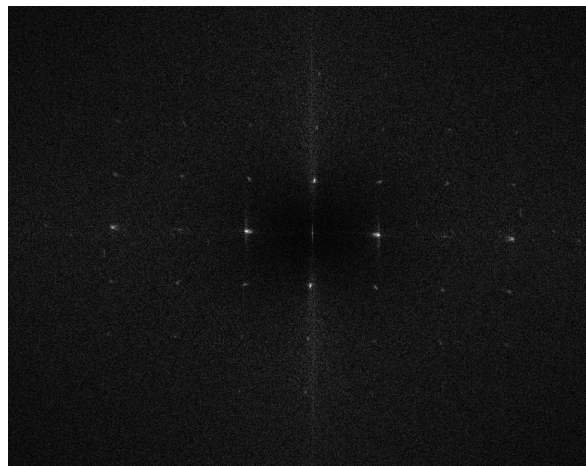


Figure 2.6: The Nyquist boundary for the unfiltered sensor (solid), green (dashed) and red/blue channels (dotted) of the Bayer CFA. All frequencies are in cycles/pixel [70].

difficult to eliminate through post-processing. Therefore, aliasing can be used as a feature for detecting recaptures. When aliasing artefacts are present in the recaptured image, the 2D DFT of the noise residual is likely to exhibit peaks in the 2D spectrum. In Figure 2.7(a) an image recaptured from an LCD monitor with noticeable aliasing is shown. The 2D DFT of its noise residual, shown in Figure 2.7(b), clearly shows the peaks in the frequency domain associated with the alias pattern. Detection of these peaks allows the identification of recaptured images [72, 70]. These aliasing artefacts, nevertheless, can be removed when the recapture is setup properly. In Section 5.1 we discuss, in practice, how to prepare a database of alias-free recaptured images.



(a) Recaptured image



(b) 2D FFT of the noise residual

Figure 2.7: An image recaptured from an LCD monitor with aliasing patterns. Peaks in the 2D FFT response of the recaptured image noise residual are clearly visible.

Noise

The two main sources of noise associated with images captured with a digital camera at normal and high levels of scene illumination are temporal noise, comprising mainly of shot noise, and fixed pattern noise which is dominated by Photo Response Non-Uniformity (PRNU) noise. The distribution of image noise in the recaptured image will be predominantly influenced by the noise characteristics of the recapture camera, the brightness setting of the LCD monitor, the capture distance and the scene content. The noise characteristics of the camera used to capture the original scene are likely to be also present in the recaptured image, but they will be band-limited due to the blurring effect introduced by the recapture process. The unique PRNU fingerprint of a camera's image sensor has been shown to be a highly successful tool for identifying the source camera from an image or a set of images [73]. The method has been applied successfully to detect the presence of the PRNU pattern in a scan of a printed image [74]. However, very small levels of rotation of the print are enough to significantly reduce the detection performance, since misalignment is introduced between the PRNU pattern and the recapture device (the scanner). This limits the applicability of their approach to our application since successful identification of the original capture device would require very low levels of misalignment between the LCD monitor pixel grid and the camera's image sensor, which in practice, would be very difficult to achieve. Thus, a recapture detection method using a noise feature is not in the scope of this thesis.

Contrast, Colour and Illumination Non-Uniformity

Almost all digital cameras and LCD monitor devices today, support the sRGB colour encoding specification [75]. In addition to specifying the gamut of colours that can be represented, the sRGB specification also describes forward and reverse non-linear tone transformation curves. In an ideal image capture display environment, the overall system tone response between input scene intensities and output display intensities at the monitor is linear. In practice, digital cameras apply a tone response function that deviates slightly from the ideal sRGB response, but is intended to provide a 'more pleasing' image that is slightly higher in contrast. In a recapture image chain where the response function of both the original and recapture cameras deviates from the sRGB specification as described above, the overall recaptured image is likely to appear higher in contrast relative to the single captured image. There may be some noticeable loss of detail and clipping of pixel values in the light and dark regions in the recaptured scene when compared with the original capture. For image contrast to be

used as a feature for recapture detection, a reliable, scene content independent method for the recovery of the global scene contrast or tone response function is required.

Colour related artefacts that may be present in recaptured images include colour balance errors, such as unwanted tints affecting the whole image, and increased colour saturation. Colour balance errors in a recaptured image can be minimised by calibrating the display monitor and by presetting the white point of the recapture camera to the LCD monitor white point before recapture. Thus, colour balance errors present in the recaptured image will have likely been introduced by the original camera that was used to capture the scene and not during the recapture process. The increase in colour saturation present in the recaptured image is likely to be due to the increase in overall image contrast as described above.

The transition to LED backlighting from cold cathode fluorescent (CCFL) backlight in LCD monitors has resulted in improved colour gamut, dynamic range and display non-uniformity. However, as monitor display sizes have increased, obtaining even backlight illumination remains a challenge for some low cost display devices. A luminance gradient may be noticeable in recaptured images containing large regions that are low in texture or detail. Identification of the luminance gradient would enable recaptured images to be detected.

Blurriness

Naturally occurring scenes contain a wide range of edges that vary in contrast and sharpness. When a scene is acquired with a digital camera, a certain level of blur, or distortion, is introduced into the image by the acquisition device. This occurs despite the fact that the image was correctly focussed by the camera at the time of capture. Imperfections in the lens, such as spherical aberration can introduce blur, as can diffraction. The latter is introduced when the diameter of the lens aperture is very small (due to a small aperture setting). Additional distortion may be introduced by processing carried out internally in the camera, such as sharpening, contrast enhancement or CFA demosaicing. The blur characteristics may, to a large extent, be considered unique to the camera at the time of acquisition. Given the importance of this footprint, we discuss how to model blurriness introduced by the acquisition processes in much more detail in Section 2.2.

Additional amount of blurriness to a digital image can also be introduced through the chain of image recapture. The increase of blurriness and blurring patterns introduced during image recapture might be useful traces for image recapture detection. In this thesis we are particularly interested in a blurriness feature and show that how we use this feature to solve

our problem in Chapter 3 and Chapter 4. We also show the Chapter 5 that our recapture detection method based on blurriness features has competitive performance when compared to existing methods using other features discussed above.

2.1.3 Related Work in Image Recapture Detection

The problem of detecting recaptured images is a relatively new area of research in image forensics. We have discussed in Section 1.1 why image recapture detection algorithms are important for image authentication and securities. In this section we provide a survey on approaches how to detect recaptured images. The method to detect images scanned from printed materials, such as photographs or magazines using specularities is presented by Yu *et al.* in [76]. Yin and Fang [77] propose an alternative approach based on dithering patterns left by printers on printed images to solve the same problem. In [74], Goljan *et al.* claim that PRNU patterns of the first acquisition device can be detected in recaptured images and an algorithm for camera model identification based on PRNU left on scanned images is proposed.

Specific features in images including the non-linearity of the tone response curve, the spatial distribution of the specularities in the image, image contrast, colour, chromaticity and sharpness are used in [78, 79] to create an algorithm to detect images recaptured from prints and from LCD monitors. Jiang *et al.* [80] develop an algorithm to identify printers from printed-and-scanned images based on Benford's law. The first digits are obtained from the most significant digits of decimal floating-point DCT coefficients computed from image blocks with size 8×8 pixels. The feature vectors are built from the first digit probability distribution of multi-size block DCT coefficients of images. The features are used to train an SVM model for printer classification.

Printed materials are not the only source of recaptured images. Images can also be reacquired by capturing images from LCD displays. In [1], Cao and Kot propose an algorithm for recapture detection based on three different features. The features based on fine texture patterns introduced during image recapture are used. The patterns are detected by computing Local Binary Pattern (LBP) features at multiple scales. The standard deviation of wavelet coefficients from multiple subbands are used as the second feature in order to detect the loss of image details. The detection of detail loss is important because it may be due to the relatively low display resolution used for recapture when compared to the camera's image sensor. The third feature used is the apparent increase in saturation in colours of the recaptured image. All three features are used to train an SVM classifier for recapture detection. The paper [81]

later suggests that the LBPV feature is more efficient than the conventional LBP feature, and shows that the detection precision can be improved using the LBPV.

Yin and Fang [82] propose two different features based on the noise and the traces of double JPEG compression. To estimate the noise features the image is denoised using three different discrete wavelet transforms. Statistical features such as mean, variance, skewness and kurtosis were computed from the histogram of the extracted noise residual. The traces of double JPEG compression is detected using the MBFDF algorithm [83]. These two features are used to create a recapture detection algorithm. Ke *et al.* [84] train a SVM classifier to detect recaptured images from LCD monitors using a 136 dimension feature set. The features are based on the descriptions of blurriness, texture, noise and colour features. Their method gives a detection rate of 97.2% when tested with a dataset of recaptured images taken with smart-phone cameras. The images used in their dataset [85] are low in resolution and quality, however, due to the smart-phone cameras used to perform the recapture.

2.2 Modelling of Blurriness Footprints

Digital images are often not perfectly sharp. A number of factors can reduce the sharpness quality of images. Poor capture conditions including camera shake, subject movement, and off-focus capture could result in blurred photographs. Some blurring patterns in fact are introduced by the acquisition process. Figure 2.8(a) exemplifies how the acquisition process can blur a sharp edge of an image. First an image of a given observed view is projected to an image sensor through a series of lens and optical filters by a focusing mechanism of the camera. Despite an in-focus condition, blurring distortions are often introduced to the image by the optical devices and an image sensor due to the imperfect properties of the devices. The amount of blur increases as the increase in number of pieces of lens and the drop of the quality of the materials. The image is then discretised to a digital format by an image sensor and blurring distortion can also be added to the image during this process.

The blurring patterns are determined by intrinsic properties of acquisition devices. When capture images of the sharp edge using different devices, edges present in the resulting images are often different in shape. The different patterns of edges can be represented in a one dimensional (1D) form of signals using edge profiles. Edge profiles are obtained through measuring the intensity of a single line that spans across edges. The acquisition in Figure 2.8(a) can be modelled using the 1D system shown in Figure 2.8(b). The sharp edge input $x(t)$ now is approximated using a step function. When it is acquired by the acquisition device, the input

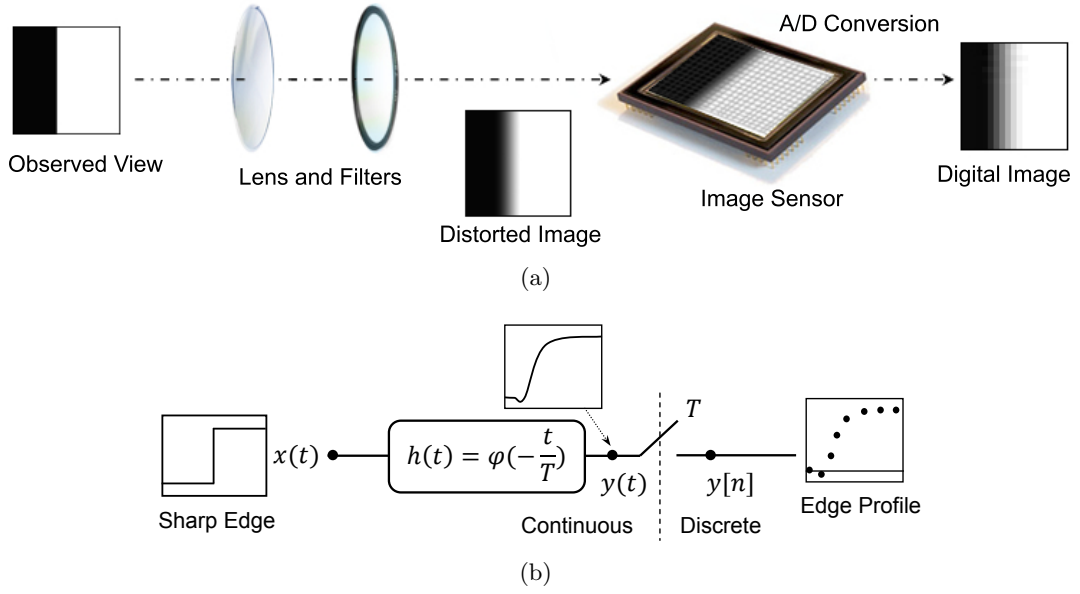


Figure 2.8: Simplified models of (a) image acquisition which describes how a sharp edge becomes blurred through an acquisition process and (b) corresponding signal diagram.

becomes blurred before being uniformly sampled to a discretised format with a sampling period T . The blurriness introduced to the input are described by the distortion caused by the pre-filter $h(t)$. The filter is determined by the imperfections of the device. The edge profile is obtained from the following measurement:

$$\begin{aligned}
 y[n] &= (x(t) * h(t))|_{t=nT} \\
 &= \int_{-\infty}^{\infty} x(t) \varphi(t/T - n) dt \\
 &= \langle x(t), \varphi(t/T - n) \rangle,
 \end{aligned} \tag{2.1}$$

where the sampling kernel $\varphi(t)$ is the time reversed and scaled version of the filter's impulse response $h(t)$ and n is a pixel indices of edge profile. The constant T is a sampling period.

In a two-dimensional imaging system, the sampling kernel φ in general can be modelled using the point spread function (PSF) of the device. The PSF describes response of an image acquisition device to a point source for in-focus condition. It determines the blurring pattern observed from the image of the point source. The PSF is typically characterised by intrinsic properties of the devices. An example of a point spread function is shown in Figure 2.9.

In practice, nevertheless, it is very difficult to measure the PSF of the device since it is difficult to create an infinitesimal point source. The alternative is to measure its line spread function (LSF). Line spread functions measure a spread response of an imaging system to a single line signal. By definition, a line spread function is a 1-D function corresponding to the

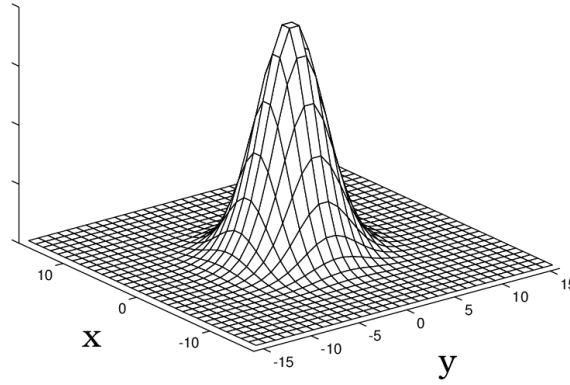


Figure 2.9: Example of a point spread function, excerpted from [86].

first derivative of the edge spread function (ESF) [87]. However, with the assumption that the PSF of a device is circularly symmetric and space invariant, the corresponding LSF can be derived from its PSF by integrating over crosssections of the PSF along points that lie on a straight line. The LSF is therefore a 1-D representation of corresponding two-dimensional PSF.

The function $\varphi(t)$ in the 1-D acquisition model in Figure 2.8(b) can be approximated using the LSF of the device. One way to measure the LSFs experimentally is to follow the slanted edge testing standard described in the international standard, ISO 12233 [88]. Examples of line spread functions of Canon EOS 600D and Nikon D40 models are shown in Figure 2.10(a) and Figure 2.10(b) respectively. The measurements were obtained using SFRMAT 3.0 software library [89].

Equation (2.1) suggests that the degrees and patterns of blurriness on edge profile $y[n]$ are dictated by the blurring kernel $\varphi(t)$ and the shape of the input edge $x(t)$. When a straight

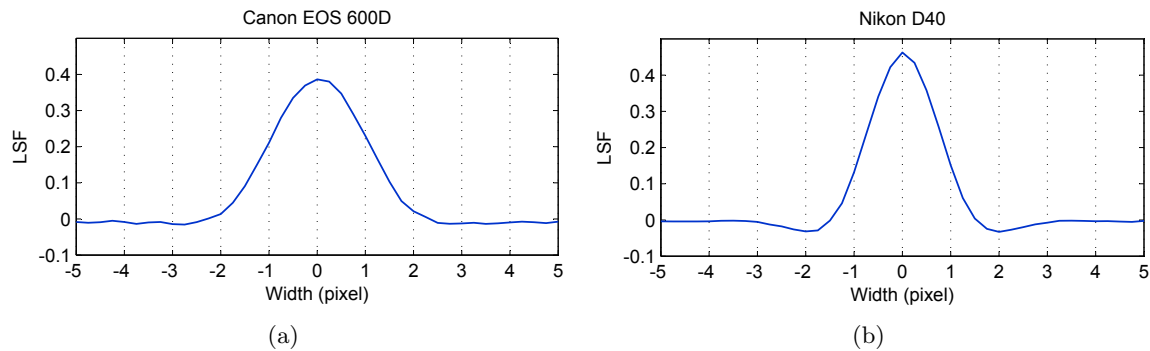


Figure 2.10: Examples of line spread functions measured from (a) Canon EOS 600D with Canon EF-S 18-135mm IS lens and (b) Nikon D40 with 18-55mm kit lens. While performing a slanted edge testing, the focal lengths of the cameras were fixed at 135mm and 55mm respectively.

sharp edge is acquired by different devices, different patterns of the resulting edge profiles are largely determined by the kernels $\varphi(t)$ of the devices. The edge profiles are therefore unique footprints left by the device on the image. In this research we use this relation between devices and edge profiles to trace back the acquisition history of images.

2.2.1 Related Work in Blurriness Measurement

Since we are particularly interested in modelling the blurriness introduced into images, it is worth reviewing the major techniques proposed to model and measure blurriness. A classical model for image degradation due to blurriness is presented in [90, 91, 92, 93]. The model describes a blurred image as the result of the convolution of sharp input image and the response of an imaging system which can be modelled using the point spread function (PSF). In [92], an extensive review for blind deconvolution techniques to decouple the blurriness and restore image quality is provided. A summary of blur identification techniques for image restoration is discussed in [94]. Marichal *et al.* [93] suggest that images with sharp edges would contain high spatial frequency components. The degree of blurriness is determined by the loss of DCT components in a high frequency band. In [95], different types of edge structures are defined according to their representations using Haar wavelet transform. A class of blurred edges corresponds to a group of edges with wavelet coefficients with low amplitude in high frequency wavelet subbands.

A number of approaches quantify the degrees of sharpness based on blurriness of edges. Dijk *et al.* propose a sharpness measurement method based on the ratio of width/amplitudes of lines and edges. A technique to measure global blurriness of images using a no-reference perceptual blur metric was pioneered by Marziliano *et al.* [96]. The method is based on the measurement of the degree of smoothing effect on edges. The blurriness metric is calculated from the average value of edge width in an image. Edge width is defined as the length from the beginning to the end of a local edge.

2.2.1.1 JNB and CPBD Blurring Metrics

The blurriness metric based on just noticeable blur (JNB) is introduced in [97] by Ferzli and Karam. The method is developed based on the sensitivity of human blur perception at different contrasts. The level of just noticeable blur (JNB) can be defined as the minimum amount of perceived blurriness around an edge given a contrast higher than the just noticeable difference (JND). The authors of the paper conducted experiments in order to obtain the JNB edge

width, that is, the width at which the subjects start to notice blurriness. The function of the probability of detecting blur at each edge in an image is then created using the derived JNB edge width w_{JNB} and can be represented in the form of a psychometric function as follows:

$$P_{blur}(e_i) = 1 - \exp\left(-\left|\frac{w(e_i)}{w_{JNB}(e_i)}\right|^\beta\right), \quad (2.2)$$

where $w(e_i)$ is the measured width of the edge e_i , $w_{JNB}(e_i)$ is the JNB width, which depends on local contrast C in the neighbourhood of the edge e_i , and β is a parameter whose value is obtained by means of least squares fitting.

From the experiment in [97], the JNB width w_{JNB} at various contrasts can be modeled as follows:

$$w_{JNB} = \begin{cases} 3 & \text{if } C \leq 50 \\ 5 & \text{if } C \geq 51 \end{cases}$$

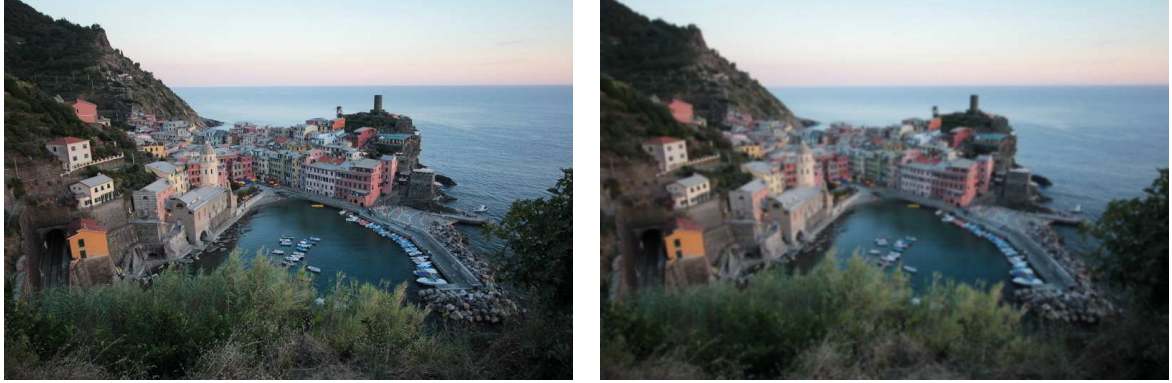
In Equation (2.2), when $w(e_i) = w_{JNB}(e_i)$, the probability of blur detection $P_{blur} = P_{JNB} = 63\%$. The probability obtained from (2.2), however, provides the probability of blur detection for a single edge. The JNB blurring metric is finally obtained from computing the probability of blur detection for all edges present in a given image using the probability summation model [98].

In the paper presented by Narvekar and Karam [99], the JNB metric is improved by evaluating the cumulative probability of blur detection (CPBD). Given a probability density function of blur detection, the CPBD metric can be computed as follows:

$$CPBD = P(P_{blur} \leq P_{JNB}) \quad (2.3)$$

From Equation (2.3), the CPBD metric represents the cumulative probability that blurred edges are not detected in a given image. The metric is therefore the measurement of image sharpness and negatively correlated to the level of perceptual blurriness. The greater value obtained from CPBD metric, the lower degree of blurriness is perceived. Figure 2.11(a) shows an example of an image with sharp edges with a corresponding CPBD at 0.57. The blurred version of the image is shown in Figure 2.11(b) and has a lower CPBD value at 0.13.

The JNB and CPBD metrics provide quantified levels of blurriness of images in a global sense and are primarily used for quality assessment. In our view, nevertheless, they can be used to classify groups of devices based on the quality of images they produced. For this reason, we discuss how to use the CPBD metric [99] as one of our features for the proposed image recapture detection algorithm of Chapter 5. The comparison of performance of the algorithm using CPBD metric and our proposed feature is also presented.



(a) Original image with CPBD=0.57

(b) Blurred image with CPBD=0.13

Figure 2.11: Two images with different levels of blurriness. (a) original image with sharp edges and corresponding CPBD at 0.57 (b) the blurred version of (a) with CPBD at 0.13.

2.3 Summary

This chapter has provided a literature survey of image forensic and recapture detection techniques. The approaches have been presented according to the types of footprints left on images. These footprints have been used to retrieve sources of images and detect image forgeries. We have also discussed about the footprints often found in recapture images. The footprints we have referred to, however, are not reliable and can be removed through a proper recapture setting. As a result in this thesis we have assumed that the only useful footprints are blurriness patterns left by acquisition devices. We later have presented the mathematical model of the blur introduced by cameras during the acquisition process. Finally a review on blurriness metrics and relevant methods to measure blurriness levels in images has been given.

Chapter 3

Reverse Engineering of Signal Acquisition Chains using the Theory of Sampling Signals with FRI

3.1 Introduction

In this chapter we propose a theoretical framework to address the problem of reverse engineering a chain of processing operators highlighted in Section 1.2. We model the chains of image acquisition and reproduction in Figure 1.3 by focusing only on multiple A/D and D/A operators. Moreover, the chains of A/D and D/A conversions are described using generalised sampling theory [100, 101] and we focus only on the case where the digital signal is recaptured at most once. Our framework is based on the fact that the features we are particularly interested in, for example edges, can be modelled using the theory of signals with finite rate of innovation (FRI). In particular, 1-D version of straight edge, a feature that is abundant in natural images, can be modelled using the step function.

An understanding of when the processing footprint left by an operator is completely removed by other operators or when a processing chain becomes too complex to be completely retrieved is, generally, of interest. It is anticipated that some operators may accidentally completely remove some footprints related to previous processing. For example, lossy compression can remove many fine features from a signal. It would, therefore, be nice to obtain an understanding of the circumstances under which a chain can always be precisely recovered and when two processing chains, overall, produce the same artefacts and, therefore, become indistinguishable.

An attempt at addressing this question is presented in this chapter. In Section 3.2 we set-up the problem. Theory of sampling signals with finite rate of innovation [102, 103] is used to provide some precise answers and an overview of this theory is provided in Section

3.3. Finally in Section 3.4 we discuss the conditions under which we can detect recapture and identify the parameters of the original A/D and D/A operators.

3.2 Problem Setup

Our research problem discussed in Section 1.2 is now modelled using a 1-D signal and the diagram presented in Figure 3.1. The diagram comprises a cascade of A/D and D/A conversions which are the most fundamental components of the acquisition chain. We model the input signal as a box function $f(t) = u(t - t_1) - u(t - t_2)$, where $u(t)$ is the unit step function, t_1 and t_2 are unknown delays of the unit step functions and $t_1 < t_2$. Initially, we consider $t_2 \rightarrow \infty$. Thus, the input $f(t)$ can be approximated by a step $u(t - t_s)$. The role of t_2 will be discussed in Section 3.4.4.

The sharp edge is first acquired by the first acquisition device with sampling kernel $\varphi_1(t)$ and a uniform sampling period T_1 . The discretised output $g[n]$ is given by $g[n] = \langle f(t), \varphi_1(t/T_1 - n) \rangle$ according to the classical sampling scheme described in Figure 2.8(b). The discrete samples $g[n]$ represent an observed edge profile in an original digital image. The characteristics of the profiles are determined by the sampling kernel $\varphi_1(t)$ and sampling period T .

After that, it is possible that the signal is reacquired. For example, in the case of images, the digital image might be displayed on an LCD screen before being recaptured. In that context the screen plays the role of the D/A converter. We assume this conversion is performed using the linear filter $\lambda(t)$. Therefore given the samples $g[n]$ the reconstructed signal is given by:

$$\hat{f}(t) = \sum_{n \in \mathbb{Z}} g[n] \lambda(t/T_2 - n). \quad (3.1)$$

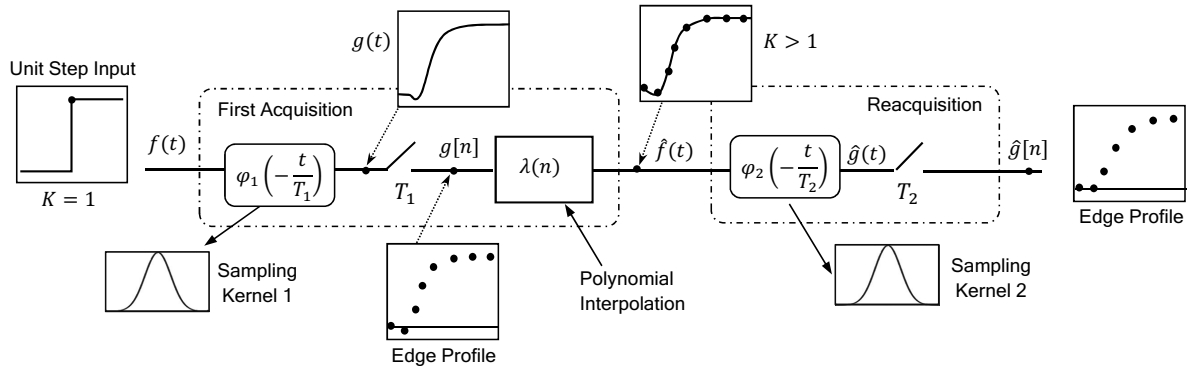


Figure 3.1: Problem statement diagram for signals with FRI in the chain of signal acquisition.

We assume that $\lambda(t)$ is a polynomial spline or a Maximum Order and Minimum Support (MOMS) function [104] of order R , and therefore $\hat{f}(t)$ is a piecewise polynomial function of maximum order R . We also assume $\hat{f}(t) \neq f(t)$.

The signal then goes through a second acquisition process where the acquisition device is modelled using $\varphi_2(t)$ and a uniform sampling period T_2 . As for the previous acquisition the acquired edge profiles $\hat{g}[n]$ is given by $\hat{g}[n] = \langle \hat{f}(t), \varphi_2(t/T - n) \rangle$.

Given a query image with an extracted edge profile $q[n]$, we would like to reverse engineer the acquisition history of the image. We assume that the known sampling kernel $\varphi_2(t)$ has the special properties that it can reproduce polynomials or exponentials as introduced in [103]. The key questions are as follows:

- (a) What stages in the chain are the samples $q[n]$ from? That is, was $q[n]$ obtained by acquiring $f(t)$ directly with $\varphi_2(t)$ or was $q[n]$ the reacquired signal $\hat{g}[n]$ as in Figure 3.1?
- (b) In the case of reacquisition, how can we retrieve the following important parameters: i) the maximum order R of polynomial used for interpolation ii) the sampling period T_1 , and iii) the sampling kernel $\varphi_1(t)$?
- (c) Under what conditions on φ_2 and T_2 , can we solve (b)?

All assumptions in this problem statement are made in order to simplify the problem and allow us to prove that we are able to solve the problem using our tools. We do not include the role of noise in this problem at this stage. The assumption that $\hat{f}(t) \neq f(t)$ is generally valid in practice since it is very difficult to achieve perfect reconstruction of signals. It is not necessary true, however, in practice that the interpolation function $\lambda(t)$ is a polynomial spline or a MOMS function. The properties of the interpolation function are typically determined by the properties of the D/A conversion device.

In the following section we provide an overview of the theory of sampling signals with FRI, and in Section 3.4 we describe how we apply this theory to develop our method to answer the above questions.

3.3 The Theory of Sampling Signals with FRI

To navigate through this chapter easily, Table 3.1 provides the most frequent notations that are used for our analysis.

Table 3.1: Frequently used notations

Symbols	Meaning
$x(t)$	Input signal
t_k, a_k, K	Innovation parameters of the train of Diracs, number of Diracs
$\varphi(t)$	Antialiasing filter prior to sampling (typically a polynomial reproducing kernel)
$y[n], N, T$	Samples, number of samples, sampling period
$\tau_p, P + 1$	Moments, number of moments (normally order of the kernel $N \geq P + 1 \geq 2K$)
h_m	Annihilating filter

3.3.1 Introduction to Signals with Finite Rate of Innovation

The notion of signals with finite rate of innovation and the corresponding sampling scheme were first introduced by Vetterli *et al.* in 2002 [102]. The theory describes how to sample and perfectly reconstruct special classes of non-band limited signals. According to the theory, FRI signals can be sampled at the sampling rate which are determined by how sparse the signals are per unit of time. Such sparsity can be described by the rate of innovation of the signals.

Given a signal $x(t)$ expressed by a set of known functions $\{f_r(t)\}_{r=0}^{R-1}$, arbitrary shift t_k , and amplitudes $\alpha_{k,r}$ as:

$$x(t) = \sum_{k \in \mathbb{Z}} \sum_{r=0}^{R-1} \alpha_{k,r} f_r(t - t_k), \quad (3.2)$$

the only free parameters of this signal are amplitudes $\alpha_{k,r}$ and shifts t_k since the functions $\{f_r(t)\}_{r=0}^{R-1}$, are known. The rate of innovation of the signal $x(t)$ is defined by the following:

$$\rho = \lim_{\tau \rightarrow \infty} \frac{1}{\tau} C_x\left(-\frac{\tau}{2}, \frac{\tau}{2}\right), \quad (3.3)$$

where $C_x(t_a, t_b)$ is a counting function which count the number of free parameters of the function $x(t)$ over the interval $[t_a, t_b]$.

A signal with finite rate of innovation (FRI) is, therefore, defined as a signal which can be represented as in Equation (3.2) and has a finite ρ as presented in Equation (3.3).

In addition, it is useful to consider a local rate of innovation over a window of size τ . Thus, we can write the rate of innovation at time t in terms of $\rho_\tau(t) = \frac{1}{\tau} C_x(t - \frac{\tau}{2}, t + \frac{\tau}{2})$, which converges to ρ as window size τ expands to infinity. The definition of local rate innovation is important when dealing with the local reconstruction of FRI signals.

3.3.2 Examples of FRI Signals

In this section we illustrate different examples of FRI signals which are useful for our forthcoming analysing. We are particularly interested in specific classes of signals with FRI which are stream of Diracs, streams of differentiated Diracs and piecewise polynomials (see Figure 3.2).

A stream of K Diracs can be written as follows:

$$x(t) = \sum_{k=0}^{K-1} a_k \delta(t - t_k). \quad (3.4)$$

From the equation we realise that the FRI signal $x(t)$ has $2K$ degrees of freedom and is completely specified by the knowledge of the K amplitudes a_k and the K locations t_k . Now we recall the definition of derivative of Diracs, which is useful to the analysis of piecewise polynomial signals. The Dirac function is a distribution function whose r -th derivative satisfies $\int_{-\infty}^{\infty} f(t) \delta^{(r)}(t - t_0) dt = (-1)^r f^{(r)}(t_0)$, where $f(t)$ is r times continuously differentiable. A stream of K differentiated Diracs can be expressed in the following form:

$$x(t) = \sum_{k=0}^{K-1} \sum_{r=0}^{R_k-1} a_{k,r} \delta^{(r)}(t - t_k). \quad (3.5)$$

In this case, the signal $x(t)$ is the combination of different derivative of Diracs $\delta^{(r)}$ with maximum order R_k at the location t_k . The third type of FRI signals that we are considering is a piecewise polynomial signal. When the polynomial function is of maximum degree $R - 1 \geq 0$,

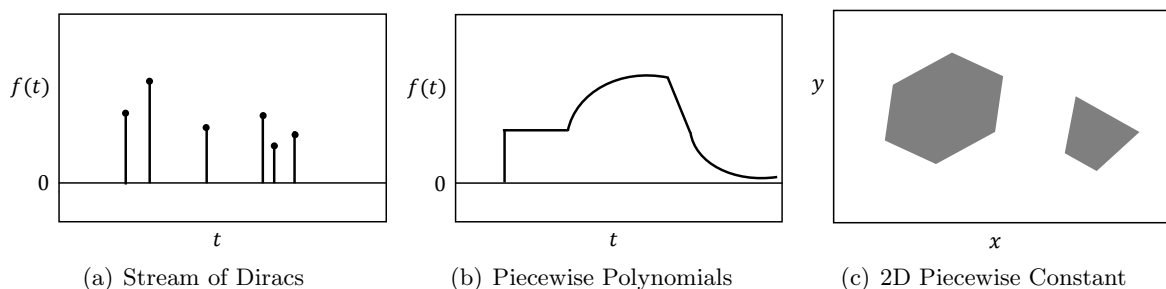


Figure 3.2: Examples of signals with finite rate of innovation (a) stream of Diracs (b) Piecewise Polynomials and (c) 2D Piecewise Constant

the FRI signal can be expressed as follows:

$$x(t) = \sum_{k=0}^{K-1} \sum_{r=0}^{R-1} a_{k,r} (t - t_k)^r. \quad (3.6)$$

We note that if we take the R -th order derivative of the above signal we obtain a stream of differentiated Diracs. The function can be described as follows:

$$x^{(R)}(t) = \sum_{k=0}^{K-1} \sum_{r=0}^{R-1} r! a_{k,r} \delta^{(R-r-1)}(t - t_k). \quad (3.7)$$

3.3.3 Sampling Signals with Finite Rate of Innovation

Consider the signal $x(t)$ given by:

$$x(t) = \sum_{k=0}^{K-1} a_k \delta(t - t_k). \quad (3.8)$$

We are interested in the situation where the input signal $x(t)$ is acquired through our standard acquisition model of Figure 2.8(b). The resulting N discrete samples $y[n]$ are given by Equation (2.1) as follows:

$$\begin{aligned} y[n] &= x(t) * h(t)|_{t=nT} \\ &= \int_{-\infty}^{\infty} x(t) \varphi(t/T - n) dt \\ &= \langle x(t), \varphi(t/T - n) \rangle. \end{aligned} \quad (3.9)$$

The input is distorted by the sampling kernel $\varphi(t)$ which we assume is a function that can reproduce polynomials of maximum order P as described in [103]. For polynomial reproducing kernels we mean functions $\varphi(t)$ satisfying

$$\sum_{n \in \mathbb{Z}} c_{n,p} \varphi(t/T - n) = t^p ; \quad p = 0, 1, 2, \dots, P, \quad (3.10)$$

for a proper choice of coefficients $c_{n,p}$.

The polynomial function is reproduced from the weighted sum of shifted version of the kernel $\varphi(t)$. The conditions a kernel has to satisfy in order to be able to reproduce polynomials are called Strang-Fix conditions and are discussed in [105]. The coefficients $c_{n,p}$ used to

reproduce the monomial t^p can be calculated as follows:

$$\begin{aligned} c_{n,p} &= \langle t^p, \tilde{\varphi}(t/T - n) \rangle \\ &= \frac{1}{T} \int_{-\infty}^{\infty} t^p \tilde{\varphi}(t/T - n) dt ; \quad p = 0, 1, 2, \dots, P, \end{aligned} \quad (3.11)$$

where $\tilde{\varphi}(t)$ is a dual of the $\varphi(t)$. Specifically $\tilde{\varphi}(t)$ is such that $\langle \tilde{\varphi}(t), \varphi(t - n) \rangle = \delta_n$.

Call $\tau_p = \sum_n c_{n,p} y[n]$; $p = 0, 1, 2, \dots, P$ the moments of the observed samples, where the coefficients $c_{n,p}$ are those in Equation (3.10) that reproduce t^p . One can derive the moments τ_p of the stream of Diracs $x(t)$ as follows:

$$\begin{aligned} \tau_p &= \sum_n c_{n,p} y[n] \\ &\stackrel{(a)}{=} \langle x(t), \sum_n c_{n,p} \varphi(t/T - n) \rangle \\ &\stackrel{(b)}{=} \langle \sum_{k=0}^{K-1} a_k \delta(t - t_k), \sum_n c_{n,p} \varphi(t/T - n) \rangle \\ &\stackrel{(c)}{=} \int_{-\infty}^{\infty} \sum_{k=0}^{K-1} a_k \delta(t - t_k) t^p dt \\ &= \sum_{k=0}^{K-1} a_k t_k^p ; \quad p = 0, 1, 2, \dots, P, \end{aligned} \quad (3.12)$$

where (a) follows from the linearity of the inner product, (b) from the input $x(t) = \sum_{k=0}^{K-1} a_k \delta(t - t_k)$, (c) from the properties of polynomial reproduction kernel in Equation (3.10). The result of the integration in (c) is the $m - th$ order moment of the original signal $x(t)$.

Once the moments τ_p ; $p = 0, 1, \dots, P$ and $P \geq 2K$ have been computed, the following Toeplitz matrix is constructed:

$$\mathbf{S} = \begin{bmatrix} \tau_K & \tau_{K-1} & \cdots & \tau_0 \\ \tau_{K+1} & \tau_K & \cdots & \tau_1 \\ \vdots & \vdots & \ddots & \vdots \\ \tau_P & \tau_{P-1} & \cdots & \tau_{P-K} \end{bmatrix}. \quad (3.13)$$

Note that, one can show that \mathbf{S} has always rank K where K is the number of Diracs in $x(t)$ and that $x(t)$ is determined from the knowledge of the null space of \mathbf{S} [103].

The estimation of the number of degrees of freedom of the input signal using this matrix \mathbf{S} is crucial for our forensic analysis. It allows us to retrieve the number of Diracs or discontinuities

K in the input signal $x(t)$ through the computation using discrete samples $y[n]$. We use this principle to develop our forensic technique in Section 3.4.

Now, nevertheless, it is worth to describe the method to reconstruct the signal $x(t)$ using the FRI sampling theory. The perfect reconstruction technique of FRI sampling theory is based on the annihilating filter method or, often referred to Prony's method [106]. The introduction of annihilating filter method for sampling theory of signals with finite rate of innovation was made by Vetterli *et al.* [102]. The goal of the algorithm is to retrieve a pair of unknown parameters (a_k, t_k) in the power sum series of K components as follows:

$$\tau_p = \sum_{k=0}^{K-1} a_k (t_k^p) ; p = 0, 1, 2, \dots, P, \quad (3.14)$$

The annihilating filter $H(z)$ is defined as a filter which has roots at the locations t_k

$$H(z) = \sum_{p=0}^K h_p z^{-p} = \prod_{k=0}^{K-1} (1 - t_k z^{-1}). \quad (3.15)$$

The filter $H(z)$ has a finite impulse response h_p of length $K + 1$. Then, one can derive that

$$h_p * \tau_p = \sum_{i=0}^K h_i \tau_{p-i} = \sum_{k=0}^{K-1} a_k t_k^p \underbrace{\sum_{i=0}^K h_i t_k^{-i}}_{=0} = 0. \quad (3.16)$$

It is proven that the annihilating filter $H(z)$ annihilates all the components of the series of powersum τ_p . Assuming $h_0 = 1$, Equation (3.16) shows that the coefficients of the filter $H(z)$ can be retrieved when any $2K$ consecutive values of τ_p are known by solving the following Toeplitz system:

$$\begin{bmatrix} \tau_{K-1} & \tau_{K-2} & \cdots & \tau_0 \\ \tau_K & \tau_{K-1} & \cdots & \tau_1 \\ \vdots & \vdots & \ddots & \vdots \\ \tau_{2N-1} & \tau_{2N-2} & \cdots & \tau_{K-1} \end{bmatrix} \cdot \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_K \end{bmatrix} = - \begin{bmatrix} \tau_K \\ \tau_{K+1} \\ \vdots \\ \tau_{2K-1} \end{bmatrix}. \quad (3.17)$$

After the filter coefficients h_i are obtained, the parameters t_k can be calculated by finding the roots of $H(z)$ as in Equation 3.15. With the results, one can solve the Equation (3.14) in order to find the parameters a_k using Vandermonde system:

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ t_0 & t_1 & \cdots & t_{K-1} \\ \vdots & \vdots & \ddots & \vdots \\ t_0^{K-1} & t_1^{K-1} & \cdots & t_{K-1}^{K-1} \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{K-1} \end{bmatrix} = \begin{bmatrix} \tau_0 \\ \tau_1 \\ \vdots \\ \tau_{K-1} \end{bmatrix}, \quad (3.18)$$

and all unknown parameters a_k, t_k are now determined. Finally we can perfectly reconstruct the original stream of Diracs $x(t)$ in Equation (3.8) using the retrieved parameters.

From the sampling setting next we consider an input signal which is a piecewise polynomial signal with K pieces of maximum degree $R \geq 0$, that is:

$$x(t) = \sum_{k=1}^K \sum_{r=0}^R a_{k,r} (t - t_k)^r. \quad (3.19)$$

Clearly the $(R + 1)$ order derivative $x^{(R+1)}(t) = \frac{d^{(R+1)}x(t)}{dt^{(R+1)}}$ is given by a train of differentiated Diracs at the locations t_k as follows:

$$x^{(R+1)}(t) = \sum_{k=0}^{K-1} \sum_{r=0}^R r! a_{k,r} \delta^{(R-r)}(t - t_k). \quad (3.20)$$

We observe that $x^{(R+1)}(t)$ is an FRI signal and the rate of innovation of the signal is determined by the total number of differentiated Diracs at the different locations t_k . We also note that the finite difference of the discrete samples $z^{(1)}[n]$ satisfies [103]:

$$\begin{aligned} z^{(1)}[n] &= y[n+1] - y[n] \\ &= \langle x(t), \varphi(t/T - n - 1) - \varphi(t/T - n) \rangle \\ &= \left\langle \frac{dx(t)}{dt}, \varphi(t/T - n) * \beta_0(t/T - n) \right\rangle, \end{aligned} \quad (3.21)$$

where β_0 is the box function. Therefore, the moments of the derivative of $x(t)$ are given by $\tau_p = \sum_n c_{n,p}^{(1)} z^{(1)}[n]$, where $c_{n,p}^{(1)}$ are the polynomial reproduction coefficients of Equation (3.10) for the new kernel $\varphi(t) * \beta_0(t)$. The moments of the $R + 1$ derivative of $x(t)$ can be obtained similarly. It is again possible to show that the Toeplitz matrix \mathbf{S} of the moments of $x^{(R+1)}(t)$ has rank equal to the number K of polynomial pieces in $x(t)$. Finally the locations of discontinuities t_k in the piecewise polynomial function can be determined in the similar way using annihilating filter method.

3.4 Reverse Engineering Technique using the FRI Theory

3.4.1 Chain Structure Identification and Recapture Detection

We are given a query digital signal $q[n]$ and we want to determine whether this is the result of acquiring the unit step function, $f(t) = u(t - t_s)$, with $\varphi_2(t)$ or whether this is the result of reacquisition. An illustrative example of the two possible shapes of $q[n]$ is shown in Figure 3.3. In Figure 3.3(b) we show the case of a single acquisition of $f(t)$ shown in Figure 3.3(a), whereas Figure 3.3(e) shows a reacquired signal obtained after linear interpolation of (b) to yield 3(d) and sampling of 3(d) with $\varphi_2(t)$. Please see the detail of the simulation in Section 3.4.3. We note that the shapes of $g[n]$ and $\hat{g}[n]$ might look similar yet they still contain all the information necessary to reverse engineer the acquisition chain as is shown next.

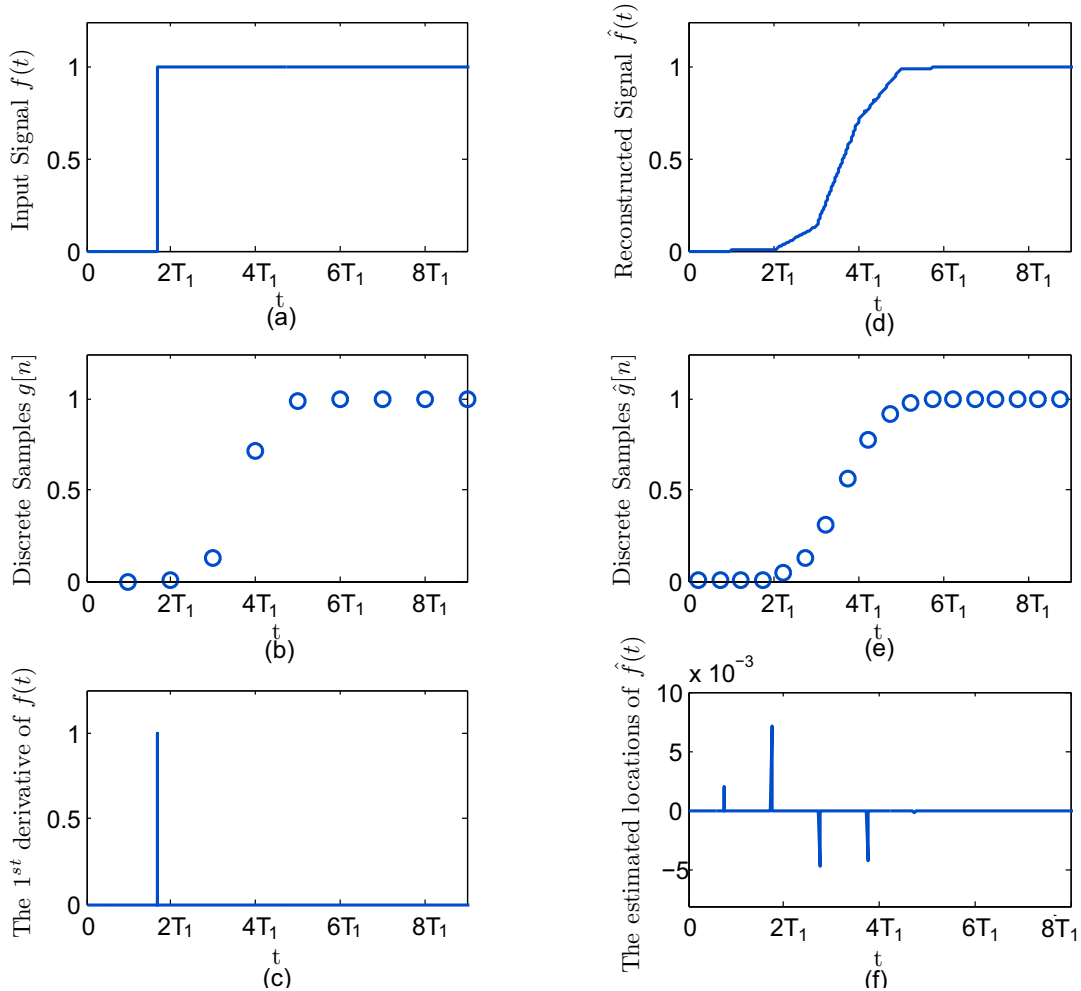


Figure 3.3: Comparative plots of discrete samples (b) $g[n]$ and (e) $\hat{g}[n]$ obtained from acquiring the continues inputs (a) step input $f(t)$ and (d) the reconstructed signal $\hat{f}(t)$ respectively. The corresponding reconstruct locations based on the samples (b) $g[n]$ and (e) $\hat{g}[n]$ using the FRI sampling theory are shown in subfigures (c) and (f) respectively.

The input signal $f(t)$ is a unit step function which is described by only one free parameter – the location of step t_s . When the input signal $f(t)$ is acquired, the observed samples $g[n]$ are distorted by the sampling kernel. All possible $g[n]$, however, are still determined by one free parameter. In contrast, $\hat{f}(t)$ is obtained by polynomial interpolation and is a polynomial function with discontinuities at multiple locations of period T_1 . The signal is a special case of FRI signals in (3.19).

We thereby use this principle to create an algorithm for reacquisition detection. Algorithm 1 in Table 3.2 presents our method to detect whether a query $q[n]$ was singly acquired or was reacquired. Since a step function is a piecewise polynomial of maximum degree $R = 0$, the moments are computed using a first order finite difference of the query as $\tau_p = \sum_n c_{n,p} q^{(1)}[n]$. The moments are then used to construct the Toeplitz matrix \mathbf{S} . The matrix \mathbf{S} of size 2×2 is sufficient for reacquisition detection since the matrix is always rank-deficient with rank = 1 if $q[n]$ was acquired from a step input. If, on the other hand, \mathbf{S} is full rank, it means $q[n]$ was obtained by reacquisition.

Table 3.2: **Algorithm 1** Reacquisition detection

Objective: Detect whether the unknown samples $q[n]$ are obtained from acquiring an original signal (the step function) or a reconstructed signal?

- 1) Compute the first order finite difference $q^{(1)}[n]$.
- 2) Compute the coefficients $c_{n,p}^{(1)}$, where $c_{n,p}^{(1)}$ are the polynomial reproduction coefficients of Equation (3.10) for the new kernel $\varphi(t) * \beta_0(t)$.
- 3) Calculate the sequence

$$\tau_p = \sum_n c_{n,p}^{(1)} q^{(1)}[n]; p = 0, 1.$$

- 4) Build the rectangular Toeplitz matrix \mathbf{S} (3.13) with size 2×2 using the measurements τ_p .
 - 5) Validate whether the matrix \mathbf{S} is full rank or rank-deficient? If \mathbf{S} is full rank, the samples $q[n]$ are from reacquisition. On the other hand, if \mathbf{S} is rank-deficient, $q[n]$ are from original acquisition.
-

3.4.2 The Retrieval of Chain Parameters

From the result of our analysis it is possible to retrieve the acquisition history of the given signal $q[n]$. When a query signal is detected as having been reacquired, the question arises of how important image chain parameters such as the sampling period T_1 and the interpolation function $\lambda(t)$ can be retrieved.

The Retrieval of Interpolation Function

The maximum order R of the polynomial interpolation function $\lambda(t)$ can be retrieved from the properties of FRI reconstructed signals. In practice the maximum degree R represents the ability of the device to reconstruct complicated curves corresponding to given samples. A monitor with a higher degree R , for an example, would be able to produce images with more complicated curve and edge structures compared to another monitor which can produce polynomials of lower maximum degree. According to Section 3.3.1, piecewise polynomial functions of maximum degree R are fully suppressed by differentiation by order $R + 1$. If we measure the number of degrees of freedom using Toeplitz matrix \mathbf{S} , the matrix will be full rank until a finite difference of order $r \geq R + 1$ is applied to query samples $q[n]$. When $r = R + 1$, the matrix will be rank deficient with rank, K , equal to the number of K pieces of a piecewise polynomial function. Algorithm 2 in Table 3.3 summarises our proposed method to estimate the parameter R , using iterative finite difference and rank measurements, for order R until \mathbf{S} is rank deficient.

Sampling Rate Retrieval

Once the matrix \mathbf{S} is rank-deficient and has rank K , we realise that the number of locations or discontinuities in the input $x(t)$ is finite. Recall that if $K > 1$ the query signal $q[n]$ is obtained from recapturing the reconstructed signal $\hat{f}(t)$. We can use the annihilating filter method as discussed in Section 3.3.3 to estimate the continuous function $\hat{f}(t)$ and all discontinuity locations t_k in the function. Each t_k represents the location of samples $g[n]$ used in the interpolation. From the retrieval results in Figure 3.3(f), the distances between the differentiated Diracs $\hat{f}^{(2)}(t)$ are uniform and the sampling period T_1 can be estimated from the average of the distances.

Table 3.3: **Algorithm 2** The retrieval of maximum order R used in interpolation

Objective: To estimate the maximum order R of polynomials used in interpolation

- 1) Run **Algorithm1** to detect whether the given samples $q[n]$ are obtained from acquiring an original signal (the step function) or a reconstructed signal?
 - 2) If $q[n]$ are from acquiring an original signal, stop the algorithm because $q[n]$ are obtained from the sharp edge with $R = 0$.
 - 3) Otherwise, if $q[n]$ are from reacquisition, begin an iterative algorithm to estimate R . Set the initial order of finite difference used $r = 2$. Repeat until \mathbf{S} is rank-deficient.
 - 4) Calculate the r^{th} order finite difference $q^{(r)}[n]$.
 - 5) Compute the coefficients $c_{n,p}^{(r)}$, where $c_{n,p}^{(r)}$ are the polynomial reproduction coefficients of Equation (3.10) for the new kernel $\varphi(t) * \beta_{r-1}(t)$.
 - 6) Calculate the sequence

$$\tau_p = \sum_n c_{n,p}^{(r)} q^{(r)}[n]; p = 0, 1, 2, \dots, P.$$
 - 7) Choose M such that M is sufficiently large than the expected value of K , and $M < P$. In the simulation we use $M < 10$. Build a new rectangular Toeplitz matrix \mathbf{S} (3.13) with size $M \times M$ using the measurements τ_p .
 - 8) Find the rank K of the matrix \mathbf{S} by computing Single Value Decomposition (SVD). Recall that, $\mathbf{S} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*$ and $\mathbf{\Sigma}$ is diagonal. The rank of the matrix \mathbf{S} is the number of non-zeros elements in the diagonal matrix $\mathbf{\Sigma}$.
 - 9) Validate whether the matrix \mathbf{S} is full rank or rank-deficient? The matrix is rank-deficient only if $K < M$. Otherwise, the matrix \mathbf{S} is full rank.
 - 10) If the matrix \mathbf{S} is rank-deficient, stop the algorithm and obtain $R = r - 1$. Otherwise, set $r = r + 1$ and go back to step 4.
-

Table 3.4: **Algorithm 3** The estimation of the sampling period T_1

Objective: Estimate the sampling period T_1 based on the retrieved locations t_k of discontinuities in the interpolated signal $\hat{f}(t)$.

- 1) Run **Algorithm 2** to estimate the parameters K , R , and the corresponding rank-deficient matrix \mathbf{S} . Stop the algorithm if the given $q[n]$ are from acquiring the original signal.
 - 2) The rank K obtained from \mathbf{S} is the number of discontinuities in the stream of Differentiated Diracs $\hat{f}^{(R)}(t)$.
 - 3) Apply the annihilating method to estimate the locations t_k of discontinuities.
 - 4) Estimate the sampling period T_1 from the retrieved t_k .
-

We note that, in practice, there might be numerical errors from the calculation and the matrix \mathbf{S} might not be always fully rank-deficient. In our algorithms, singular values $\sigma_{i,i}$ that are extremely small or smaller than $1/1000$ of $\sigma_{i-1,i-1}$ are neglect and set to zero, where $\sigma_{i,i}$ are singular values at the row and column (i, i) of the rectangular diagonal matrix $\mathbf{\Sigma}$.

3.4.3 Numerical Simulation

In this experiment we present a simulation in which the step signal $f(t) = u(t - t_s)$ has gone through the chain of acquisition and reproduction as shown in Figure 3.3. For the sake of simplicity we set $T_1 = 1$ with a signal resolution at $1/64$. Since our primary goal is to prove that in theory it is possible to reverse engineer the chain of acquisition using our method, no noise was added to the simulation.

The input step signal $f(t)$ in Figure 3.3 (a) was first acquired using the first acquisition device with the first sampling kernel $\varphi_1 = \beta_3(t)$, where $\beta_3(t)$ is the B-spline of third order. The signal was then uniformly sampled with a sampling period T_1 and we obtained the discrete sample $g[n]$ as shown in Figure 3.3 (b). Given $g[n]$, we were able to retrieve $K = 1$ using Algorithm 1 and could verify that the samples $g[n]$ were from the first acquisition.

The samples $g[n]$ were then reconstructed to a continuous signal using linear interpolation and the reconstructed signal $\hat{f}(t)$ was obtained. The signal $\hat{f}(t)$ was next reacquired using the second device with the B-spline of 8^{th} order before the samples $\hat{g}[n]$ were obtained. The sampling frequency used in the reacquisition was as twice as the sampling frequency of the first acquisition, or $T_2 = 1/2T_1$. The reconstructed edge $\hat{f}(t)$ was reacquired with a shift by $\Delta t = +\frac{5}{4}T_1$ thus $g[n]$ and $\hat{g}[n]$ were not in phase.

The samples $\hat{g}[n]$ are shown in Figure 3.3 (e). Given the samples $\hat{g}[n]$, we were able to estimate the maximum order used for interpolation $R = 1$ using Algorithm 2. The estimation result verifies that the interpolation technique used was linear interpolation. We next ran Algorithm 3 in order to estimate t_k and the sampling period T_1 . The results for location retrieval according to Figure 3.3 (f) are shown in Table 3.5. We note that the results obtained from the estimation of the locations t_k are shifted by $\frac{5}{4}T_1$ due to the shift during the reacquisition.

Table 3.5: The retrieved locations t_k of the discrete samples $g[n]$ estimated using the annihilating filter method

	Location t_1	Location t_2	Location t_3	Location t_4
Actual locations	0.75000	1.75000	2.75000	3.75000
Estimated Locations	0.74786	1.75219	2.75347	3.74833
Errors	0.00214	0.00219	0.00347	0.00167

Table 3.6: The estimated periods with the average

Periods	Estimated Periods
T_{12}	1.00504
T_{23}	1.00128
T_{34}	0.99486
Average	1.00157

We observe that the annihilating filter method provides good precision in estimating the locations t_k . The average error of the estimation is 0.00237 for $T = 1$. Then we use the results to estimate the sampling period T_1 . In Table 3.6 we obtain the estimation $\bar{T}_1 = 1.00157$ which is close to the actual value of $T_1 = 1$ previously set in the simulation.

3.4.4 Sufficient Conditions for Reverse Engineering

The estimation of the samples $g[n]$ can be obtained using the retrieved $\hat{f}(t)$ and T_1 through the reverse sampling. Finally, the retrieval of φ_1 can be further achieved using the best match between the samples and all possible dictionary elements as proposed in [107]. We omit this proof and instead focus on providing the sufficient conditions on φ_2 and T_2 that allow us to retrieve the chain.

Firstly, the maximum degree P of polynomial which the second kernel φ_2 can reproduce must be sufficiently large. From [103], the kernel must be able to reproduce polynomials of

maximum degree $P \geq 2(R+1)K - R - 2$ in order to achieve perfect reconstruction of a piecewise polynomial of maximum degree R with K discontinuities. In our case, the unit step input signal is sampled with uniform sampling period T_1 and the samples are then interpolated to the continuous domain again. The number of discontinuities can be computed as $K \leq \frac{L_1}{T_1} + 1$, where L_1 is the support of the first sampling kernel φ_1 . Therefore, the order P which provides the precisely retrieved results is given as follows:

$$P \geq 2(R+1)\frac{L_1}{T_1} + R. \quad (3.22)$$

Secondly, we consider the role of t_2 which is now a constant with $t_2 > t_1$. The input, a rectangular pulse $f(t) = u(t - t_2) - u(t - t_1)$, is then acquired and reproduced by the chain. Since signal reconstruction creates a new group of K piecewise polynomials from samples of a unit step input, one needs to ensure that the two groups of piecewise polynomials are sufficiently distant in order to avoid overlap. The minimum interval required is greater than $2KT_1$. From [103], a piecewise polynomial function with two groups of K pieces of maximum degree R can influence an interval of size $2K(L_2 + R + 1)T_2$. One therefore can calculate the bound:

$$T_1 > (L_2 + R + 1)T_2, \quad (3.23)$$

where L_2 is the support of $\varphi_2(t)$.

The derived sufficient conditions in Equation (3.22) and (3.23) impose the constraints that the second device must have better qualities when compared to the first acquisition device. The first condition suggests that the second sampling kernel must be able to reproduce polynomial of sufficiently high degree. This minimum degree P is determined by the length of the first sampling kernel L_1 , sampling period of the first acquisition T_1 , and the maximum order R of the polynomial interpolation. This condition ensures that proper linear combinations of the kernel and its shifted versions can reproduce polynomials such that the FRI signal $\hat{f}(t)$ is uniquely reconstructed from the samples $\hat{g}[n]$.

The second condition imposes a constraint on the maximum sampling period T_2 . That is, it is required to sample the signal in the second acquisition stage with sufficiently high sampling rate compared to the rate used in the first acquisition. In image case, we are required to recapture image using a camera with the sufficiently high resolution.

When sampling signals with these requirements, a one-to-one mapping between discrete samples and chain structures is guaranteed. While we cannot guarantee that the conditions provided are necessary for a one-to-one mapping, we can provide a counter example to show

that signals obtained from different acquisition chains may not be distinguished when the sufficient conditions are violated.

Let $q_a[n]$ and $q_b[n]$ be query discrete samples acquired from different chain structures. The signal $q_a[n]$ is obtained from a single acquisition of the pulse input $f_a(t) = u(t - 3T_2/2) - u(t - 4T_2)$ using a box spline kernel [108] and $T_{1a} = T_2$ or $\varphi_{1a}(t) = \frac{1}{T_2}\beta_0(\frac{t}{T_2})$. From Figure 3.4, one can compute $q_a[n] = \langle f_a(t), \beta_0(t/T_2 - n) \rangle = [0 \ 1/2 \ 1 \ 1]$.

The signal, $q_b[n]$, is from a reacquisition. Given $f_b(t) = u(t - 5T_2/4) - u(t - 4T_2)$ is the initial input, the signal is first sampled using $\varphi_{1b}(t) = \frac{2}{T_2}\beta_0(\frac{2t}{T_2})$ or $T_{1b} = T_2/2$. The resulting samples are given by $[0 \ 0 \ 1/2 \ 1 \ 1 \ 1 \ 1]$. Then the samples are reconstructed using linear interpolation and the reconstructed signal $\hat{f}_b(t)$ is obtained. The reconstructed signal is next reacquired using $\varphi_{2b}(t) = \varphi_{1a}(t) = \frac{1}{T_2}\beta_0(\frac{t}{T_2})$ as shown in Figure 3.4. The samples $q_b[n]$ is given by $q_b[n] = \langle \hat{f}_b(t), \beta_0(t/T_2 - n) \rangle = [0 \ 1/2 \ 1 \ 1]$. We have $q_a[n] = q_b[n]$ despite the fact that each set of samples has been obtained from different acquisition chain structures. The signals from different chains become indistinguishable because the kernels used can reproduce polynomials to degree $P = 0$ which violates the sufficient conditions. Thus, a one-to-one mapping is not guaranteed and the proposed algorithm cannot retrieve a unique chain solution.

In this chapter we have proposed a framework for reverse engineering of signal acquisition chains. We have solved the problem from a very theoretical point of view. We have shown that under some favorable conditions, in theory, it is possible to reverse engineer chains of reacquisition using our method based on the theory of signal with FRI. Our constructive method and the validation simulation are based on a noise-free condition in order to simplify the problem. Since we have limited time and resources, we consider that the development of the algorithms for practical applications with the presence of noise is beyond the scope of our theoretical framework.

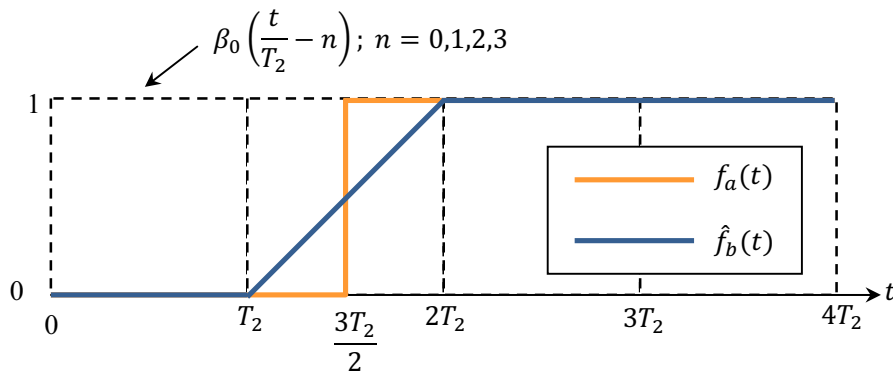


Figure 3.4: Counter examples when sampling $f_a(t)$ and $\hat{f}_b(t)$ with a sampling kernel $\varphi_a(t) = \varphi_b(t) = \beta_0(\frac{t}{T_2})$

Our propose method in practice has theoretical constraints when our assumptions are violated. In our research we assume that the sampling kernels are the kernels that can reproduce polynomials. The kernels, however, are unstable at higher orders and sensitive to noise. The matrix \mathbf{S} in Algorithm 1 would be always full-rank in practice and our method would fail to detect reacquisition in a noisy condition. The problem can be solved using the methods of Urigüen *et al.* [109, 110] based on exponential reproducing kernels and Cadzows denoising method [111]. The analysis of the impact of noise on the performances of the algorithms can be studied following the methods presented in [112, 113].

3.5 Summary

This chapter has presented our proposed reverse engineering technique based on the theory of sampling signals with finite rate of innovation. We first reviewed how to sample and perfectly reconstruct FRI signals. Next we have discussed how to apply the sampling theory of signals with FRI to retrieve signals in the previous stages of signal acquisition chains. Our technique is capable of detecting whether the signal was recaptured based on interpolation artefacts left on edges. Once the image is classified as recaptured, we have shown that it is possible to retrieve important parameters in the acquisition chains using our reverse engineering technique. We proved sufficient conditions required such that we can detect the recapture and reverse engineer the chain process. Finally, we have discussed theoretical constraints of our proposed framework.

Chapter 4

Image Recapture Detection using Features from Edge Profiles

In Chapter 3 we have presented the theoretical framework to determine whether a signal was originally acquired or whether it was reacquired in the chain of A/D and D/A conversions. We also argued that with particular conditions it is possible to retrieve important parameters in the chain the signal has gone through. In this chapter we address the problem of recaptured detection from a more applied point of view and propose an algorithm to detect whether images were recaptured from an LCD monitor. Our method is based on the analysis of blurring patterns on edges. The process of displaying an image on a monitor and recapturing it with a second digital camera increases the level of blur relative to the originally captured image. The largest contributor to the increase in blur seen in the recaptured image is the drop in spatial resolution of the image due to the LCD monitor. Moreover, each stage of the image acquisition chain introduces a unique pattern of distortion into the image. In particular, besides the loss in sharpness, increase in distortion, such as ringing, that is introduced in the edge when it is captured, displayed and recaptured propagates through the chain and is present in the final image. The edges in the image, therefore, contain useful information, that can provide vital clues which enable us to reliably detect whether an image has been originally captured or whether it has been recaptured from a monitor display. For this reason, the algorithm described in this chapter makes extensive use of this feature.

Note that there is a significant difference between monitors with an analogue and a digital connection to the host computer. It requires additional D/A and A/D conversions when the monitor with an analogue connection is used. In this research we have simplified the problem by considering at most one reacquisition in the chain. We therefore focus only the case that the LCD monitor used has a digital connector to host computer, and there is no additional D/A and A/D conversions between the computer and the monitor.

4.1 Overview of Our Proposed Method

We stated, in Chapter 2, that, for practical reasons, the line spread function is used to model blurriness patterns which are footprints left on images. Our fundamental idea is that the LSFs estimated from edges from single captured images are markedly different from the LSF obtained from recaptured images. This conjecture is supported by the measurements in Figure 4.1. The edges we encounter on a daily basis are, nevertheless, not always straight and sharp. In practice it is very difficult to precisely estimate line spread functions of devices from edges found in natural images using the slanted edge testing technique. As a result we, instead, model blurriness patterns by measuring, and statistically combining, the *line spread profiles* of edges. This feature is computed from the first derivative of edge profiles.

The proposed algorithm consists of a training stage, in which a support vector machine (SVM) classifier is trained with known images, and a detection stage where the trained classifier is used to classify a given image. A diagram of the classifier training process is shown in Figure 4.2. Two sets of known images are used: a set of single capture images, \mathbf{I}_{SC} , and a set of recaptured images \mathbf{I}_{RC} . The images in each set are indexed with the superscript j and originate from a wide range of known cameras. The number of the images in each set, P and R , may differ.

The first step of the classifier stage is to determine a set of edge profiles from each image in each set that represent the sharpest edges found in the image. The first derivative of the edge profiles is then taken to determine a corresponding set of line spread profiles for the image. Thus, for a given image from the set of single capture training images, a matrix \mathbf{Q}_{SC}^j , is generated in which each column of the matrix corresponds to an extracted line spread profile.

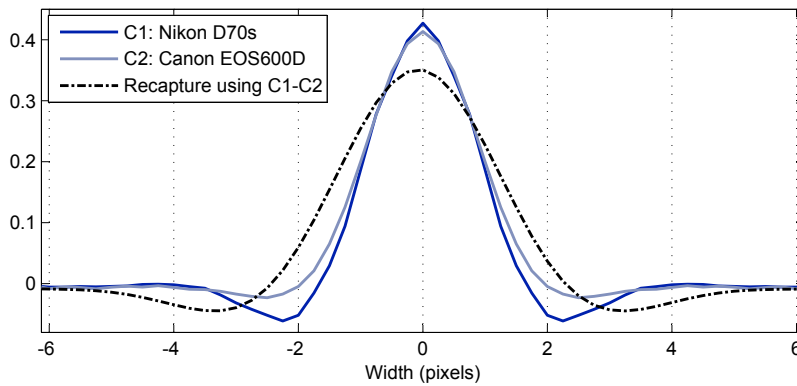


Figure 4.1: The comparison of line spread functions obtained from slanted edge testing using single captured images and recapture simulation. The cameras used from single capture are Nikon D70s and Canon EOS 600D. The LSF for recapture is created using the recapture combination of the two cameras.

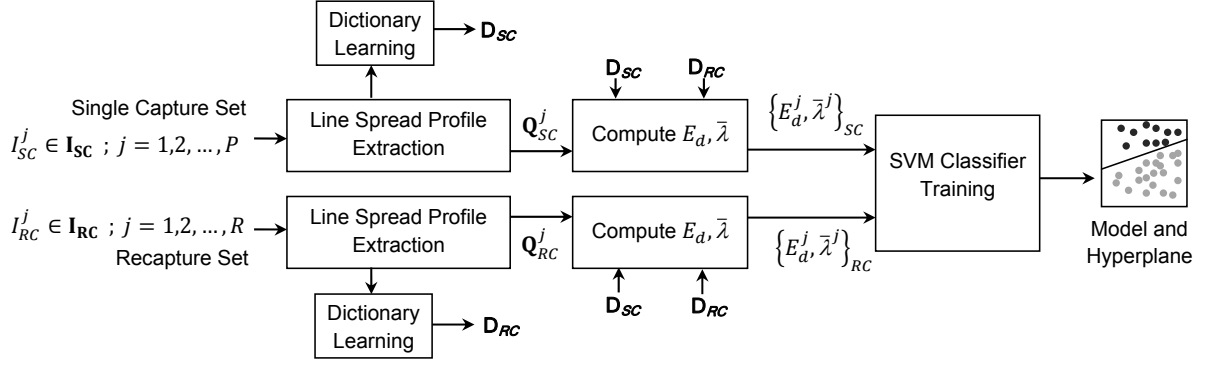


Figure 4.2: Diagram showing an overview of the training process for our proposed algorithm. Following the dictionary learning process, the learned dictionaries, \mathbf{D}_{SC} and \mathbf{D}_{RC} , are used to compute a pair of parameters $\{E_d, \bar{\lambda}\}$ for each training image. The classifier is then trained using all pairs of parameters $\{E_d, \bar{\lambda}\}$ which are labelled according to the class of training images.

The equivalent matrix for an image from the recaptured set is \mathbf{Q}_{RC}^j .

Two over-complete dictionaries are constructed by training using the K-SVD approach [114]. The first over-complete dictionary, \mathbf{D}_{SC} is trained using the set of single captured images and the second, \mathbf{D}_{RC} , using the set of recaptured images. Each dictionary is trained to provide an optimal sparse representation of the line spread profiles extracted from the training set of images.

To characterise the differences between the line spread profiles of originally captured and recaptured images, we introduce two parameters related to edges: a sparse representation error E_d and an average line spread width $\bar{\lambda}$. These parameters were chosen because they provide a concise but informative description of the differences between the line spread profiles of original and recaptured images. The first metric, E_d , represents the difference in the errors, E_{SC} and E_{RC} , between the extracted line spread profiles and their sparse representations determined using the dictionaries, \mathbf{D}_{SC} and \mathbf{D}_{RC} , respectively. The rationale being that $E_{SC} < E_{RC}$ if the image considered was original and $E_{SC} \geq E_{RC}$ if the image was a recaptured image. The value of E_d is determined by taking the differences between E_{SC} and E_{RC} . The second metric, $\bar{\lambda}$, provides a description of the width of an extracted line spread profile. Large values of $\bar{\lambda}$ correspond to blurry edges, while small values to sharp edges.

For each image, j , in the training set of single and recaptured images, I_{SC}^j and I_{RC}^j , a pair of parameters, $\{E_d^j, \bar{\lambda}^j\}_{SC}$ and $\{E_d^j, \bar{\lambda}^j\}_{RC}$ respectively, are obtained. The parameter pairs are collected on an image by image basis and the set of parameter pairs is then used to train a 2-dimensional SVM classifier. When the training procedure is complete a hyperplane that

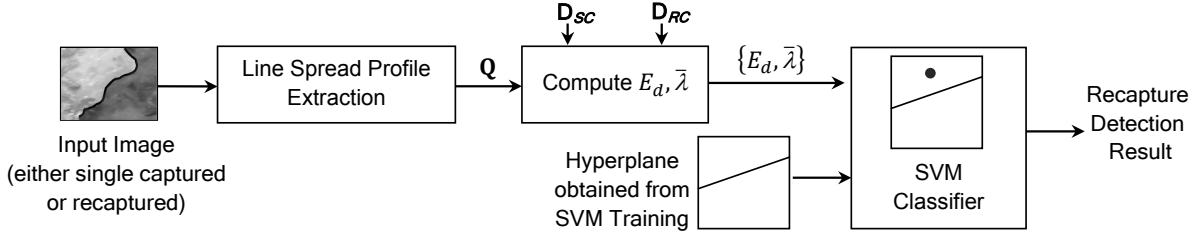


Figure 4.3: Overview working diagram of the classification scheme of our proposed recapture detection algorithm

optimally separates the two sets of images based on their values E_d and $\bar{\lambda}$ is determined.

A diagram of the detection stage is shown in Figure 4.3. For any given single or recaptured image, a line spread profile matrix, \mathbf{Q} , is obtained using the same method that was applied to the training images during classification. The parameters, E_d and $\bar{\lambda}$ are calculated using the trained dictionaries, \mathbf{D}_{SC} and \mathbf{D}_{RC} . The parameters are fed to the trained classifier and are classified as single or recaptured based on their location in the $E_d, \bar{\lambda}$ feature coordinate space relative to the SVM hyperplane.

The method for extracting the line spread profile is described in Section 4.2. In Section 4.3 we describe the dictionary learning procedure. A detailed description of the line spread width parameter, λ , and of the classifier training and recapture detection procedure are provided in Section 4.4.

4.2 Automatic Edge Detection and Feature Extraction

The diagram in Figure 4.4 illustrates how our proposed algorithm extracts line spread profiles from edges found in the image. Firstly, the query image is converted to greyscale and all edges contained in the image are detected using a Canny Edge Detector [115]. Edge profiles are extracted locally. Therefore, the query image is divided into a number of non-overlapping square blocks $B(m, n)$ of size $W \times W$ with $W = 16$ pixels. Here m and n are the vertical and horizontal indices of the block respectively. The block size at 16×16 is used according to our

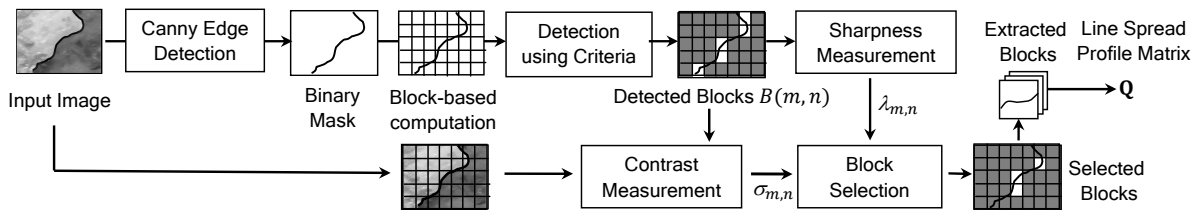


Figure 4.4: Working diagram of the proposed automatic block-based edge detection algorithm.

algorithm design in order to have non-overlapping blocks for our image width used at 2048 pixels. From our experiment, it is the optimal block size among the block sizes with multiples of 2 pixels. It is the block size that allows us to extract maximum number of blocks that meet out criteria for block-based edge detection and also sufficiently large such that the length of the edge profile vector is greater than width values of all edges we encounter.

For each block we first check whether it contains a horizontal or near horizontal sharp single edge. We then rotate the block by 90° to see whether it contains vertical or near vertical edges. The block selection procedure is implemented by examining the binary mask of the block and counting the number of columns, η , containing only one non-zero value. The block will be detected only when the condition $\eta \geq \beta W$ is satisfied where β has been set experimentally to $\beta = 0.6$. An example of a block that meets the selection criteria is shown in Figure 4.5(a) and two examples of blocks that fail to meet the selection criteria are shown in Figure 4.5(b) and Figure 4.5(c).

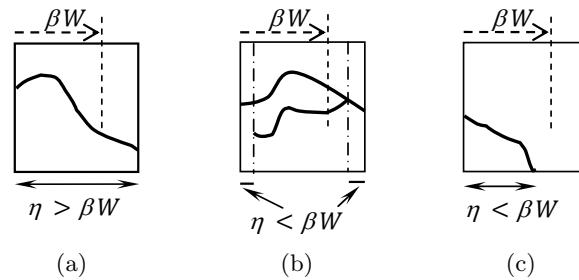


Figure 4.5: Examples of blocks with a binary mask of edges that are detected (a) and discarded (b and c). The block in Figure (a) satisfies all our selection criteria. Blocks in Figure (b) and c) do not qualify because the majority of columns in the block shown in Figure (b) contain double edges and in Figure (c) the number of columns containing an edge is less than βW .

The detected blocks, $B(m, n)$, shown in Figure 4.4, are then ranked according to their sharpness and edge contrast. This enables us to select regions that are in focus and that contain the most prominent edge features. Block sharpness is determined using the technique described in Section 4.4 in which the average width $\bar{\lambda}_{m,n}$ of line spread profiles of the blocks is estimated. The contrast of a block is measured by computing the block-based variance, $\sigma_{m,n}$, of the input image at the detected block. The distributions of $\bar{\lambda}_{m,n}$ and $\sigma_{m,n}$ over a range of images are shown in Figure 4.6. Figure 4.6(a) and Figure 4.6(b) show the distributions of $\bar{\lambda}_{m,n}$ and $\sigma_{m,n}$ which were created from 135 original images in our training set. Figure 4.6(c) and Figure 4.6(d) present the distributions of $\bar{\lambda}_{m,n}$ and $\sigma_{m,n}$ for a set of 135 recaptured images. The recaptured images were obtained from recapturing the original images using Canon EOS 600D camera. We observe that there is only a small number of blocks with sharp edges

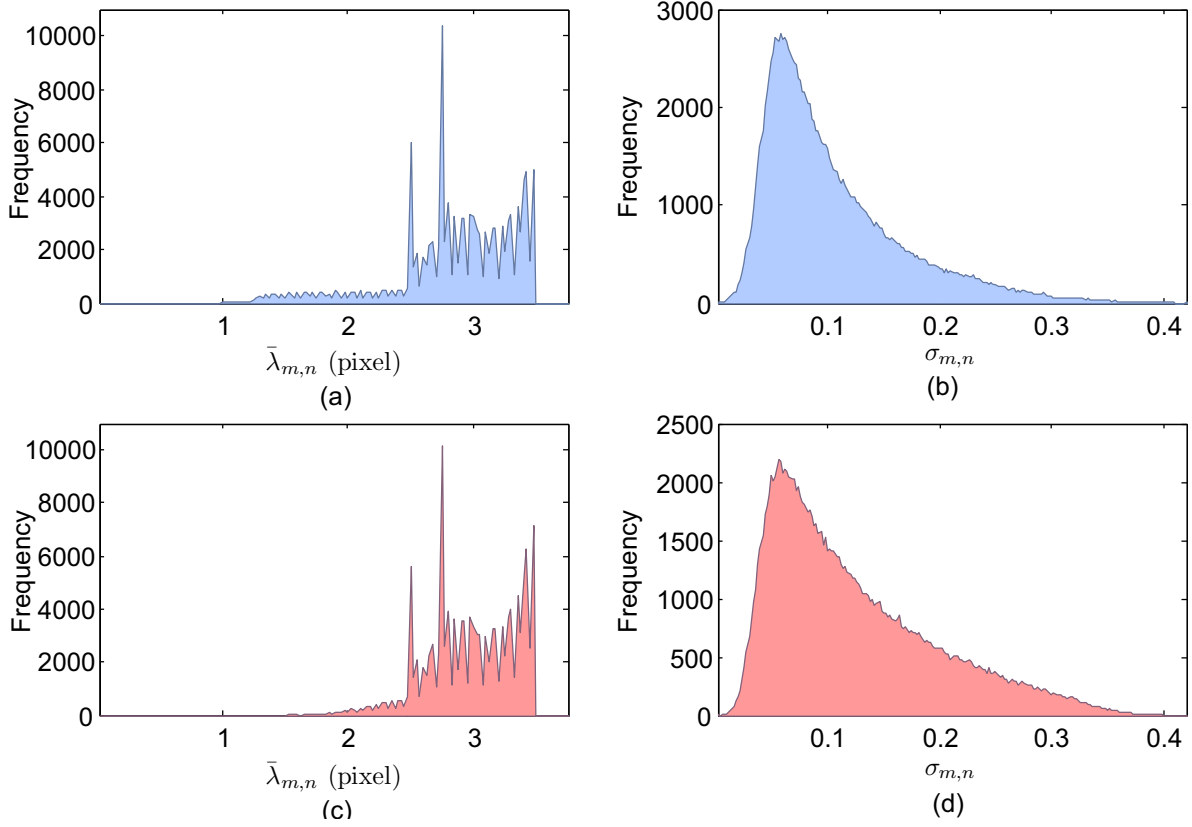


Figure 4.6: The distributions (a) average width of line spread profiles $\bar{\lambda}_{m,n}$ and (b) contrast $\sigma_{m,n}$ were computed from 135 original images. The distributions (c) $\bar{\lambda}_{m,n}$ and (d) $\sigma_{m,n}$ were created using 135 images recaptured from the original images used in building (a) and (b).

which correspond to small values of $\bar{\lambda}_{m,n}$ present in the distributions in Figure 4.6(a) and Figure 4.6(b). These blocks are most likely extracted from the in-focus areas where sharp input edges were present. In addition, we are likely to obtain blocks with sharper edges from a set of single captured images.

Next, suitable blocks are chosen based on the distributions of $\bar{\lambda}_{m,n}$ and $\sigma_{m,n}$ built over all the detected blocks in a given image. Since blocks with sharpest edges and highest contrast are desirable, only blocks whose average width, $\bar{\lambda}_{m,n}$, falls within the narrowest 10% of the detected block widths and whose value of $\sigma_{m,n}$ falls within the largest 20% of computed values are selected. The thresholds at 10% and 20% are chosen based on our experiment. They are the optimal values among the coarse thresholds we tested in the experiment. Larger threshold values allow larger numbers of blocks with lower contrast and more blurred edges to be selected. It reduces the performance of the algorithm. The smaller values of thresholds, however, reduce the number of selected blocks. It would reduce statistical reliability of the algorithm when only few blocks with edges are detected.

For selected blocks, let $\mathbf{Y} \in \mathbb{R}^{W \times W}$ be a matrix which represents the grey scale values of a block. Each column, $\mathbf{y}_i; i = 1, 2, \dots, W$, of the matrix \mathbf{Y} may, therefore, be considered to represent an edge profile of the image. We determine a normalised line spread profile, \mathbf{q}_i , by evaluating $\mathbf{q}_i = \mathbf{y}_i^{(1)} / \|\mathbf{y}_i^{(1)}\|_2$ where $\mathbf{y}_i^{(1)}$ is the first difference of \mathbf{y}_i . The first difference value of the edge profile is then normalised in order to standardise the feature.

Our feature vector, \mathbf{q}_i , now contains the line spread profile at column i of the input block. Since edges might appear at any location in the blocks, pre-processing is required to move the line spread profiles to the center of the feature vector \mathbf{q}_i to reduce the number of shifts in overcomplete dictionaries. The line spread profiles are first cropped with the length equal to the width of line spread profiles plus additional two pixels. If an edge is close to the boundary of the block, that corresponding line spread profile will be discarded. The cropped line spread profile is then placed in the center of the feature vector \mathbf{q}_i such that the peak of the profile is at the center of the feature vector. The rest of empty data points in the feature vector is then filled with zeros in order to maintain a length of W elements.

Once the line spread profiles for all the selected blocks in the image have been determined, a line spread profile matrix, $\mathbf{Q} \in \mathbb{R}^{W \times M}$ is formed by concatenating the total M line spread profiles, \mathbf{q}_i , from all the selected blocks. This feature matrix is used for training and testing purposes.

4.3 Dictionary Learning Algorithm

The objective of dictionary learning is to obtain two overcomplete dictionaries, \mathbf{D}_{SC} and \mathbf{D}_{RC} , that provide an optimal sparse representation of line spread profiles from single captured and recaptured images, respectively. Dictionary training can be used as a tool to learn the characteristics of the distortion patterns present in edges found in most naturally occurring images. The key insight being that the descriptions in single capture and recaptured images are fundamentally different due to the sharpness degradation introduced by the recapture process.

The first step in dictionary training is to determine the training feature matrices, \mathbf{S}_{SC} and \mathbf{S}_{RC} , for single captured and recaptured images, respectively. For each set of training images, \mathbf{I}_{SC} and \mathbf{I}_{RC} , the set of line spread profiles, \mathbf{Q}_{SC}^j and \mathbf{Q}_{RC}^j , is constructed using the method described in Section 4.2. The supercript, j , denotes the individual images contained in each training set. The training feature matrices, \mathbf{S}_{SC} and \mathbf{S}_{RC} , are determined by concatenating horizontally the extracted line spread profiles matrices, \mathbf{Q}_{SC}^j and \mathbf{Q}_{RC}^j , over all the training

images in each respective set. Thus, the resulting training feature matrix, $\mathbf{S} \in \mathbb{R}^{W \times N}$, contains N training line spread profiles $\mathbf{q}_i \in \mathbb{R}^W$, where $i = 1, 2, \dots, N$, and $N \gg W$.

Given the training feature matrix \mathbf{S} , the goal of dictionary training is to obtain the best dictionary, $\mathbf{D} \in \mathbb{R}^{W \times K}$, that provides an optimal sparse representation for all the line spread profiles in the training matrix \mathbf{S} , that is

$$\min_{\mathbf{D}, \mathbf{X}} \|\mathbf{S} - \mathbf{D}\mathbf{X}\|_F^2 \text{ subject to } \forall i, \|\mathbf{x}_i\|_0 \leq L, \quad (4.1)$$

where $\mathbf{X} \in \mathbb{R}^{K \times N}$ is built from the column vectors \mathbf{x}_i used to represent the feature \mathbf{q}_i and $i = 1, 2, \dots, N$. The notation $\|\mathbf{A}\|_F^2$ refers to the Frobenius norm, which is defined as $\|\mathbf{A}\|_F^2 = \sum_{ij} |A_{ij}|^2$. The constant L is the maximum number of atoms permitted. The choice of L is generally a trade off between approximation precision and sparsity and we discuss its selection later in this section.

Our dictionary is designed using the K-SVD learning approach [114]. The K-SVD method is an iterative learning scheme based on two important steps for each round of computation: sparse coding and dictionary update.

In sparse coding, given an initial dictionary \mathbf{D} , \mathbf{X} is chosen such that each of its columns \mathbf{x}_i provides the best L -sparse representation of \mathbf{q}_i . Specifically:

$$\min_{x_i} \|\mathbf{q}_i - \mathbf{D}\mathbf{x}_i\|_2^2 \text{ subject to } \|\mathbf{x}_i\|_0 \leq L \quad (4.2)$$

In practice, this is achieved using the orthogonal matching pursuit (OMP) algorithm [116] which is known to provide near-optimal sparse coding. Next, given \mathbf{X} , \mathbf{D} is updated so as to achieve

$$\min_{\mathbf{D}} \|\mathbf{S} - \mathbf{D}\mathbf{X}\|_F^2. \quad (4.3)$$

In K-SVD, the dictionary atoms are updated, one column at a time, at the k^{th} column index, where $k = 1, 2, \dots, K$. The residual error in (4.3) is computed using only the training profiles that use the k^{th} atom for approximation. Next, the atom which minimises the residual error can be obtained using a singular value decomposition (SVD) approach. We replace the k^{th} column with this new atom. The process is then repeated for all K columns. Given the new \mathbf{D} , a new \mathbf{X} is found by sparse coding and the process is repeated. As a result, the training error is reduced over several iterations and the dictionary \mathbf{D} has been trained to fit all training profiles in \mathbf{S} . Figure 4.7 shows the errors computed from 100 iterations of K-SVD training for sets of single and recaptured images with values for L equal to 1, 3, and 5. The overall training error is reduced as the value for L is increased.

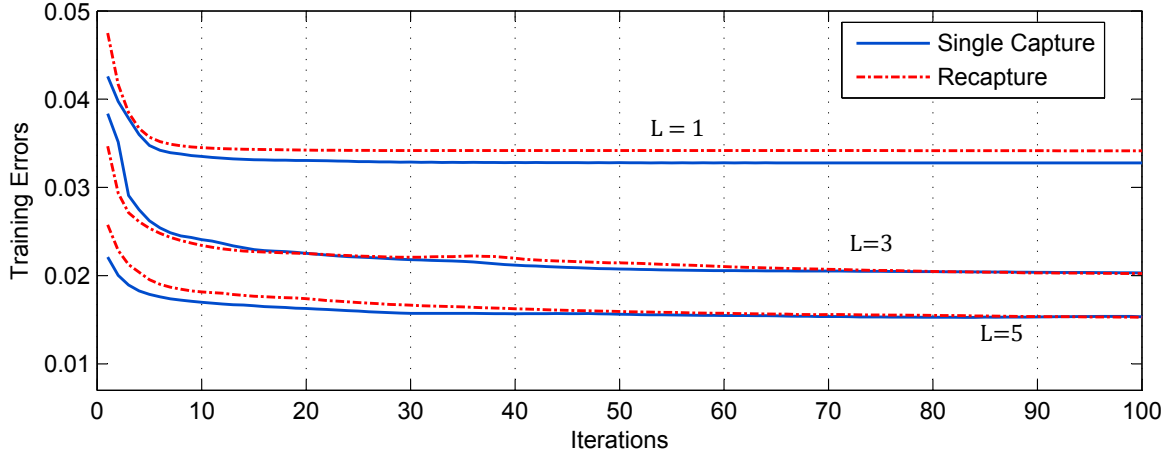


Figure 4.7: The root-mean-squared error computed from K-SVD training with sets of single captured and recaptured images. The errors reduce over 100 training iterations for $L = 1, 3$, and 5.

By training two dictionaries, \mathbf{D}_{SC} and \mathbf{D}_{RC} , using the training feature matrices \mathbf{S}_{SC} and \mathbf{S}_{RC} , we ensure that the patterns from line spread profiles extracted from single captured and recaptured images will have been learned. Each dictionary provides an optimal sparse representation of the line spread features from each class of images.

We now discuss the selection of the optimal number of atoms, L , in the dictionary. Since each dictionary was trained using the specific blurring patterns from a given class of images, only one dictionary will provide a good sparse approximation of line spread profiles from the query image. We, therefore, require a value for L that is large enough to provide a good approximation. However, if too many atoms are used, the algorithm is unable to discriminate between the two image classes since both dictionaries are now able to provide good approximations.

To determine the optimal value for L the idea is that the approximation error, $e_t(L)$, decreases with L . However, once the essential information of the signal has been captured, $e_t(L)$ will stop decaying rapidly since the algorithm is now capturing noise and non-discriminative information. This transition point can be detected by finding the peak of the second derivative of e_t . The effect on the training error, e_t , when the number of atoms used for representation is varied is shown in Figure 4.8(a). The optimal number of atoms is then calculated from the peak of the second derivative of the error function. From Figure 4.8(b) we can determine that the peak of the second derivative for our training sets occurs at approximately $L = 3$.

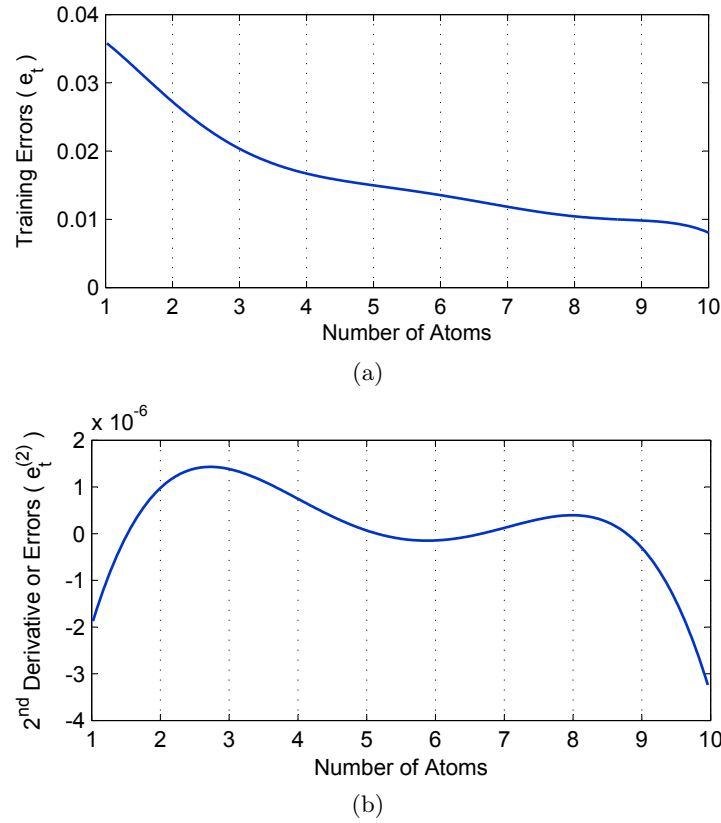


Figure 4.8: (a) The root-mean-squared error from K-SVD training using single captured images over 160 iterations when the number of atoms used is varied from $L = 1, 2, 3, \dots, 10$. (b) The optimal number of atoms used is obtained by observing the number of atoms at which the errors begin to converge. This can be estimated from the number of atoms that correspond to the peak of the second derivative of the training error. From our experiment, the optimal value $L = 3$.

4.4 Classification for Recapture Detection

We extract two important feature parameters, $\bar{\lambda}$ and E_d , from the line spread profile matrix \mathbf{Q} which we use for classification. The first parameter, $\bar{\lambda}$, which models the average line spread profile width, is computed as follows: the value of λ is defined as the distance that allows 95% of the spectral energy of the spread function to be captured and is represented by the grey area shown in the Figure 4.9.

To determine $\bar{\lambda}$ we compute the spread widths, λ_i , for all the line spread profiles $\mathbf{q}_i \in \mathbb{R}^W$ taken from $\mathbf{Q} \in \mathbb{R}^{W \times M}$ and by then taking the average.

Note that a blurred edge generally has a wider spread function compared to a sharp edge. Thus, the value of $\bar{\lambda}$ computed from a recaptured image is expected to be greater than the value obtained from the equivalent single capture image. The distributions of $\bar{\lambda}$ obtained from experiments using single captured and recaptured training sets are shown in Figure 4.10.

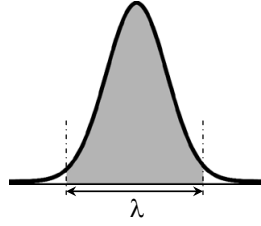
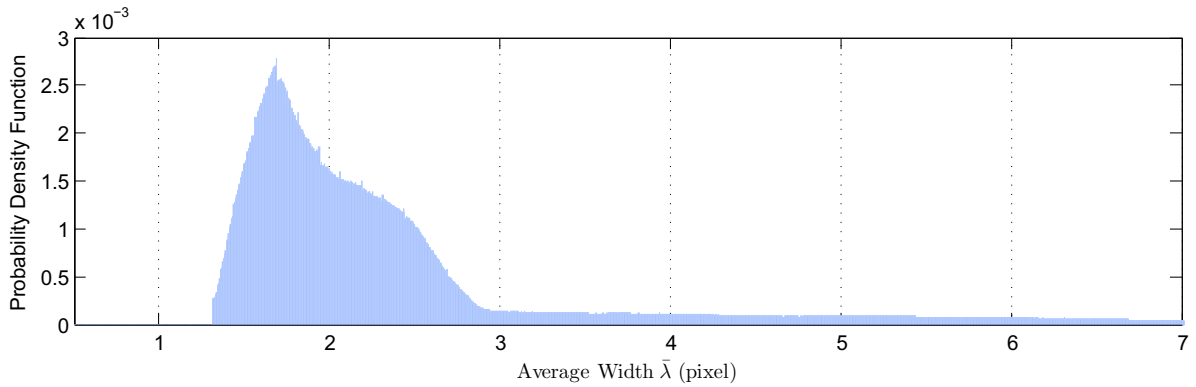
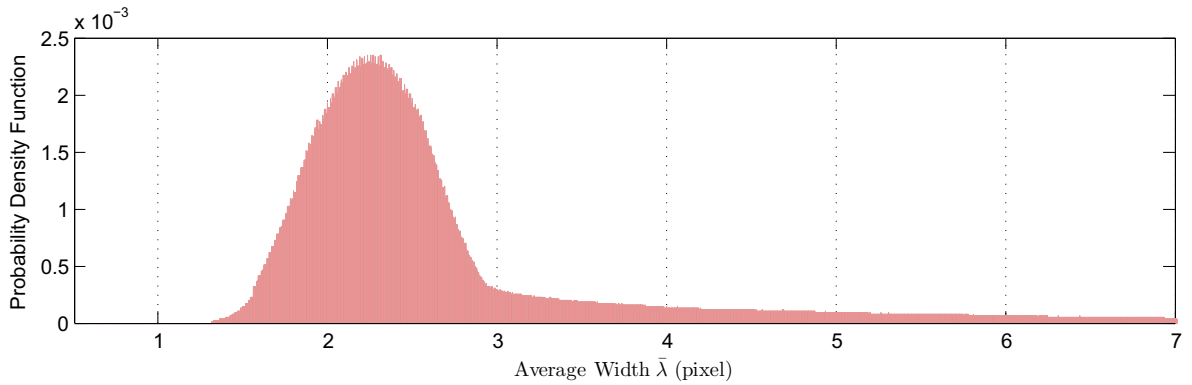


Figure 4.9: The criteria for the calculation of the width λ of the spread function. The width is the minimum distance that allows the shape of the edge spread function to be approximated using an estimate of the energy spectral density.



(a) Single capture dataset



(b) Recapture dataset.

Figure 4.10: The distributions of average width of line spread profiles computed from (a) single capture dataset and (b) recapture dataset.

We can conclude that the mode of the distribution of edge spread width values from the set of recaptured images is higher than the mode of the distribution obtained from the single captured image set. In addition, the sharp peak that can be observed in the distribution of edge widths from single capture blocks indicates that a larger number of single capture blocks contained very sharp edges when compared to the recaptured blocks where the distribution of edge widths is smooth and bell shaped. Note that the small number of blocks with blurred edges that appear at the tail of single capture distribution are from the acquisition of low

contrast, poorly defined, edges in the scene.

The second parameter used is the difference of approximation errors E_d . The value of E_d is used to compare the abilities of the two dictionaries, \mathbf{D}_{SC} and \mathbf{D}_{RC} , to provide a sparse representation of line spread profiles from a query image. Given a line spread profile matrix \mathbf{Q} from an unknown image, we define an approximation error using a dictionary trained from single captured images \mathbf{D}_{SC} as $E_{SC} = \|\mathbf{Q} - \mathbf{D}_{SC}\mathbf{X}_1\|_F^2$, where \mathbf{X}_1 is the corresponding coefficients matrix computed using \mathbf{D}_{SC} . In the same way, a representation error using a dictionary trained from recaptured images is given by $E_{RC} = \|\mathbf{Q} - \mathbf{D}_{RC}\mathbf{X}_2\|_F^2$.

The approximation errors, E_{SC} and E_{RC} , describe how well each dictionary fits the line spread profile matrix \mathbf{Q} . To perform recapture classification we compare E_{SC} with E_{RC} . A query image is classified as recaptured if $E_{SC} \geq E_{RC}$. Otherwise it is considered as single captured. We define the difference of approximation errors (E_d) as follows:

$$E_d = E_{SC} - E_{RC} = \|\mathbf{Q} - \mathbf{D}_{SC}\mathbf{X}_1\|_F^2 - \|\mathbf{Q} - \mathbf{D}_{RC}\mathbf{X}_2\|_F^2. \quad (4.4)$$

One way that Equation (4.4) may be interpreted is that the image is more likely to be single captured if E_d is negative. If E_d is positive the image is more likely to be recaptured. The histogram of error E_d computed from experiments using single captured images and recaptured images are shown in Figure 4.11(a) and Figure 4.11(b) respectively.

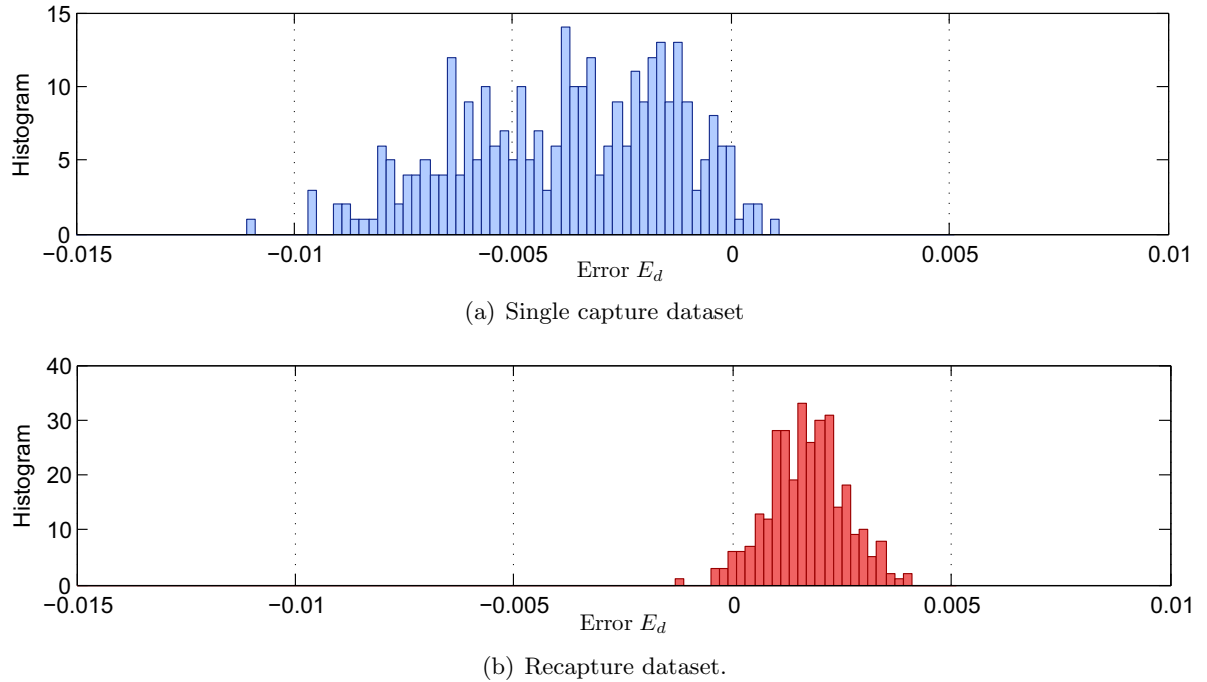


Figure 4.11: The histograms of error E_d computed from (a) single capture dataset and (b) recapture dataset.

In our training process a set of pairs of parameters, $\bar{\lambda}$ and E_d , extracted from each image in the training set is formed. The parameter pairs are then labelled as either single or recaptured, depending on the class of training images used. They are then used for training the classifier. Figure 4.12 is a plot in the feature coordinate space, $(\bar{\lambda}, E_d)$, of the parameter pairs from all the images used for training purposes. A support vector machine (SVM) classifier with a linear kernel was trained. The classification hyperplane (solid line in Figure 4.12) was then generated.

From Figure 4.12, we can observe that the difference in representation error, (E_d) , can be used as a feature to effectively distinguish between single and recaptured images. The majority of images in the single and recaptured groups were separated correctly by the criterion $E_d = 0$ (dotted line). However, the hyperplane obtained from the SVM training process (solid line) resulted in better classifier performance since the mean width of the edge spread function, $\bar{\lambda}$, was taken into account. We have assumed that the distribution of features from the training

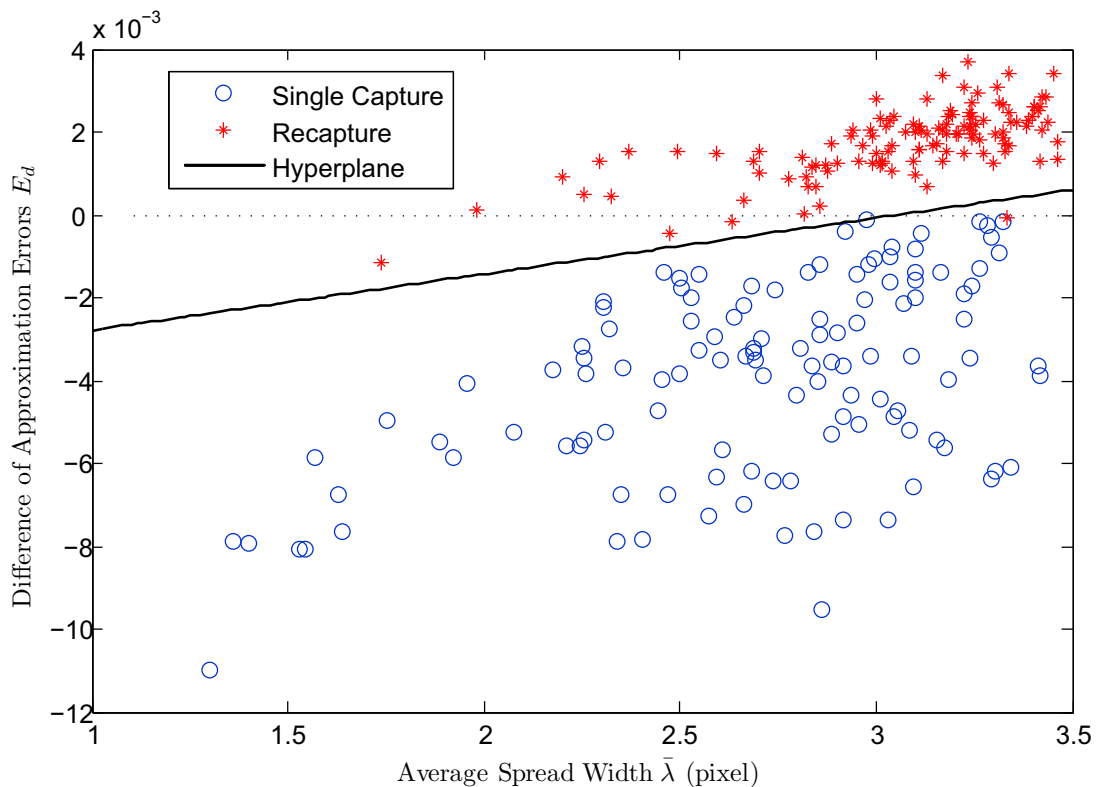


Figure 4.12: A plot of the distributions of features extracted from training images with the average width of spread function ($\bar{\lambda}$) shown on the horizontal axis and the difference of representation error (E_d) on the vertical axis. The hyperplane for recapture classification was obtained using SVM training and is defined as the line that separate the features from the recaptured (star) and single captured (circle) images with minimum classification error. A query image is classified based on the coordinate location of the feature pair, $\bar{\lambda}, E_d$, determined from the image.

images is typical of the type of images we are likely to encounter on a daily basis. The trained classifier is, therefore, used for our recapture detection algorithm.

We note that other no-reference blur metrics that operate on the whole image [97, 99] could be used instead of $\bar{\lambda}$. In Section 5.2.3 we evaluate the recapture detection performance of the algorithm when the spread width $\bar{\lambda}$ are replaced by the CPBD blurring metrics. However, we have observed through the experiments that the blurring metrics are less effective at classifying single and recaptured images than $\bar{\lambda}$, especially when combined with other discriminative features such as E_d . For this reason we have decided to use $\bar{\lambda}$ in our work.

4.5 Summary

In this chapter we have proposed a novel and practical algorithm to detect recaptured images. The algorithm is based on differences between the edge blurring patterns present in original and recaptured images. The algorithm uses two important features to extract the differences.

The first feature E_d represents the degrees of similarity between line spread profiles extracted from a given image and the spread profiles that dictionaries approximate. It is computed from the difference between the errors E_{SC} and E_{RC} when approximate line spread profiles of the given image using dictionaries \mathbf{D}_{SC} and \mathbf{D}_{RC} respectively. We have first trained the dictionaries \mathbf{D}_{SC} and \mathbf{D}_{RC} to learn provide optimal sparse representations of the profiles from single captured and recaptured images. The dictionary that can better approximate the given profiles, therefore, would produce smaller approximation error. Since we have defined $E_d = E_{SC} - E_{RC}$, the error E_d is likely to be negative for original captured images and positive for recaptured images.

The second features we have used is the average width of spread profiles $\bar{\lambda}$. It is computed from the average value of the distance that allows 95% of the spectral energy of the line spread profiles to be captured. We have statistically shown that the width parameters computed from recaptured images have greater values when compared to the width from single captured images. We finally trained an SMV classifiers using pairs of the features $\bar{\lambda}$ and E_d obtained from labelled sets of single captured and recaptured images. The resulting hyper-plane that optimally separates features from single capture and recapture sets is ready to use for classification.

Chapter 5

Experiments

In this chapter we discuss experiments on the performance evaluation of our proposed algorithm. The performance is measured in terms of precision of detecting images from known single captured and recaptured classes. In the first section we present how to prepare image database for performance testing. In particular the conditions for recapturing images without introducing visible aliasing artefacts are discussed. In experimental sections we describe parameter setting and test the algorithm in three different aspects. First we test the performance of the algorithm we have presented in Chapter 4. Then we modify the algorithm and replace our proposed feature $\bar{\lambda}$ by the state-of-the-art blurring metrics before it is tested with the same datasets. The precision of our algorithm and the modified version are compared in order to prove the discrimination ability of our distinctive features. Finally we test the universality of our proposed algorithm. The precision of the algorithm is measured when it is tested with images taken from cameras that were not included in the training stage. Finally, we show that the proposed method outperforms existing recaptured detection algorithms.

5.1 Database of Recaptured Images

A database of images recaptured from an LCD monitor was developed for the purposes of testing and evaluating the performance of the recapture detection algorithm described in Chapter 4. The recapture database comprises 315 single capture images from nine cameras and 2520 images recaptured using eight cameras. The database will be made publicly available in order that it can be used as a common database for researchers in the field of image forensics who wish to benchmark their algorithms. Currently available image databases include the ‘Dresden Image Database’ [117]. This is probably the most well known still image database

Note: The contributions of this chapter are shared with Dr. Hani Muammar who prepared a dataset for alias-free recaptured images.

for forensic applications, but it does not include any recaptured images.

5.1.1 Image Capture and Display Equipment

Nine cameras were used to photograph the original scenes and eight to recapture the images from the LCD monitor. Five of the cameras used to carry out the original captures were used for recapture thereby resulting in a total of twelve cameras in the database. A list of cameras used, their specifications, and usage is shown in Table 5.2. They include six compact digital cameras with fixed zoom lenses, five digital single lens reflex (DSLR) cameras with interchangeable lenses and one compact camera with interchangeable lenses. With the exception of the three Kodak cameras and the Panasonic TZ7, all cameras provided both automatic and manual exposure settings. The two Kodak V550 models are equivalent in specification and differ only in their finish. They are indicated as silver and black in Table 5.2. All the images were recaptured from an NEC MultiSync EA232Wmi 23" IPS LCD monitor with LED backlighting and a resolution of 1920 x 1080 pixels.

5.1.2 Original scene capture

The database comprises mainly natural scenes photographed indoors and outdoors under different types and levels of illumination. Some examples of originally captured scenes (top row) and the recaptured images (bottom row) are shown in Fig 5.1. A significant proportion

Table 5.1: A table of the digital cameras used in the recapture database.

Camera Make and Model	Year	Sensor Type	MPixels Capture	Original	Recapture
Kodak V550 (silver)	2005	1/2.5" CCD	5	Yes	No
Kodak V550 (black)	2005	1/2.5" CCD	5	Yes	No
Kodak V610	2006	1/2.5" CCD	6	Yes	No
Nikon D40	2006	APS-C CCD	6	Yes	No
Panasonic TZ10	2010	1/2.33" CCD	12.1	No	Yes
Nikon D3200	2012	APS-C CCD	24.2	No	Yes
Canon 60D	2012	APS-C CMOS	18	No	Yes
Nikon D70s	2004	APS-C CCD	6	Yes	Yes
Panasonic TZ7	2009	1/2.33" CCD	10	Yes	Yes
Canon 600D	2012	APS-C CMOS	18	Yes	Yes
Olympus E-PM2	2012	Four thirds CMOS	16.1	Yes	Yes
Sony RX100	2012	1" EXMOR CMOS	20.2	Yes	Yes



(a) Architectural scene



(b) Recaptured architectural scene



(c) Natural scene



(d) Recaptured natural scene



(e) Wildlife scene



(f) Recaptured wildlife scene



(g) Indoor scene



(h) Recaptured indoor scene

Figure 5.1: Images from the recapture database showing examples of originally captured and recaptured scenes.

of the images were taken outdoors under sunny or overcast conditions. Those taken indoors were acquired mostly under natural illumination, but also included a scene with a MacBeth Colorchecker test chart captured under natural illumination and using the camera's internal flash, where available. Each scene in the database was photographed once by each of the test

cameras under equivalent, or nearly equivalent, illumination conditions. This allowed for a one to one correspondence between a scene and each test camera. All the cameras were set to automatically select the exposure setting, ISO and white balance setting, with the exception of the Macbeth test chart scene where different ISO settings were selected. The database contains 35 images per camera giving a total of 315 single captured images over all 9 cameras.

5.1.3 Recapture

A high priority when developing the recapture database was that the recaptured images would be high in perceived quality and finely recaptured. All image recaptures were conducted in a darkened room, to eliminate unwanted reflections from the monitor and the surrounding environment. The single captured images were prepared for display by resizing them using a bicubic interpolation kernel to the pixel dimensions of the NEC monitor. The camera used to recapture the images from the LCD monitor was placed on a sturdy tripod. Before recapturing the images, the LCD monitor was calibrated to the sRGB standard with $\gamma = 2.2$ and a monitor white point luminance of 240 cd/m^2 . The lens focal length of each camera was set to a value that minimised the level of geometric distortion introduced in the recaptured image as much as was practically possible.

In addition, it is important to minimise aliasing which is one of the most common visible artefacts left on recaptured images. In general it is very difficult to obtain alias-free images unless a precise calibration is performed. All recaptured images used in our experiments were prepared by Dr. Hani Muammar. The monitor to camera distance was determined by applying the alias frequency maximisation method described in [70]. In this method the recapture distance is set such that the frequency of the aliasing pattern in the green channel is maximised, while at the same time the amplitudes of the alias pattern in the red and blue channels are minimised. Using this technique, the principle frequency components of the aliasing patterns are situated on the Nyquist boundary of the red and blue channels in Figure 2.6 and are not visible after the CFA interpolation. The high frequency of the aliasing pattern in the green channel is then removed using a low-pass filter technique by adjusting appropriate value of the camera's aperture.

The procedure for determining the capture distance and lens aperture setting is described with the aid of the following example in which a Canon 600D camera was used to recapture images from the NEC monitor. The image sensor in the Canon 600D camera has a pixel pitch of $4.30652 \mu\text{m}$. The pixel pitch of the NEC monitor is 0.2650 mm . According to the estimation technique presented in [70], a capture distance of 1445.2 mm was used for a lens focal length of

30 mm. An aperture setting of $f/11$ was used to eliminate visible aliasing from the recaptured image due to aliasing in the green channel. The aperture settings and capture distances used for each camera model are shown in Table 5.2. Note that the focal lengths presented on the table corresponding to Nikon, Canon, and Olympus cameras are obtained from reading the values manually set on the focal adjustment ring of each camera’s lens. The focal lengths for the compact cameras including Sony RX100, Panasonic TZ7, Panasonic TZ10 were set using electronic zooming mechanism of the cameras. The focal lengths presented on the table corresponding to these models are obtained from reading Exif metadata of the image files.

After setting the camera to monitor distance, the camera’s image sensor was aligned with the plane of the monitor faceplate. The camera’s ISO setting was manually set to 400, and the camera was allowed to select the exposure automatically. This ISO value was used such that the cameras did not introduce excessive levels of image noise in the recaptured images. To eliminate colour balance errors by the recapture camera, the camera’s white point was preset by estimating it from a white patch displayed on the monitor. The recaptured images were cropped to remove the LCD monitor surround.

Note that our image datasets were prepared without additional noise control. There were only two stages in the chains that we controlled the levels of noise using ISO adjustment such that noise levels of images did not degrade image quality. The ISO levels used for original capture vary from 100 to 800 depending on different capture conditions. The ISO levels for recapture at from 100 to 400 were set during image recapture. These noise levels represent noise levels we can find in most common digital images. All the experiments in this thesis were test using the images with these noise levels.

Table 5.2: Distance and camera settings used for recapture, listed by camera models.

Camera Model	Capture Distance (mm.)	Focal Length (mm.)	Aperture Setting
Nikon D3200	1600	30	$f/10$
Nikon D70s	790	29	$f/20$
Canon 600D	1450	30	$f/13$
Canon 60D	1450	30	$f/13$
Sony RX100	1500	18 (Exif)	$f/5.6$
Olympus E-PM2	1350	25	$f/9$
Panasonic TZ7	830	6.1 (Exif)	$f/4$
Panasonic TZ10	830	6.1 (Exif)	$f/4$

5.2 Experimental Results

5.2.1 Experimental Design and Parameter Setting

Our algorithm was tested by applying it to the set of single and recaptured images from the database described in Section 5.1. The single and recaptured images were initially divided into two groups; one for training and the other for testing purposes. Each set of 35 single capture images from the nine cameras used to capture the original scenes was partitioned into 15 images for training and 20 images for performance evaluation. This resulted in a total of 135 images for training and 180 images for testing. As mentioned in Section 5.1, the recaptured images in the database were taken over 72 different single/recapture camera combinations resulting in a total of 2520 images. These were divided into 1080 images for training and 1440 images for testing. The set of training images was further reduced to 216 images by randomly selecting three images from each recapture camera combination. This was done to eliminate any bias introduced during training due to large differences in the number of single and recaptured images. All the images were resized to a width of 2048 pixels because of differences in the size of the images resulting from variations in the camera image sensor pixel counts. The image height during resizing was set to a value that preserved the original aspect ratio of the image.

The K-SVD software library that is publicly provided by the authors [114] was used to train our dictionaries. The training feature matrices \mathbf{S}_{SC} and \mathbf{S}_{RC} were built from the edge spread features \mathbf{q}_i with length $W=16$ collected using the blocks selected from single captured and recaptured images respectively. One in every four spread features were used in order to reduce the size of the training data. Before being used for K-SVD dictionary training each spread feature was interpolated by 4x to increase the number of data points to 64. The initial set of atoms was constructed from the line spread functions of the nine single capture cameras and 63 different line spread functions determined from randomly selected image recapture camera combinations. The dictionaries were trained over 160 iterations of the K-SVD algorithm at which point the errors converged to a predetermined minimum value. Figure 5.2 shows examples of dictionary atoms from \mathbf{D}_{SC} (dark blue) and \mathbf{D}_{RC} (red) obtained after the dictionary learning process. The dictionary atoms are from the same shift in each dictionary. We observe that the dictionary atom trained from recaptured images has wider spread width compared to the atom trained from single captured images.

Total 351 pairs of training features were extracted from the 351 training images according

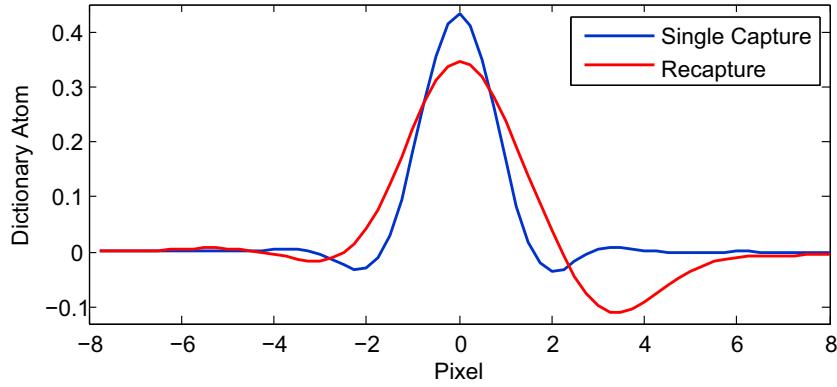


Figure 5.2: Examples of dictionary atoms from \mathbf{D}_{SC} (dark blue) and \mathbf{D}_{RC} (red) obtained after the dictionary learning process.

the method discussed in Section 4.4. The obtained features were labelled according to the classes of images before being used for SVM training. The resulting classification hyperplane is shown in Figure 4.12.

5.2.2 Performance Evaluation of the Proposed Algorithm

Our proposed algorithm was tested with the test image set described in Section 5.2.1 and the results are shown in Table 5.3 and Table 5.4. In Table 5.3 the results are presented according to the cameras used for single capture. A total of 170 out of the 180 single captured images were correctly classified corresponding to a 94.44% true negative rate (TN). In Table 5.4 the results are presented in groups according to the camera used to recapture the images. Performance is

Table 5.3: The performance of the algorithm in detecting original captured images. The algorithm is tested with the original capture dataset and the results are listed according to the cameras used for capture.

Camera Model	No. of images	Classification Results		Performance (%)
		Single	Recapture	
Kodak V550 (Black)	20	18	2	90
Kodak V550 (Silver)	20	15	5	75
Kodak V610	20	18	2	90
Nikon D40	20	20	0	100
Nikon D70s	20	19	1	95
Canon EOS 600D	20	20	0	100
Sony RX100	20	20	0	100
Olympus E-PM2	20	20	0	100
Panasonic TZ7	20	20	0	100
Average				94.44

Table 5.4: The performance of the algorithm in detecting recaptured images when the algorithm is tested with the recapture dataset. The results according to the camera used for recapture are listed in the rows and the cameras used during the original capture are listed in the columns.

Recapture Camera	No. of detection according to the camera used in the first capture									No. of Success	Perf. (%)
	V550b	V550s	V610	D40	D70s	600D	RX100	E-PM2	TZ7		
D3200	20	20	20	20	20	20	20	20	20	180	100
D70s	20	20	20	20	20	20	20	19	20	179	99.44
600D	20	20	20	20	20	19	20	19	20	178	98.89
60D	20	20	20	20	20	20	20	19	20	179	99.44
RX100	20	20	20	20	20	20	20	20	20	180	100
E-PM2	20	20	20	19	19	20	16	19	20	173	96.11
TZ7	20	20	20	20	20	20	20	20	20	180	100
TZ10	20	20	20	20	20	20	19	20	20	179	99.44
Total No. of Success	160	160	160	159	159	159	155	156	160	1428	
Average Performance											99.17

consistently good, spanning from a 96.11% true positive (TP) rate for the E-PM2 to a 100% success rate for the Nikon D3200, Sony RX100 and Panasonic TZ7 models.

It is interesting to note the drop in detection rate for single captured images that were originally captured using the Kodak cameras. One possible reason is that the images captured using these cameras are, on average, less sharp than the images obtained by the other ‘single capture’ cameras. This may be due to the fact that they are all budget compact cameras that, with the exception of the Nikon D70s DSLR camera, are production models released three or more years earlier than the other cameras used in this test (see Table 5.1). The newer cameras in the set are likely to have benefited from advances in lens design enabling them to generate significantly sharper images than earlier camera models. Some new and mid-end cameras including E-PM2 and RX100 models also have built-in automatic enhancement algorithms and can produce very sharp images. The performance of the algorithm in detecting original images from these camera models are 100%.

The cameras that produce sharper images, however, cause slight drops of the performance in detecting recaptured images. The reason is that images from these cameras are sharper than the majority of images used in the training process. When these cameras were used for recapture, the sharpness quality and sharpening algorithm of the cameras sometimes might have produced recaptured images with sharp edges. Few of theses images are most likely to

be misclassified to the original capture group.

In Table 5.3 the results are listed in columns by the cameras used during the original capture. It can be observed that the algorithm has lowest recapture detection performances against the images first captured using the E-PM2 and RX100 models. In addition, our algorithm has lowest performance against the recapture combination using the RX100 and E-PM2 models for the first and second captures respectively.

5.2.3 Blurring Metric as a Feature for Classification

According to Section 4.4, the level of blurriness is one of the features that can classify single captured and recaptured images. We have used the width of line spread profiles $\bar{\lambda}$ as a metric to determine the degrees of blurriness of edges. From the feature coordinate in Figure 4.12, we note that while the error E_d is probably the more important feature, $\bar{\lambda}$ is also useful in that it has increased the discrimination ability of the classifier. The feature $\bar{\lambda}$, however, is not the only metric for blurriness measurement. In this section we test the algorithm when our proposed feature $\bar{\lambda}$ is replaced by a state-of-the-art blurriness metric. The purpose of the test is to compare the abilities to detect recaptured images of our proposed distinctive feature and one of the existing metrics. We consider using the CPBD metric [99] as it is one of the well-known no-reference image blur metrics [96, 97, 99] used to quantify the sharpness quality of digital images.

We note that the blurriness measurements obtained using our feature $\bar{\lambda}$ and CPBD metric are different by definition. Our feature $\bar{\lambda}$ measures the average width of line spread profiles in a group of sharpest edges extracted from a given image. The CPBD metric, in contrast, measures the probability of blur detection in a given image based on human perceptual models. According to Section 2.2.1.1, a sharper image would have a higher CPBD value. When the blurriness in the image increases, the resulting CPBD metric would decrease monotonically. Consequently we can expect that the CPBD metric measured from single captured images would have greater values when compared to the values measured from recaptured images.

In Table 5.5 the average of CPBD metrics computed from single captured images and their corresponding recaptured versions are compared. The CPBD values are calculated using a MATLAB software package made publicly available by Narvekar and Karam [99]. We can observe that the CPBD values of the images from all cameras drop after the images were recaptured. The average CPBD value for single captured images is 0.581 while the average CPBD for recaptured images is 0.409. To build a new SVM classifier using the CPBD metric as one of the features, we repeat the training process of Section 4.1. We compute CPBD

Table 5.5: CPBD metrics of images according to cameras used for original capture. The metrics computed before and after image recapture are compared.

Camera Model	CPBD Metric	
	Single Capture	After Recapture
Kodak V550 (Black)	0.526	0.382
Kodak V550 (Silver)	0.531	0.383
Kodak V610	0.500	0.359
Nikon D40	0.623	0.450
Nikon D70s	0.619	0.411
Canon EOS 600D	0.521	0.397
Sony RX100	0.618	0.443
Olympus E-PM2	0.611	0.403
Panasonic TZ7	0.676	0.418
Average	0.581	0.409

values, denoted by θ , for all the training images. A new pair of features $\{\theta, E_d\}$ for each image is constructed using the obtained θ and the already computed error E_d . After that all pairs of training features are used to train an SVM classifier in order to obtain the new hyperplane. The scatter plot of features with the resulting hyperplane (solid line) is shown in Figure 5.3.

We can observe that the hyperplane (solid line) increases the classification ability of the classifier when compared to the classifier that uses only the feature E_d (dotted line). However, the slope of the hyperplan suggests that the discrimination ability of the classifier is largely determined by the feature E_d . In addition, this hyperplane has lower degree of slope when compared to the hyperplane obtained from our proposed algorithm using $\bar{\lambda}$ feature. We therefore argue that the feature $\bar{\lambda}$ used in our proposed algorithm has higher ability to extract distinctive characteristics between line spread profiles from single captured and recaptured images.

The hyperplane obtained from the training process is then used for testing purpose. We repeat the performance evaluation of the modified algorithm following the similar process and parameter setting previously stated in Section 5.2.1. The classification performance of the algorithm when it is tested with single captured images is presented in Table 5.6. The results show that the modified algorithm using the CPBD metric is better than our proposed algorithm with a success rate of 95.56 % against 94.44 % achieved using the spread width $\bar{\lambda}$.

The results in Table 5.7, nevertheless, suggest that our proposed algorithm using $\bar{\lambda}$ has higher classification performance in detecting recaptured images. The algorithm using $\bar{\lambda}$ feature has higher performance for all 8 groups of images recaptured from 8 different cameras.

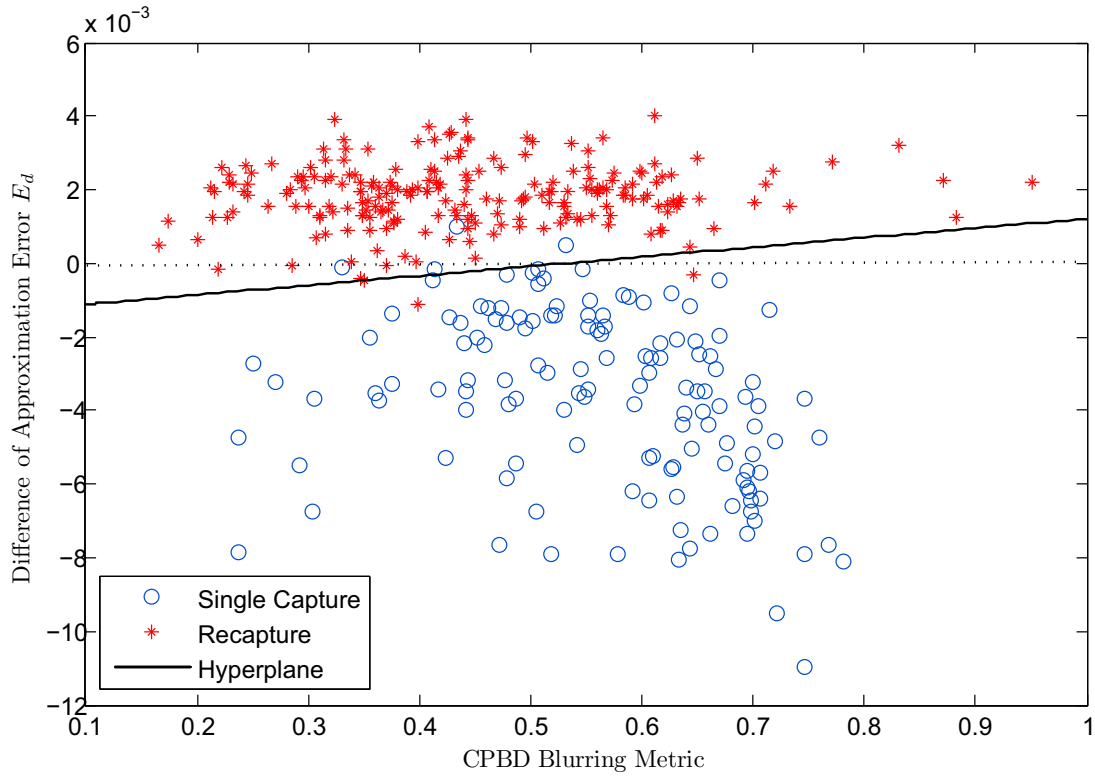


Figure 5.3: A plot of the distributions of features extracted from training images with CPBD metric (θ) shown on the horizontal axis and the difference of representation error (E_d) on the vertical axis. The hyperplane for recapture classification was obtained using SVM training and is defined as the line that separate the features from the recaptured (star) and single captured (circle) images with minimum classification error. A query image is classified based on the coordinate location of the feature pair, $\{\theta, E_d\}$ determined from the image. The dotted line ($E_d = 0$) shows the classification threshold when only the feature E_d is used for training.

Table 5.6: The performance of the algorithm in detecting single captured images according to the feature used for blurriness measurement.

Camera Model	Performance (%)	
	Spread Width $\bar{\lambda}$	CPBD Metric
Kodak V550 (Black)	90	95
Kodak V550 (Silver)	75	95
Kodak V610	90	90
Nikon D40	100	100
Nikon D70s	95	90
Canon EOS 600D	100	100
Sony RX100	100	100
Olympus E-PM2	100	90
Panasonic TZ7	100	100
Average	94.44	95.56

Table 5.7: The performance of the algorithm in detecting recaptured images according to the feature used for blurriness measurement.

Camera Model	Performance (%)	
	Spread Width $\bar{\lambda}$	CPBD Metric
Nikon D3200	100	97.22
Nikon D70s	99.44	98.33
Canon EOS 600D	98.89	98.33
Canon EOS 60D	99.44	98.89
Sony RX100	100	98.89
Olympus E-PM2	96.11	62.22
Panasonic TZ7	100	98.33
Panasonic TZ10	99.44	97.78
Average	99.17	93.75

The overall performance of our algorithm is 99.03 % which is higher than the performance of 93.75 %. achieved when using the CPBD metric.

Our proposed blurring feature $\bar{\lambda}$ is therefore overall superior. This is because most blurring metrics including the CPBD metric were introduced to measure the global level of blurriness in images, since these metrics are used normally for quality assessment of digital images rather than forensics. It is therefore necessary for these metrics to highlight global degrees of blurriness as they are based on the measurement of blurriness levels perceived by the human visual system. These metrics are often influenced by content of images and the depth of field (DOF) of cameras. For examples, when edges with various degrees of sharpness are present in a captured scene, the value of blurring metrics would be determined by the average sharpness of all edges in the image. In addition, when the image is taken using a narrow depth of field, large areas of images in off-focused area are blurred. These factors will increase the value of blurriness making these metrics less able to discriminate between single captured and recaptured images.

This is contrast with the method based on line spread profile width $\bar{\lambda}$. In Section 4.2, the parameters $\bar{\lambda}$ is computed using only a group of sharpest blocks in images and the majority of blurred edges are discarded. In this way the obtained width value would not be affected by the blurriness in off-focused areas. This makes the $\bar{\lambda}$ more suitable for our recapture detection application.

5.2.4 Universality of the Algorithm

In the previous section we assumed that the edge profiles of a wide range of image capture devices and recapture chains are known. In practice, the range of devices and image chains that are characterised may be limited. In this section, the performance of the algorithm was tested when applied to images acquired from unknown capture devices. It is anticipated that, in practice, the edge profiles of single and recaptured images are fundamentally different, and that recapture detection is possible even though a small range of devices and chains representative of the population have been characterised. An experiment was devised in which the algorithm was trained with single and recaptured images taken with a different set of cameras to that used to capture the images used for testing. Table 5.8 illustrates the configuration of the training and testing image sets. Note that the cameras used for single capture and recapture differ for the training and testing sets. A total of 60 single captured images and 108 recaptured images were used for training. For testing purposes, 100 single captured images and 720 recaptured images were used.

Table 5.8: Group configuration for images used to test the universality of the algorithm

Image Set	Single Capture Camera	Recapture Camera
Training	Nikon D40 Canon EOS 600D Kodak V550 Black Case Kodak V610	Nikon D3200 Canon EOS 600D Olympus - E-PM2 Panasonic TZ10
Testing	Nikon D70s Olympus - E-PM2 SONY RX100 KodakV550 Silver Case Panasonic TZ7	NikonD70s Canon EOS 60D SONY RX100 Panasonic TZ7

The new hyperplane resulting from the training of the K-SVD dictionary and SVM, was used for performance evaluation. The recapture detection results are shown in Table 5.9. A recapture detection rate, or true positive (TP) rate, of 99.31% was obtained over all cameras. For the detection of single captured images, a true negative (TN) rate of 97% was obtained. We can conclude, therefore, that the overall performance of the algorithm has been maintained despite using images from different cameras for training and testing.

The experimental work conducted in this section has shown that the proposed algorithm is robust to a wide range of images and can be applied successfully to a range of capture devices. Furthermore, it is able to cope with query images from unknown sources and image chains.

Table 5.9: Recapture detection performance of the algorithm for the universality testing

Camera Model	No. of images	Classification Results		Performance (%)
		Single	Recapture	
Single Capture Database				
See Table 5.8	100	97	3	97.00
Recapture Database				
Nikon D70s	180	2	178	98.89
Canon EOS 60D	180	1	179	99.44
Sony RX100	180	0	180	100
Panasonic TZ7	180	2	178	98.89
Average				99.31

5.3 Performance Comparison

Next we compare the performance of our proposed algorithm with state-of-the-art techniques for detecting recaptured images in our alias-free recapture dataset.

5.3.1 Recaptured Detection using LBP, MSWS, and Colour Features

Our comparison is primarily against the method presented by Cao and Kot [1] because it is one of the first and most referenced approaches for detecting recaptured images from LCD monitors. In addition, this chosen method was previously tested on a set of finely recaptured images thus it is most comparable to our method. In this method, classification relies on the combination of 3 types of features as follows:

Local Binary Feature (LBP)

The Local Binary Pattern (LBP) [118] is widely used for texture analysis. The LBP feature and its statistics are used in [1, 84, 81] for recapture detection because the authors found that the LBP features extracted from recaptured and single captured images are markedly different. Such differences are characterised by texture anomalies introduced to recaptured images during a recapture process. In our experiments, the images were first converted to grayscale before we extracted the Rotation Invariant Uniform LBP $LBP_{8,1}^{riu2}$, $LBP_{16,2}^{riu2}$, and $LBP_{24,3}^{riu2}$ from each image using the LBP MATLAB software package provided at [119]. The resulting features extracted from individual image were three normalised histograms with 10, 18, and 26 histogram bins respectively. The histograms were then concatenated to form a feature row of 54 elements.

Multi-Scale Wavelet Statistics (MSWS)

In [1], the authors suggest that once images are recaptured, the loss of image details is inevitable. The detail loss is caused by mismatch between the resolutions of the LCD screen and the image sensor. In the paper the Multi-Scale Wavelet Statistics (MSWS) was used to measure the detail loss that marks the differences between recaptured and original images.

First, three levels of 2D wavelet decomposition is applied separately to R, G, and B channels of a given image using a standard Haar filter. Only detail coefficients HL, LH, HH of each decomposition level were used. This results in $3 \times 3 \times 3 = 27$ sets of detail coefficients for all colour channels and all levels of decomposition. Next, mean and variance of each set of detail coefficients were computed. We therefore obtain 54 features that represent Multi-Scale Wavelet Statistics of the image.

Colour Features

RGB colour features including average pixel values, colour channel correlations, neighbour distribution centers of mass, and pairs energy ratios were initially used in [120] for blind source camera identification. The method has high performance because tones of images are often unique to camera models. In [1], these features are used with colour moments computed from HSV colour space [121] for recapture detection purpose. In the experiment, the first 4 colour features were extracted separately from the RGB colour channels thereby resulting in 12 features. The additional 9 features including mean, standard deviation and skewness of colour components in the HSV space were then computed. The total number of dimensions for colour features is 21.

Since the software presented in [1] is not publicly available, we implemented the algorithm following the instructions of the authors. We next ran the algorithms to extract the features from the same training set we used in our technique. A total of 129 features were obtained for each given image and labelled as a recaptured or original captured according to the class of the images. The extracted features were then used to train a support vector machine (SVM) classifier using the LIBSVM Tools following the technical guideline in [122]. This results in a hyperplane which can classify a given image to a recaptured or original captured group based on its extracted features. The classifier was then tested with the testing set.

5.3.2 Recaptured Detection using Higher-order Wavelet Statistics

In addition to the technique presented in [1], recapture detection algorithm based on features from Higher-order Wavelet Statistics was frequently used as a benchmark to compare performance in detecting recaptured images. These features were first used in forensic problems to discriminate between photographic and computer graphic images [123, 124]. They were used in [1, 84] to compare recapture detection performances because the features can extract unique fine texture patterns that are present in original images. These patterns, in contrast, are likely to be absent in computer graphic or recaptured images. The feature extraction involves image decomposition using separable quadrature mirror filters (QMFs) [125, 126]. The resulting vertical, horizontal, and diagonal subbands are used to compute mean, variance, skewness, kurtosis, and error statistics.

We used the software library provided by the authors [124] to extract and obtain 216 features for a given RGB image. We then repeated all training process using LIBSVM Tools and obtained the algorithm for recapture detection.

5.3.3 Comparison Results

The performances of the methods [1] and [123], and our proposed techniques are compared using the same dataset we previously used in Section 5.2.2. The results are presented in Table 5.10. We compare the performances in terms of true positive (TP) rate, false negative (FN) rate, true negative (TN) rate, false positive (FP) rate, and the number of features used when the algorithms were tested with our dataset. According to the results, our propose method outperforms both the existing techniques in classifying original and recaptured images. Our technique has higher true positive (TP) rate at 99.03 % compared to the TP rates of the methods [1] and [123] at 92.02% and 90.04% respectively.

Our proposed method also has the highest performance in detecting original captured images from the original capture set with the TN rate at 94.44 %. This rate is higher than the TN rates of the methods [1] and [123] at 84.44% and 87.22% respectively.

In addition to the higher performance, our method uses only 2 features extracted from each given images when compared to 129 and 216 features required in the methods [1] and [123] respectively. Our method, however, requires block-based edge extraction and dictionary learning processes to obtain a smaller number of features.

The main reason that our proposed method has higher classification performance is that the

Table 5.10: The comparison of performances of the algorithms in classifying original and recaptured images.

Method	Number of Features	Performance (%)			
		TP	FN	TN	FN
MSWS+LBP+Colour Features [1]	129	92.02	7.98	84.44	15.56
Higher-order Wavelet Statistics [123]	216	90.04	9.96	87.22	12.78
Proposed Method	2	99.03	0.97	94.44	5.56

features we use can better extract distinctive characteristics between original and recaptured images in our dataset. In particular the aim of our method is to detect images that were finely recaptured to create near-duplicate spoofs of original images. The database we used hence was created from the settings that allow to obtain alias-free recaptured images. We also tried to eliminate the tiny patterns of surface texture of the LCD screen such that the recaptured images are not easily detected by human inspection. Our base line is that the algorithm must be able to detect recaptured images when these traces are minimally visible. These patterns and textures, however, were useful traces for the methods [1] and [123] but are not used in our technique. When these patterns are not present in the test images, this did not affect the performance of our algorithm but decreased the classification abilities of the existing methods, in particular the method using LBP features.

The other important reason is that our features were extracted locally. They are the selected edge profile features that are most likely to mark the differences between original and recaptured images. Edge profiles outside the regions of interest, for examples off-focused areas and the blocks without edges, were discarded. This is contrast to the methods to compute texture features in [1] and [123] where the statistics of wavelet coefficients were computed globally. These wavelet statistics generally have high discriminative power when larger areas of images are spanned with high details. On the other hand, when images contain large areas of plain space and only small regions with fine textures, the loss of fine textures in recaptured images would not significantly change the statistical values of wavelet coefficients because they were averaged across the images. The wavelet statistics extracted globally therefore have less sensitivities and lower discriminative power compared to our features for these types of images.

5.4 Summary

In this section we have presented experiments to evaluate performance of our proposed algorithm in detecting recaptured images. We first described how to create a database for alias-free recaptured images. These images were used to test our algorithm. The results have shown that the proposed algorithm is robust to a wide range of images and can be applied successfully to a wide range of capture devices. Furthermore, it is able to cope with query images from unknown sources and image chains. Finally we have tested the performances of well-known methods using our dataset. The results have shown that our algorithm has better performances in classifying original captured and recaptured images when compared to those benchmark methods.

Chapter 6

Conclusions

6.1 Thesis Summary

We have presented in this thesis two different forensic techniques for the reverse engineering of image acquisition chains. Our techniques are based on blurriness footprints introduced to digital images during acquisition processes. In the first approach we have proposed a theoretical framework to address the problem using the theory of sampling signals with finite rate of innovation. We simplify the problem by describing chains of processing operators using a 1-D signal model and focussing on the case that signals are reacquired at most once. In particular conditions, the proposed technique allows to retrieve the acquisition history of unknown discrete signals. We have been able to provide a precise answer whether the given signal was recaptured before or obtained directly through the original acquisition. In reacquisition case, our technique makes it possible to estimate the parameters of the chain the signal has gone through including the sampling period, and the interpolation function.

The main requirement necessary to this framework is that the characteristics of the most recent acquisition device must be known in advance. For example in an image system, line spread functions of the devices used for last capture must be known. The sampling kernels assumed in that context are the kernels that can reproduce polynomials, such as B-spline functions. Using the FRI sampling theory we can estimate the number of K discontinuities of unknown continuous signals from their corresponding samples. Our reacquisition detection algorithm is based on the fact that the original signal and reconstructed signals are described differently in terms of number of discontinuities in the FRI signals. Specifically the original signal is assumed to be a sharp straight edge which can be modelled using the step function. The function is an FRI signal with a number of discontinuity $K = 1$. In contrast, we have found that particular types of signal reconstruction, e.g. polynomial interpolation, produce

piecewise polynomial functions which are FRI signals with $K > 1$.

The FRI sampling theory also allows us to achieve perfect reconstruction of the continuous signals. This reconstruction technique permits to reverse engineer the acquisition process and to retrieve important parameters of the chain the signals have gone through. Finally we have proven two sufficient conditions that make the reverse engineering technique feasible. The sufficient conditions impose the constraints that the second device must have better qualities when compared to the first acquisition device. That is, first the second sampling kernel must be capable of reproducing polynomial of sufficiently high degree. This order is determined by the length of the previous sampling kernel, sampling rate of the previous acquisition, and the order of polynomial interpolation. Second, the reacquisition must be performed at sufficiently high sampling rate when compared to the sampling rate of the first acquisition.

Our second approach has focused on detecting recaptured images. We have explained how to verify whether a given query image was genuine or a recaptured image. Our forensic analysis is based on unique characteristics of blurriness patterns introduced to edges during acquisition processes. We have described the blurring patterns using line spread profile features extracted from edges present in digital images. From our studies, line spread profiles obtained from single captured images are markedly different from the profiles from recapture images. We have extracted such differences using two features, namely width of line spread profiles $\bar{\lambda}$ and the error E_d . The first feature is the measurement of the level of blurriness introduced to images. The image with higher degrees of blurriness would result in larger value of line spread profile width $\bar{\lambda}$. Our experimental data has suggested that image recapture increases the degrees of blur to the images. We have observed that the increased levels of blurriness are significant and markedly different from the levels measured from single captured images. Such differences have been used for classification between original captures and recaptured images.

The second feature E_d corresponds to the shapes of line spread profiles. We have found that the shapes of line spread profiles from single captured and recaptured images are different. We trained two different overcomplete dictionaries to learn the shapes of line spread profiles extracted from each group of images. The error E_d is obtained from the difference between sparse approximation errors of the profile based on the two overcomplete dictionaries. The value of E_d is determined by the similarity between the profiles from the query image and the approximated profiles that each dictionary generates. The value is likely to be negative for a single captured image and positive for a recaptured image. Next we trained an SMV classifier using pairs of features $\{\bar{\lambda}, E_d\}$ extracted from separated sets of labelled single captured and

recapture images. The result of training was the hyperplane which was used to classify a pair features from a query image. The trained classifier was then used for recapture detection.

We have created a dataset for experiments. The dataset comprises 315 original images captured from 9 different cameras and 2520 recaptured images obtained from 72 different recapture combinations. The recapture was setup to remove aliasing artefacts which are one of the visible traces of recapture. The images were divided for training and testing purposes. The experimental results have shown that our proposed algorithm has high performance in detecting recaptured images. The detection precision of the algorithm is 99.17% against the dataset of 1440 recaptured images. The precision of the algorithm is 94.44% when it was tested with 180 single captured images.

We also conducted three additional experiments. The first experiment was to compare the classification abilities of the algorithm when using our feature $\bar{\lambda}$ and the CPBD blurring metric. We have proved that our proposed feature is more suitable for the image recapture detection algorithm. In the second experiment, we tested the universality of the proposed algorithm. The results have suggested that our algorithm could maintain high classification precision although it was tested with images from unknown sources and image chains. Finally we compared the performances of our algorithm with two well-known benchmark methods using our dataset. The results have shown that our method outperforms both algorithms in detecting recaptured and original captured images when using a smaller number of features. We have suggested that our algorithm has higher performances because our method has been able better to extract local distinctive features that mark the difference between original and recaptured images. In particular we gained better performances when artefacts from aliasing and tiny cells of the LCD screen were not present in the images, and also when images contained small areas with fine details.

6.2 Future Research

To conclude this thesis we discuss some open questions and possible directions for future research. We split this discussion into two parts: the extensions for theoretical framework for reverse engineering of signal processing chains and the improvement of the image recapture algorithm.

Theoretical framework for reverse engineering of signal processing chains

The theoretical framework proposed in this thesis is based on the most simple chains of

operators. In particular, we have assumed that the signal is reacquired at most once. It would be interesting to consider the problem of more complex chain structures, for example a longer chain of multiple A/D and D/A conversions. A possible future research question might address a method to retrieve the number of acquisitions and reconstructions the signals have gone through.

In addition, for the focus of the research we have formulated the problem using only acquisition and reconstruction operations in this thesis. Chains of processing in practice, however, comprise more than two types of operators. The extended chains of acquisition might also include other operators, such as quantisation and transform coders. The proposed framework might be extended and use the analytical technique presented in [127, 128]. This new problem setup can lead to the development of forensic techniques for detecting or identifying quantisers and coders in complex chains of operators. The future research might also study the chains with combinations of sampling operations in the continuous and discrete domain. The understanding in this type of chains would allow to detect whether images were digitally manipulated with resizing or rotating techniques before recapture.

The extension can also consider the analysis of the impact of noise on the proposed algorithms similarly to the work of Blu *et al.* [112]. In particular, deriving performance bounds, such as the Cramer-Rao bounds, would provide much insights as to how the noise affects the reverse engineering schemes.

Other possible future research might include the extension to 2-D signal models and different classes of sampling kernels.

The improvement of image recapture algorithm

In Chapter 5 we have shown that the proposed method has high performance in classifying original captures and recaptured images. The algorithm, however, has failed to produce correct results for some few cases. In particular, when no sharp edges are present in the scenes, some original images are classified as recaptured because the blurriness pattern measured from those images deviates from the typical training patterns. On the other hand, recaptured images can also be misclassified in some cases. For example, if the monitor or second devices have very high quality, the resulting recaptured images can be very sharp. The edges extracted from these images could be sharp enough and very similar to edges obtained from original captured images, making the classification more difficult. In addition, image sharpening algorithm can also spoof our algorithm if recaptured images are enhanced and have sharper edges.

One way to improve the algorithm is to train the classifier with larger and richer kinds of images. The major reason, however, is that the algorithm is heavily dependent on the feature from acquisition blurriness. The future research might lie in combining the features used in the proposed algorithm with other types of features including noise, saturation, contrast, and the loss of high frequency details. We believe that the improvement using these features will increase the robustness of the algorithm.

The algorithm can also be improved using side information from relevant database. First the query image might be used to retrieve a near duplicate image from online or available database. If the similar image is found, it is likely that one of the images is a duplicate or recaptured version. Then we can use the techniques proposed in this thesis to extract features that can distinguish between two versions of images. If the query image has higher degree of blurriness or better matched with the dictionary of recaptured images, it is anticipated that the image is a recaptured version.

The extended problem also includes the detection of other operators, such as compression, quantisation, and image enhancement applied to images before recapture. We are also interested in understand how footprints left from those operations are transformed or survive through image recapture.

Bibliography

- [1] H. Cao and A. C. Kot, “Identification of recaptured photographs on LCD screens,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2010, pp. 1790–1793.
- [2] Izitru Trusted Photos, Fourandsix Technologies Inc., <http://www.izitru.com>, accessed 10 August 2014.
- [3] M. K. Johnson and H. Farid, “Exposing digital forgeries through chromatic aberration,” in *ACM Multimedia and Security Workshop (MM&Sec)*, 2006, pp. 48–55.
- [4] T. V. Lanh, S. Emmanuel, and M. S. Kankanhalli, “Identifying source cell phone using chromatic aberration,” in *IEEE International Conference on Multimedia and Expo (ICME)*, 2007, pp. 883–886.
- [5] K. S. Choi, E. Y. Lam, and K. K. Y. Wong, “Automatic source camera identification using the intrinsic lens radial distortion,” in *Optics Express*, vol. 14, 2006, pp. 11 551–11 565.
- [6] J. Lukás, J. Fridrich, and M. Goljan, “Digital camera identification from sensor pattern noise,” in *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 2, 2006, pp. 205–214.
- [7] M. Chen, J. Fridrich, M. Goljan, and J. Lukás, “Determining image origin and integrity using sensor noise,” in *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 1, 2008, pp. 74–90.
- [8] Y. Sutcu, S. Bayram, H. Sencar, and N. Memon, “Improvements on sensor noise based source camera identification,” in *IEEE International Conference on Multimedia and Expo (ICME)*, 2007, pp. 24–27.
- [9] B.-B. Liu, Y. Hu, and H.-K. Lee, “Source camera identification from significant noise residual regions,” in *IEEE International Conference on Image Processing (ICIP)*, 2010, pp. 1749–1752.

- [10] H. Gou, A. Swaminathan, and M. Wu, "Noise features for image tampering detection and steganalysis." in *International Conference on Image Processing (ICIP)*, 2007, pp. 97–100.
- [11] N. Khanna, A. K. Mikkilineni, G. Chiu, J. P. Allebach, and E. J. Delp, "Forensic classification of imaging sensor types," in *SPIE Conference on Security, Stego., and Wmk of Multimedia Contents IX*, vol. 6505, 2007, pp. 650–655.
- [12] C. McKay, A. Swaminathan, H. Gou, and M. Wu, "Image acquisition forensics: Forensic analysis to identify imaging source," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2008, pp. 1657–1660.
- [13] J. Fan, H. Cao, and A. C. Kot, "Estimating EXIF parameters based on noise features for image manipulation detection," in *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 4, 2013, pp. 608–618.
- [14] S. Gao, G. Xu, and R.-M. Hu, "Camera model identification based on the characteristic of CFA and interpolation," in *Proceedings of the 10th International Conference on Digital-Forensics and Watermarking*, 2012, pp. 268–280.
- [15] Z. Lin, R. Wang, X. Tang, and H.-Y. Shum, "Detecting doctored images using camera response normality and consistency," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005, pp. 1087–1092.
- [16] Y.-F. Hsu and S.-F. Chang, "Image splicing detection using camera response function consistency and automatic segmentation." in *IEEE International Conference on Multimedia and Expo (ICME)*, 2007, pp. 28–31.
- [17] A. C. Popescu and H. Farid, "Exposing digital forgeries in color filter array interpolated images," in *IEEE Transactions on Signal Processing*, vol. 53, 2005, pp. 3948–3959.
- [18] S. Bayram, H. T. Sencar, N. D. Memon, and I. Avcibas, "Source camera identification based on CFA interpolation," in *IEEE International Conference on Image Processing (ICIP)*, 2005, pp. 69–72.
- [19] P. Ferrara, T. Bianchi, A. D. Rosa, and A. Piva, "Image forgery localization via fine-grained analysis of CFA artifacts," in *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 5, 2012, pp. 1566–1577.

-
- [20] H. Cao and A. C. Kot, "Accurate detection of demosaicing regularity for digital image forensics," in *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 4, 2009, pp. 899–910.
- [21] C.-H. Choi, H.-Y. Lee, and H.-K. Lee, "Estimation of color modification in digital images by CFA pattern change," in *Forensic Science International*, vol. 226, no. 1-3, 2013, pp. 94–105.
- [22] S. Gao, G. Xu, and R. Hu, "Camera model identification based on the characteristic of CFA and interpolation," in *International Workshop on Digital Forensics and Watermarking*, no. 7128, 2011.
- [23] W.-H. Chuang and M. Wu, "Robustness of color interpolation identification against anti-forensic operations," in *International Workshop on Information Hiding*, 2012, pp. 16–30.
- [24] Z. Fan and R. L. d. Queiroz, "Maximum likelihood estimation of JPEG quantization table in the identification of bitmap compression history," in *International Conference on Image Processing (ICIP)*, 2000, pp. 948–951.
- [25] Z. Fan and R. de Queiroz, "Identification of bitmap compression history: JPEG detection and quantizer estimation," in *IEEE Transactions on Image Processing*, vol. 12, no. 2, 2003, pp. 230–235.
- [26] G. Qadir, X. Zhao, and A. T. S. Ho, "Estimating JPEG2000 compression for image forensics using Benford's law," in *Proc. SPIE 7723, Optics, Photonics, and Digital Technologies for Multimedia Applications*, vol. 7723, 2010.
- [27] F. Benford, "The law of anomalous numbers," in *Proceedings of the American Philosophical Society*, 1938, pp. 551–572.
- [28] J. Lukas and J. Fridrich, "Estimation of primary quantization matrix in double compressed JPEG images," in *Digital Forensic Research Workshop*, Aug. 2003.
- [29] V. Holub, J. Fridrich, and T. Denemark, "Universal distortion function for steganography in an arbitrary domain," in *EURASIP Journal on Information Security*, 2014.
- [30] A. C. Popescu and H. Farid, "Statistical tools for digital forensics," in *Information Hiding*, ser. Lecture Notes in Computer Science, vol. 3200. Springer, 2004, pp. 128–147.

- [31] J. He, Z. Lin, L. Wang, and X. Tang, “Detecting doctored JPEG images via DCT coefficient analysis,” in *European Conference on Computer Vision*, 2006, pp. 423–435.
- [32] F. Huang, J. Huang, and Y. Q. Shi, “Detecting double JPEG compression with the same quantization matrix,” in *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 4, 2010, pp. 848–856.
- [33] Y.-L. Chen and C.-T. Hsu, “Detecting recompression of JPEG images via periodicity analysis of compression artifacts for tampering detection,” in *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 2, 2011, pp. 396–406.
- [34] T. Bianchi and A. Piva, “Reverse engineering of double JPEG compression in the presence of image resizing,” in *Workshop on Information Forensics and Security*, 2012, pp. 127–132.
- [35] ———, “Image forgery localization via block-grained analysis of JPEG artifacts,” in *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, 2012, pp. 1003–1017.
- [36] S. Milani, M. Tagliasacchi, and S. Tubaro, “Discriminating multiple JPEG compression using first digit features,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2012, pp. 2253–2256.
- [37] S. Ye, Q. Sun, and E.-C. Chang, “Detecting digital image forgeries by measuring inconsistencies of blocking artifact,” in *IEEE International Conference on Multimedia and Expo (ICME)*, 2007, pp. 12–15.
- [38] W. Li, Y. Yuan, and N. Yu, “Passive detection of doctored JPEG image via block artifact grid extraction,” in *ACM Signal Process Journal*, vol. 89, no. 9, 2009, pp. 1821–1829.
- [39] M. Johnson and H. Farid, “Exposing digital forgeries by detecting inconsistencies in lighting,” in *ACM Multimedia and Security Workshop*, 2005, pp. 1–10.
- [40] M. K. Johnson and H. Farid, “Exposing digital forgeries in complex lighting environments,” in *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 2, 2007, pp. 450–461.
- [41] E. Kee and H. Farid, “Exposing digital forgeries from 3-D lighting environments,” in *IEEE International Workshop on Information Forensics and Security*, 2010, pp. 1–6.

-
- [42] M. K. Johnson and H. Farid, "Exposing digital forgeries through specular highlights on the eye," in *International Workshop on Information Hiding*, 2007, pp. 311–325.
- [43] C. Riess and E. Angelopoulou, "Scene illumination as an indicator of image manipulation," in *International Workshop on Information Hiding*, 2010, pp. 66–80.
- [44] W. Fan, K. Wang, F. Cayre, and Z. Xiong, "3D lighting-based image forgery detection using shape-from-shading," in *Proceedings of the European Signal Processing Conference (EUSIPCO)*, 2012, pp. 1777–1781.
- [45] Q. Liu, X. Cao, C. Deng, and X. Guo, "Identifying image composites through shadow matte consistency," in *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, 2011, pp. 1111–1122.
- [46] E. Kee, J. O'Brien, and H. Farid, "Exposing photo manipulation with inconsistent shadows," in *ACM Transactions on Graphics*, vol. 32, no. 4, 2013, pp. 28:1–28:12.
- [47] V. Conotter, G. Boato, and H. Farid, "Detecting photo manipulation on signs and billboards," in *IEEE International Conference on Image Processing (ICIP)*, 2010, pp. 1741–1744.
- [48] P. Kakar, N. Sudha, and W. Ser, "Exposing digital image forgeries by detecting discrepancies in motion blur," in *IEEE Transactions on Multimedia*, vol. 13, no. 3, 2011, pp. 443–452.
- [49] J. Fridrich, D. Soukal, and J. Lukas, "Detection of copy move forgery in digital images," in *Digital Forensic Research Workshop*, 2003.
- [50] A. Popescu and H. Farid, "Exposing digital forgeries by detecting duplicated image regions," Department of Computer Science, Dartmouth College, Tech. Rep. TR2004-515, 2004.
- [51] G. Li, Q. Wu, D. Tu, and S. Sun, "A sorted neighborhood approach for detecting duplicated regions in image forgeries based on DWT and SVD," in *IEEE International Conference on Multimedia and Expo (ICME)*, 2007, pp. 1750–1753.
- [52] X. Pan and S. Lyu, "Region duplication detection using image feature matching," in *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 4, 2010, pp. 857–867.

- [53] W. Li and N. Yu, "Rotation robust detection of copy-move forgery," in *IEEE International Conference on Image Processing (ICIP)*, 2010, pp. 2113–2116.
- [54] S. Bayram, H. T. Sencar, and N. Memon, "An efficient and robust method for detecting copy-move forgery," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2009, pp. 1053–1056.
- [55] R. Davarzani, K. Yaghmaie, S. Mozaffari, and M. Tapak, "Copy-move forgery detection using multiresolution local binary patterns," in *Forensic Science International*, vol. 231, no. 1, 2013, pp. 61–72.
- [56] A. C. Popescu and H. Farid, "Exposing digital forgeries by detecting traces of resampling," in *IEEE Transactions on Signal Processing*, vol. 53, 2005, pp. 758–767.
- [57] M. Kirchner, "Fast and reliable resampling detection by spectral analysis of fixed linear predictor residue," in *ACM Multimedia and Security Workshop*, 2008, pp. 11–20.
- [58] A. C. Gallagher, "Detection of linear and cubic interpolation in JPEG compressed images," in *Proceedings of the 2nd Canadian Conference on Computer and Robot Vision*, 2005, pp. 65–72.
- [59] B. Mahdian and S. Saic, "Blind authentication using periodic properties of interpolation," in *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 3, 2008, pp. 529–538.
- [60] M. C. Stamm and K. J. R. Liu, "Blind forensics of contrast enhancement in digital images," in *IEEE International Conference on Image Processing (ICIP)*, 2008, pp. 3112–3115.
- [61] —, "Forensic detection of image manipulation using statistical intrinsic fingerprints," in *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 3, 2010, pp. 492–506.
- [62] A. Bovik, "Streaking in median filtered images," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 35, 1987, pp. 493 – 503.
- [63] M. Kirchner and J. Fridrich, "On detection of median filtering in digital images," in *SPIE Conference on Media Forensics and Security*, 2010.

-
- [64] G. Cao, Y. Zhao, R. Ni, L. Yu, and H. Tian, "Forensic detection of median filtering in digital images," in *IEEE International Conference on Multimedia and Expo (ICME)*, 2010, pp. 89–94.
- [65] T. Pevný, P. Bas, and J. Fridrich, "Steganalysis by subtractive pixel adjacency matrix," in *Proceedings of the 11th ACM Workshop on Multimedia and Security*, 2009, pp. 75–84.
- [66] H.-D. Yuan, "Blind forensics of median filtering in digital images," in *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 4, 2011, pp. 1335–1345.
- [67] C. Chen, J. Ni, R. Huang, and J. Huang, "Blind median filtering detection using statistics in difference domain," in *International Workshop on Information Hiding*, 2012, pp. 1–15.
- [68] R. Palom, "Optical antialiasing filters," in *Single-Sensor Imaging: Methods and Applications for Digital Cameras (Image Processing Series)*, 1st ed., R. Lukac, Ed. CRC Press, September 2008, ch. 4, pp. 105–135.
- [69] J. Adams, K. Parulski, and K. Spaulding, "Color processing in digital cameras," in *IEEE Micro*, vol. 18, no. 6, 1998, pp. 20–30.
- [70] H. Muammar and P. L. Dragotti, "An investigation into aliasing in images recaptured from an LCD monitor using a digital camera," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2242–2246.
- [71] T. Koyama, "Optics in digital still cameras," in *Image Sensors and Signal Processing for Digital Still Cameras*, J. Nakamura, Ed. CRC Press, 2006, pp. 21–51.
- [72] A. da Silva Pinto, H. Pedrini, W. Schwartz, and A. Rocha, "Video-based face spoofing detection through visual rhythm analysis," in *Conference on Graphics, Patterns and Images (SIBGRAPI)*, 2012, pp. 221–228.
- [73] M. Goljan, J. Fridrich, and T. Filler, "Large scale test of sensor fingerprint camera identification," in *Proceedings of the SPIE - The International Society for Optical Engineering*, vol. 7254, USA, 2009, pp. 72 540I.1–72 540I.12.
- [74] M. Goljan, J. Fridrich, and J. Lukas, "Camera identification from printed images," in *Symposium on Electronic Imaging Science, Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*, 2008.

- [75] *IEC 61966-2-1:1999 Multimedia systems and equipment - Colour measurement and management - Part 2-1: Colour management - Default RGB colour space - sRGB*, International Electrotechnical Commission (IEC), 1999.
- [76] H. Yu, T.-T. Ng, and Q. Sun, "Recaptured photo detection using specularly distribution," in *IEEE International Conference on Image Processing (ICIP)*, 2008, pp. 3140–3143.
- [77] J. Yin and Y. Fang, "Markov-based image forensics for photographic copying from printed picture," in *ACM International Conference on Multimedia*, 2012, pp. 1113 – 1116.
- [78] J. Bai, T.-T. Ng, X. Gao, and Y.-Q. Shi, "Is physics-based liveness detection truly possible with a single image?" in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2010, pp. 3425–3428.
- [79] X. Gao, T.-T. Ng, B. Qiu, and S.-F. Chang, "Single-view recaptured image detection based on physics-based features," in *IEEE International Conference on Multimedia and Expo (ICME)*, 2010, pp. 1469–1474.
- [80] W. Jiang, A. T. S. Ho, H. Treharne, and Y. Q. Shi, "A novel multi-size block Benford's law scheme for printer identification," in *Advances in Multimedia Information Processing - PCM 2010*, vol. 6297, 2010, pp. 643–652.
- [81] N. Kose and J.-L. Dugelay, "Classification of captured and recaptured images to detect photograph spoofing," in *International Conference on Informatics, Electronics Vision (ICIEV), 2012*, 2012, pp. 1027–1032.
- [82] J. Yin and Y. Fang, "Digital image forensics for photographic copying," in *Proceedings of the SPIE - The International Society for Optical Engineering*, vol. 8303, 2012, p. 83030F (7 pp.).
- [83] B. Li, Y. Shi, and J. Huang, "Detecting doubly compressed JPEG images by using mode based first digit features," in *IEEE 10th Workshop on Multimedia Signal Processing*, oct. 2008, pp. 730 –735.
- [84] Y. Ke, Q. Shan, F. Qin, and W. Min, "Image recapture detection using multiple features," in *International Journal of Multimedia and Ubiquitous Engineering*, vol. 8, no. 5, 2013, pp. 71–82.

-
- [85] X. Gao, B. Qiu, J. Shen, T.-T. Ng, and Y. Q. Shi, "A smart phone image database for single image recapture detection," in *International Workshop on Digital Watermarking (IWDW)*, 2011, pp. 90–104.
- [86] B. Wandell, *Foundations of vision*. Sinauer Associates, 1995.
- [87] A. Cunningham and A. Fenster, "A method for modulation transfer function determination from edge profile with correction for finite element differentiation," in *Medical Physics*, vol. 14, no. 4, 1987, pp. 533–537.
- [88] ISO, "ISO 12233:2014 photography – electronic still picture imaging – resolution and spatial frequency responses." ISO, Geneva, Switzerland, 2014.
- [89] P. Burns, "Slant edge analysis tool SFRMAT 3.0."
<http://losburns.com/imaging/software/SFRedge/> , accessed 1 August 2014.
- [90] R. Bates, "Astronomical speckle imaging," ser. Physics reports. North-Holland, 1982, pp. 203–297.
- [91] K. Faulkner, C. Kotre, and M. Louka, "Veiling glare deconvolution of images produced by X-ray image intensifiers," in *Third International Conference on Image Processing and its Applications*, 1989, pp. 669 – 673.
- [92] D. Kundur and D. Hatzinakos, "Blind image deconvolutions," in *IEEE Signal Processing Magazine*, vol. 13, 1996, p. 4363.
- [93] X. Marichal, W.-Y. Ma, and H. Zhang, "Blur determination in the compressed domain using DCT information," in *IEEE International Conference on Image Processing (ICIP)*, 1999, pp. 386–390.
- [94] M. Banham and A. Katsaggelos, "Digital image restoration," in *IEEE Signal Processing Magazine*, vol. 14, 1997, pp. 24 – 41.
- [95] T. T., M. Li, H. Zhang, and C. Zhang, "Blur detection for digital images using wavelet transform," in *IEEE International Conference on Multimedia and Expo (ICME)*, vol. 1, 2004, pp. 17 – 20.
- [96] P. Marziliano, F. Dufaux, S. Winkler, and T. Ebrahimi, "A no-reference perceptual blur metric," in *IEEE International Conference on Image Processing (ICIP)*, 2002, pp. 57–60.

- [97] R. Ferzli and L. J. Karam, “A no-reference objective image sharpness metric based on the notion of just noticeable blur (JNB).” in *IEEE Transactions on Image Processing*, vol. 18, no. 4, 2009, pp. 717–728.
- [98] J. Robson and N. Graham, “Probability summation and regional variation in contrast sensitivity across the visual field,” in *Vision research*, vol. 21, no. 3, 1981, pp. 409–418.
- [99] N. D. Narvekar and L. J. Karam, “A no-reference image blur metric based on the cumulative probability of blur detection (CPBD),” in *IEEE Transactions on Image Processing*, vol. 20, no. 9, 2011, pp. 2678–2683.
- [100] C. E. Shannon, “Communication in the presence of noise,” in *Proceeding of IRE*, vol. 37, 1949, pp. 10–21.
- [101] M. Unser, “Sampling—50 Years after Shannon,” in *Proceedings of the IEEE*, vol. 88, no. 4, April 2000, pp. 569–587.
- [102] M. Vetterli, P. Marziliano, and T. Blu, “Sampling signals with finite rate of innovation,” in *IEEE Transactions on Signal Processing*, vol. 50, no. 6, June 2002, pp. 1417–1428.
- [103] P. L. Dragotti, M. Vetterli, and T. Blu, “Sampling moments and reconstructing signals of finite rate of innovation: Shannon meets Strang-Fix,” in *IEEE Transactions on Signal Processing*, vol. 55, no. 5, 2007, pp. 1741–1757.
- [104] T. Blu, P. Thévenaz, and M. Unser, “MOMS: Maximal-Order interpolation of minimal support,” in *IEEE Transactions on Image Processing*, vol. 10, no. 7, 2001, pp. 1069–1080.
- [105] G. Strang and G. Fix, “A Fourier analysis of the finite element variational method,” in *Proc. Constructive Aspect of Functional Analysis*, 1971, pp. 796–830.
- [106] B. G. R. de Prony, “Essai expérimental et analytique: sur les lois de la dilatabilité de fluides élastiques et sur celles de la force expansive de la vapeur de l’alcool, à différentes températures,” in *Journal de l’Ecole Polytechnique*, vol. 22, no. 1, 1975, pp. 24–76.
- [107] T. Thongkamwitoon, H. Muammar, and P.-L. Dragotti, “Identification of image acquisition chains using a dictionary of edge profiles,” in *Proceedings of the European Signal Processing Conference (EUSIPCO)*, 2012, pp. 1757–1761.
- [108] M. Unser, A. Aldroubi, and M. Eden, “B-spline signal processing: Part I—Theory and Part II—Efficient design and applications,” in *IEEE Transaction on Signal Process.*, vol. 41, no. 2, 1993, pp. 821–848.

-
- [109] J. A. Urigüen, T. Blu, and P. Dragotti, “FRI sampling with arbitrary kernels,” in *IEEE Transactions on Signal Processing*, vol. 61(21), 2013, pp. 5310–5323.
- [110] J. A. Urigüen, P. L. Dragotti, and T. Blu, “On the exponential reproducing kernels for sampling signals with finite rate of innovation,” in *Proceedings of the Ninth International Workshop on Sampling Theory and Applications (SampTA11)*, Singapore, 2011.
- [111] J. A. Cadzow, “Signal enhancement—a composite property mapping algorithm,” in *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 36, no. 1, 1988, pp. 49–62.
- [112] T. Blu, P. L. Dragotti, M. Vetterli, P. Marziliano, and L. Coulot, “Sparse sampling of signal innovations,” in *IEEE Signal Processing Magazine*, vol. 25, March 2008, pp. 31–40.
- [113] P. L. Dragotti and F. Homann, “Sampling signals with finite rate of innovation in the presence of noise,” in *IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, 2009, pp. 2941–2944.
- [114] M. Aharon, M. Elad, and A. Bruckstein, “K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation,” in *IEEE Transactions on Signal Processing*, vol. 54, no. 11, 2006, pp. 4311–4322.
- [115] J. Canny, “A computational approach to edge detection,” in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, 1986, pp. 679–698.
- [116] Y. C. Pati, R. Rezaeiifar, Y. C. P. R. Rezaeiifar, and P. S. Krishnaprasad, “Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition,” in *Proceedings of the 27th Annual Asilomar Conference on Signals, Systems, and Computers*, 1993, pp. 40–44.
- [117] T. Gloe and R. Bohme, “The ‘Dresden Image Database’ for benchmarking digital image forensics,” in *Proceedings of the ACM Symposium on Applied Computing*, 2010, pp. 1584–1590.
- [118] T. Ojala, M. Pietikainen, and T. Maenpää, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, 2002, pp. 971 – 987.
- [119] —, “A general local binary pattern (LBP) implementation for matlab.” <http://www.cse.oulu.fi/CMV/Downloads/LBPMatlab>, accessed 21 August 2014.

- [120] M. Kharrazi, H. T. Sencar, and N. D. Memon, “Blind source camera identification,” in *IEEE International Conference on Image Processing (ICIP)*, 2004, pp. 709–712.
- [121] Y. Chen, Z. Li, M. Li, and W. ying Ma, “Automatic classification of photographs and graphics,” in *IEEE International Conference on Multimedia and Expo (ICME)*, vol. 0, 2006, pp. 973–976.
- [122] C. W. Hsu, C.-C. Chang, and C. J. Lin, “A practical guide to support vector classification,” Department of Computer Science, National Taiwan University, Tech. Rep., 2003. [Online]. Available: <https://www.cs.sfu.ca/people/Faculty/teaching/726/spring11/svmguide.pdf>
- [123] H. Farid and S. Lyu, “Higher-order wavelet statistics and their application to digital forensics,” in *IEEE Workshop on Statistical Analysis in Computer Vision*, 2003.
- [124] S. Lyu and H. Farid, “How realistic is photorealistic?” in *IEEE Transactions on Signal Processing*, vol. 53, no. 2-2, 2005, pp. 845–850.
- [125] P. Vaidyanathan, “Quadrature mirror filter banks, m-band extensions and perfect-reconstruction techniques,” in *IEEE Transactions on Signal Processing*, vol. 4, 1987, pp. 4–20.
- [126] M. Vetterli, “A theory of multirate filter banks,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 35, no. 3. IEEE, 1987, pp. 356–372.
- [127] M. Tagliasacchi, M. V. Scanzanella, P. L. Dragotti, and S. Tubaro, “Transform coder identification,” in *IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, 2013, pp. 5785–5789.
- [128] —, “Transform coder identification with double quantized data,” in *IEEE International Conference on Image Processing (ICIP)*, 2013, pp. 1660–1664.