

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Aravind Kumar Mikkilineni

Entitled
INFORMATION HIDING IN PRINTED DOCUMENTS

For the degree of Doctor of Philosophy

Is approved by the final examining committee:

George T.-C. Chiu

Chair

Edward J. Delp

Jan P. Allebach

Patricia Davies

To the best of my knowledge and as understood by the student in the *Research Integrity and Copyright Disclaimer (Graduate School Form 20)*, this thesis/dissertation adheres to the provisions of Purdue University's "Policy on Integrity in Research" and the use of copyrighted material.

Approved by Major Professor(s): George T.-C. Chiu

Approved by: David Anderson

Head of the Graduate Program

07/25/2012

Date

**PURDUE UNIVERSITY
GRADUATE SCHOOL**

Research Integrity and Copyright Disclaimer

Title of Thesis/Dissertation:

INFORMATION HIDING IN PRINTED DOCUMENTS

For the degree of Doctor of Philosophy

I certify that in the preparation of this thesis, I have observed the provisions of *Purdue University Executive Memorandum No. C-22*, September 6, 1991, *Policy on Integrity in Research*.*

Further, I certify that this work is free of plagiarism and all materials appearing in this thesis/dissertation have been properly quoted and attributed.

I certify that all copyrighted material incorporated into this thesis/dissertation is in compliance with the United States' copyright law and that I have received written permission from the copyright owners for my use of their work, which is beyond the scope of the law. I agree to indemnify and save harmless Purdue University from any and all claims that may be asserted or that may arise from any copyright violation.

Aravind Kumar Mikkilineni

Printed Name and Signature of Candidate

July 24, 2012

Date (month/day/year)

*Located at http://www.purdue.edu/policies/pages/teach_res_outreach/c_22.html

INFORMATION HIDING IN PRINTED DOCUMENTS

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Aravind K. Mikkilineni

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

August 2012

Purdue University

West Lafayette, Indiana

ACKNOWLEDGMENTS

These many years as a member of the VIPER lab have been very beneficial to me in the completion of this thesis. I would like to thank my advisor, Prof. Delp, for making available many opportunities to expand and improve my understanding in the areas of image and video processing. He has also willingly put up with me these many years, providing encouragement, advice, “The Laser” when necessary, and has done his best to “keep my mind right.” I would like to thank my advisor, Prof. Chiu, as well as committee members Prof. Allebach and Prof. Davies for providing critical feedback during the progress of my work which has forced me to reevaluate assumptions and methodologies to get me on the right track toward completion. The work in this thesis started out as a project between Prof. Allebach, Prof. Chiu, and Prof. Delp. I am grateful to have been a part of the project from the beginning. It is unusual to experience first hand the different levels of research and analysis starting from general high level modeling of the problem to low level modeling and simulation of every individual aspect.

I would also like to thank Prof. Gelfand for providing key insights into approaches to problem solving and modeling. The perspective to modeling problems that he brought to the table gave me pause and made me look at earlier work I had done on this project in a new way that allowed me to progress in the later work.

Finally I would like to thank the many people with whom I have had the opportunity to work with while at Purdue. I have learned a lot, both technical and non-technical, through the many lively discussions with everyone, and I hope along the way I was able to help them as well.

Portions of this work were supported by National Science Foundation Grants CNS-0219893 and CNS-0524540.

TABLE OF CONTENTS

	Page
LIST OF TABLES	v
LIST OF FIGURES	vii
ABSTRACT	xiv
1. INTRODUCTION	1
1.1 Electrophotographic Printing Process	2
1.2 Printer Identification - Current Approaches	4
1.3 Data Hiding in Printed Documents - Current Approaches	4
1.3.1 Data Hiding in Text	5
1.3.2 Data Hiding in Halftone Images	6
1.4 Goals of this Research	8
1.5 Contributions of The Research Described in This Thesis	10
2. INTRINSIC SIGNATURES FOR EP PRINTERS	12
2.1 Banding	12
2.2 Texture Features	14
2.2.1 System for Intrinsic Printer Identification	37
2.3 Forensic Characterization and Variability in Document Content . .	54
2.4 Survivability of the Intrinsic Signature	61
2.4.1 System Overview	62
2.4.2 Attacks	64
2.4.3 Experiments	66
2.4.4 Results	66
2.5 Time Variation of the Intrinsic Signature	70
2.5.1 Experiments and Results	81
3. EXTRINSIC SIGNATURES FOR EP PRINTERS	87
3.1 EP Embedding Techniques	87
3.1.1 Relationship Between Laser Power and Developed Dot Profile	88
3.1.2 Effects of Laser Intensity Modulation	91
3.2 Modulation Parameters and Image Quality	93
3.3 Embedding and Detection	95
3.3.1 System 1 - Time Domain	96
3.3.2 System 1 - Experimental Results	101
3.3.3 System 2 - Frequency Domain	103
3.3.4 System 2 - Experimental Results	107

	Page
3.4 Non-Extrinsic Embedding Techniques	112
3.4.1 Embedding and Detection	115
3.4.2 Results	121
3.5 Counterfeit and Tamper Detection	125
3.5.1 Counterfeit Detection	128
3.5.2 Detecting Content-Tampering	128
3.5.3 Data-Hiding Technique Parameters	130
3.5.4 Constructing a Threshold for Counterfeit Detection	131
3.5.5 Results	131
3.6 Printer Model	133
4. SUMMARY AND FUTURE DIRECTIONS	147
4.1 Summary	147
4.2 Contributions Resulting From This Research	149
4.3 Future Directions	150
4.4 Publications Resulting From This Work	151
LIST OF REFERENCES	156
VITA	164

LIST OF TABLES

Table	Page
2.1 Principle banding frequencies for several printers.	14
2.2 Printers used for classification.	47
2.3 Confusion matrix for GLCM based printer identification using 22 features, 5NN classifier, and $dr = 2$. (Blank indicates -0-)	50
2.4 Confusion matrix for GLCM based printer identification using 4 features, 5NN classifier, and $dr = 2$. (Blank indicates -0-)	55
2.5 Confusion matrix for GLCM based printer identification using 22 features, SVM classifier, and $dr = 2$. (Blank indicates -0-)	56
2.6 Four variables considered for forensic identification experiment.	57
2.7 Percent correct classification for varying font size. (% after SVM)	58
2.8 Percent correct classification for varying font type. (% after SVM)	59
2.9 Percent correct classification for varying font size. (% after SVM)	60
2.10 Percent correct classification for varying age. Training and testing data generated 5 months apart.	61
2.11 Printers used in attack experiments.	64
2.12 Images of attacked “e”s.	77
2.13 Printers used in our experiments.	80
2.14 Confusion matrix for Experiment 1. Best training set with 56% average accuracy.	83
2.15 Confusion matrix for Experiment 1. Worst training set with 37% average accuracy.	83
2.16 Confusion matrix for Experiment 2. Best training set with 71% average accuracy.	84
2.17 Confusion matrix for Experiment 2. Worst training set with 46% average accuracy.	84
2.18 Confusion matrix for Experiment 3. 52% average accuracy.	85
2.19 Confusion matrix for Experiment 4. 65% average accuracy.	85

Table	Page
3.1 Percent decoding error at character level.	101
3.2 Percent decoding error at line level; Symbol error for current embedding model.	102
3.3 Preliminary results for determining document genuinity. ($T = 92.38$, $s = 3.08$)	132
3.4 Parameters for dot model estimated using embedded edges.	137
3.5 Model coefficients for each parameter.	137

LIST OF FIGURES

Figure	Page
1.1 Diagram of the electrophotographic process: (A) charging, (B) exposure, (C) development, (D) transfer, (E) fusing, (F) cleaning.	3
2.1 Banding frequencies from the image produced by the printer and the printer mechanism of an HP LaserJet 4050.	13
2.2 (a) Projection and spectral analysis of a 25% fill pattern printed on an HP LaserJet 6MP. (b) Spectrum of the projection (FFT) showing peaks at 132 and 150 cycles/inch.	15
2.3 Idealized text character.	17
2.4 Generation of $C(n, m)$	18
2.5 Image for GLCM Example 1. The non-shaded region is the region of interest from which the GLCM is obtained. Each number represents a graylevel pixel value between 1 and 5.	19
2.6 Image for GLCM Example 2. The non-shaded region is the region of interest from which the GLCM is obtained. Each number represents a graylevel pixel value between 1 and 5. This image has a structure imposed upon the pixel values that simulates banding in the vertical direction.	20
2.7 Image for GLCM Example 3. The non-shaded region is the region of interest from which the GLCM is obtained. Each number represents a graylevel pixel value between 1 and 5. This image has a structure imposed upon the pixel values that simulates banding in the vertical direction.	22
2.8 Image for GLCM Example 4. The non-shaded region is the region of interest from which the GLCM is obtained. Each number represents a graylevel pixel value between 1 and 5. This image has a structure imposed upon the pixel values that simulates banding in the vertical direction similar to the image in Figure 2.6, however the region of interest in this case is not rectangular.	24
2.9 Graylevel co-occurrence Example 5. Subfigure (a) shows a 600×600 pixel 8-bit grayscale image with uniform graylevel. Subfigure (b) shows the corresponding graylevel co-occurrence matrix where darker pixels indicate larger values in the GLCM.	26

Figure	Page
2.10 Graylevel co-occurrence Example 6. Subfigure (a) shows a 600×600 pixel 8-bit grayscale image with a 0.0017 cycle/pixel sinusoidal variation in the vertical direction representing 1 cycle/inch banding in a 600 DPI print process. Subfigure (b) shows the corresponding graylevel co-occurrence matrix where darker pixels indicate larger values in the GLCM.	27
2.11 Graylevel co-occurrence Example 7. Subfigure (a) shows a 600×600 pixel 8-bit grayscale image with a 0.250 cycle/pixel sinusoidal variation in the vertical direction representing 150 cycle/inch banding in a 600 DPI print process. Subfigure (b) shows the corresponding graylevel co-occurrence matrix where darker pixels indicate larger values in the GLCM.	28
2.12 Graylevel co-occurrence Example 8. Subfigure (a) shows a 600×600 pixel 8-bit grayscale image with a 0.4167 cycle/pixel sinusoidal variation in the vertical direction representing 250 cycle/inch banding in a 600 DPI print process. Subfigure (b) shows the corresponding graylevel co-occurrence matrix where darker pixels indicate larger values in the GLCM.	29
2.13 Graylevel co-occurrence Example 5 with Gaussian noise. Subfigure (a) shows a 600×600 pixel 8-bit grayscale image with uniform graylevel, plus Gaussian noise. Subfigure (b) shows the corresponding graylevel co-occurrence matrix where darker pixels indicate larger values in the GLCM.	30
2.14 Graylevel co-occurrence Example 6 with gaussian noise. Subfigure (a) shows a 600×600 pixel 8-bit grayscale image with a 0.0017 cycle/pixel sinusoidal variation in the vertical direction, representing 1 cycle/inch banding in a 600 DPI print process, plus Gaussian noise. Subfigure (b) shows the corresponding graylevel co-occurrence matrix where darker pixels indicate larger values in the GLCM.	31
2.15 Graylevel co-occurrence Example 7 with Gaussian noise. Subfigure (a) shows a 600×600 pixel 8-bit grayscale image with a 0.250 cycle/pixel sinusoidal variation in the vertical direction, representing 150 cycle/inch banding in a 600 DPI print process, plus Gaussian noise. Subfigure (b) shows the corresponding graylevel co-occurrence matrix where darker pixels indicate larger values in the GLCM.	32
2.16 Graylevel co-occurrence Example 8 with Gaussian noise. Subfigure (a) shows a 600×600 pixel 8-bit grayscale image with a 0.4167 cycle/pixel sinusoidal variation in the vertical direction, representing 250 cycle/inch banding in a 600 DPI print process, plus Gaussian noise. Subfigure (b) shows the corresponding graylevel co-occurrence matrix where darker pixels indicate larger values in the GLCM.	33

Figure	Page
2.17 System diagram of intrinsic printer identification scheme using texture based features.	37
2.18 Binary classification problem solved using a single linear hyperplane. .	39
2.19 Linear support vector machine classification example. The two classes of data points represented by 'X' and 'O' are linearly separable.	42
2.20 Linear support vector machine classification example. Two classes of data points represented by 'X' and 'O' are linearly separable in this case. The large 'X' and 'O' markers indicate the training data points, with those markers in bold indicating the two support vectors in this case. In this case there is a linear boundary separating the two training classes bisecting the line segment connecting the two support vectors.	43
2.21 Non-linear support vector machine classification example. The two classes of data points represented by 'X' and 'O' are not linearly separable. .	45
2.22 Non-linear support vector machine classification example. Two classes of data points represented by 'X' and 'O' are not linearly separable in this case. The large 'X' and 'O' markers indicate the training data points, with those markers in bold indicating the nine support vectors in this case. In this case there is a non-linear boundary separating the two training classes. Black dots mark the decision boundary obtained in this case.	46
2.23 Percent correct classification of individual feature vectors versus dr using all 22 features and 5NN classifier.	49
2.24 Percent correct classification of individual feature vectors versus dr using 4 selected features and 5NN classifier.	51
2.25 Scatter plot of features h_{Img} vs σ_{Img}^2	52
2.26 Scatter plot of features μ_r vs $Energy$	53
2.27 System diagram of printer identification scheme to handle attacks. . . .	63
2.28 SVM classification results of individual "e"s after attacking with a single frequency sinusoid (attack 1).	67
2.29 SVM classification results of individual "e"s after attacking with a random frequency sinusoid (attack 3).	68
2.30 SVM classification results of individual "e"s after attacking with Gaussian noise (attack 5).	69

Figure	Page
2.31 Scatter plot of first two LDA features from non-attacked “e”s. Ellipses identify the boundary of the training clusters. All points are from the non-attacked document for that printer and are used to define the training cluster. (a) LDA on GLCM Features. (b) LDA on DFT Features. . . .	71
2.32 Scatter plot of first two LDA features for attack 1 (fixed sinusoid: $A = 50$, $f = 90$). Ellipses identify the boundary of the training clusters. All other points are from the attacked document for that printer. (a) LDA on GLCM Features. (b) LDA on DFT Features.	72
2.33 Scatter plot of first two LDA features for attack 3 (random frequency sinusoid: $A = 25$). Ellipses identify the boundary of the training clusters. All other points are from the attacked document for that printer. (a) LDA on GLCM Features. (b) LDA on DFT Features.	73
2.34 Scatter plot of first two LDA features for attack 4 (random frequency binarized sinusoid: $\nu = 4$). Ellipses identify the boundary of the training clusters. All other points are from the attacked document for that printer. (a) LDA on GLCM Features. (b) LDA on DFT Features.	74
2.35 Scatter plot of first two LDA features for attack 5 (Gaussian noise: $\sigma = 8$). Ellipses identify the boundary of the training clusters. All other points are from the attacked document for that printer. (a) LDA on GLCM Features. (b) LDA on DFT Features.	75
2.36 System diagram for forensic printer identification.	76
2.37 Linear discriminant analysis applied to a two class problem and two-dimensional feature space. The two classes are not linearly separable using either of the two original features. Once the data is projected onto the LDA axis it becomes linearly separable.	79
2.38 Each subplot shows the intraclass (solid line) and interclass (dashed line) distances for a single class. The top-left plot is for class one with numbering continuing in row order to class 9 in the bottom right plot.	82
2.39 Distributions (pmf) of h_{xy1} for P01 over all dates.	86
2.40 Distributions (pmf) of ρ for P01 over all dates.	86
3.1 Process block diagram for embedding of extrinsic signature using laser intensity modulation.	88
3.2 Laser power profile.	89
3.3 Dot profiles for varying V_{ref}	91
3.4 Dot size versus V_{ref}	92

Figure	Page
3.5 Large amplitude exposure modulation. First line has no modulation, Second line has 20 cycles/in modulation. Third line has 40 cycles/in modulation.	93
3.6 Determination of the R60 transition point.	94
3.7 Relation between amplitude modulation parameters and raggedness. . .	96
3.8 Modulation scheme for text documents.	96
3.9 Process for extracting embedded information - System 1.	98
3.10 Process steps to find edges from which to extract edge profile.	99
3.11 Line level results for $(f_0, f_1) = (30, 60)$, $A = 0.2$. (a) Percentage of characters in each line of text, p_l^0 and p_l^1 , voting for either B_0 or B_1 respectively. (b) Difference between p_l^0 and p_l^1	103
3.12 Line level results for $(f_0, f_1) = (60, 120)$, $A = 0.2$. (a) Percentage of characters in each line of text, p_l^0 and p_l^1 , voting for either B_0 or B_1 respectively. (b) Difference between p_l^0 and p_l^1	104
3.13 Modulation scheme for text documents - System 2.	104
3.14 Process for extracting embedded informationi - System 2.	106
3.15 Test document for system 2.	107
3.16 Baseline power spectral density from 432 characters.	108
3.17 PSD of last character in text line with $\mathbf{b} = \{01010000\}$	109
3.18 PSDs of 10 characters in text line with $\mathbf{b} = \{01010000\}$. PSD increases monotonically from the leftmost character to the rightmost character in a text line.	109
3.19 Decoded symbols using only last character in text line.	110
3.20 Decoded symbols using last 25% of characters in text line.	110
3.21 Decoded symbols using only last character in text line and 6 highest bits.	111
3.22 Decoded symbols using last 25% of characters in text line and 6 highest bits.	112
3.23 Symbol level operational curve using DFT decoder.	113
3.24 Symbol level operational curve using correlation decoder.	113
3.25 Bit level operational curve using DFT decoder.	114
3.26 Bit level operational curve using correlation decoder.	114

Figure	Page
3.27 Large amplitude exposure modulation. First line has no modulation. Second line has 20 cycles/in modulation. Third line has 40 cycles/in modulation.	116
3.28 Modulation scheme for text documents.	116
3.29 Block diagram of data hiding system.	118
3.30 Example of embedded signal.	119
3.31 The top plot shows an instance of \hat{s} , a high pass filtered edge profile. The center plot shows the same signal after matched filtering, $d[k]$. The bottom plot illustrates sampling instances $d[4iT]$ at which decisions are made as to which bit is present in each period T . In this case all bits were detected correctly.	122
3.32 Experimental pdf of $n_h[k]$ overlaid on top of Gaussian pdf with estimated $\sigma_{n_h}^2 = 0.0661$	123
3.33 Autocorrelation of $n_h[k]$	124
3.34 Block diagram of our modified embedding system.	126
3.35 Estimated values of μ and parametric fit to the data. Dashed line represents original parameter values as defined in [94].	138
3.36 Estimated values of Λ and parametric fit to the data. Dashed line represents original parameter values as defined in [94].	139
3.37 Estimated values of $\Delta\tilde{x}$ and parametric fit to the data. Dashed line represents original parameter values as defined in [94].	140
3.38 Vertical bar printed on an HP Color LaserJet 4500 at 600 DPI and scanned using an Epson 4490 at 4800 DPI. Dimensions of the bar as printed are 140×10 px at 600 DPI or approximately 0.233×0.0167 inches.	142
3.39 Vertical bar simulated using Chiang's method with original parameters.	143
3.40 Vertical bar simulated using Chiang's method with new parameters.	144
3.41 Results of simulation methods and experiments, embedding a square wave with amplitude 0.5. Subfigure (a) shows the relative edge offset of the left edge, while subfigure (b) shows the 20% to 80% transition width of the left edge.	145
3.42 Results of simulation methods and experiments, embedding a square wave with amplitude 0.5. Subfigure (a) shows the relative edge offset of the right edge, while subfigure (b) shows the 20% to 80% transition width of the right edge.	146

ABSTRACT

Mikkilineni, Aravind K. Ph.D., Purdue University, August 2012. Information Hiding in Printed Documents. Major Professors: George T.-C. Chiu, School of Mechanical Engineering, and Edward J. Delp, School of Electrical and Computer Engineering, School of Mechanical Engineering.

In today's digital world securing different forms of content is very important in terms of protecting copyright and verifying authenticity. One example is watermarking of digital audio and images. We believe that a marking scheme analogous to digital watermarking but for documents is very important. There currently exist techniques to secure documents such as bank notes using paper watermarks, security fibers, holograms, or special inks. There are a number of applications in which it is desirable to be able to identify the technology, manufacturer, model, or specific unit that was used to print a given document even if the printer in question does not make use of these existing security devices to explicitly identify itself. It would be useful to achieve the same or a better level of protection without the use of any additional devices or technologies. Two strategies are proposed for printer identification based upon examination of a printed document. The first strategy is passive. It involves characterization of the printer by finding features in the printed document that are intrinsic to that particular printer, model, or manufacturer's products. The second strategy is active. It involves the embedding of an extrinsic signature into a printed page. This signature can be generated by modulating the process parameters of the printer mechanism to encode identifying information such as the printer serial number and date of printing. It is shown that good separation between printers is achievable using gray-level co-occurrence based texture features obtained from text documents. Experiments using ten printers and a support vector machine classifier show very low classification error even between printers with the same electrome-

chanical structure. The technique is also shown to work for various font sizes, font types, paper types, and printer age. The features are observed to migrate with the age of the consumables indicating that it may be possible to estimate the age of the consumables at the time of printing. In addition, the intrinsic nature of the features makes it difficult to obscure or remove them without physically modifying the printer itself. Combining both texture features and banding features it is possible to identify a printer under several attack scenarios. A coding technique for embedding extrinsic signatures in text documents is presented. Both time and frequency domain signaling and detection schemes are investigated. It is shown that better performance is achieved using a time domain signaling scheme with a correlation detector due to the limited length of text character edges. It is also shown that by treating the document as a communication channel, a coding technique allowing approximately 3600 bits in a full page of 12 point text is achievable with a 7.74% bit error rate. By using the data hiding technique described above, a counterfeit and tamper detection method based on combinatorial group testing is developed and investigated. The low error rate achievable by the data hiding system allows reliable determination of document authenticity and the location of tampered data within a document. From results of previous work a printer dot model is proposed to simulate the printing of cluster-dot halftone patterns. It has been shown that the original parameters chosen for that model do not adequately represent vertical edges in saturated regions such as text. Estimating the parameters by minimizing the error between the simulated and experimental edge profiles and edge sharpness for both the left and right edges provides values that more accurately represent the actual edge with and without embedded signals.

1. INTRODUCTION

In today's digital world securing different forms of content is important in terms of protecting copyright and verifying authenticity [1–7]. One example is watermarking of digital audio and images. A marking scheme analogous to digital watermarking but for documents would be useful for many of the same reasons [1]. Printed material is a direct accessory to many criminal acts. Examples include forgery or alteration of documents used for purposes of identity, security, or recording of transactions. In the course of conducting illicit activities printed material such as instruction manuals, team rosters, meeting notes, and correspondence may be used. In both cases, the ability to identify the device or type of device used to print the material in question would provide a valuable aid for law enforcement and intelligence agencies. Additionally, the ability to print secure documents is also needed for average users to print documents such as boarding passes, postage, and bank transactions.

There currently exist techniques to secure documents such as bank notes using paper watermarks, security fibers, holograms, or special inks [8–11]. The problem is that the use of these security techniques can be cost prohibitive. Most of these techniques either require special equipment to embed the security features, or are too expensive for an average consumer. Additionally, there are a number of applications in which it is desirable to be able to identify the technology, manufacturer, model, or specific unit that was used to print a given document even if the printer in question does not make use of these existing security devices to explicitly identify itself. It would be useful to achieve the same or better level of protection without the use of any additional devices or technologies.

Several methods, such as those proposed in [12, 13], have been developed for use as counterfeit deterrents that rely on various print quality metrics obtained from specially designed patterns that are difficult to reproduce. However, these methods

only help detect counterfeit documents, and do not necessarily identify the printer that created them.

Two strategies are proposed for printer identification based upon examination of a printed document. The first strategy is passive. It involves characterization of the printer by finding features in the printed document that are intrinsic to that particular printer, model, or manufacturer’s products. This is referred to as the *intrinsic signature* [14]. Development of analysis tools for the detection of intrinsic signatures in a printed page requires an understanding of the printer mechanism and a model that predicts its behavior. The second strategy is active. It involves the embedding of an *extrinsic signature* into a printed page. This signature can be generated by modulating the process parameters of the printer mechanism to encode identifying information such as the printer serial number and date of printing. Extrinsic signature can be detected by using the tools developed for intrinsic signature detection.

Several printing technologies exist on the market today, however this research will focus primarily on electrophotographic (EP) printers, more commonly known as laser printers. In the remainder of this section the EP printing process will be introduced. Properties of this process are used in this research to develop methods for both intrinsic characterization and extrinsic signature embedding.

1.1 Electrophotographic Printing Process

An understanding of the electrophotographic (EP) printing process is necessary in order to gain insight into the types of features that can be used for intrinsic characterization. This same understanding can then be used to develop methods for extrinsic signature embedding. The printed output from any printer contains defects or artifacts caused by electromechanical fluctuations or imperfections in the print mechanism [15]. As will be shown in the Chapter 2, these “print quality defects” are

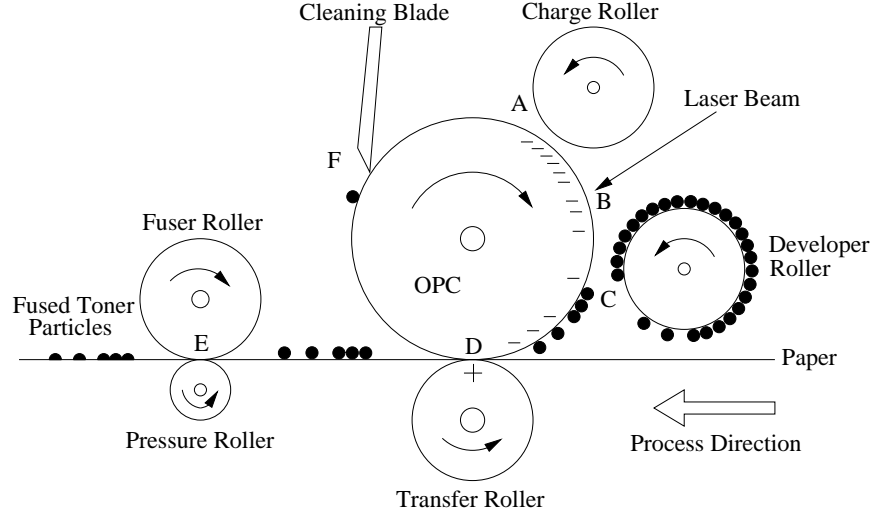


Figure 1.1. Diagram of the electrophotographic process: (A) charging, (B) exposure, (C) development, (D) transfer, (E) fusing, (F) cleaning.

directly related to the printer mechanism and can be viewed as an intrinsic signature of the printer.

Figure 1.1 is a side view of a typical EP printer. The print process consists of six steps. The first step is to uniformly charge a photoconducting drum (OPC)¹. A laser scans the drum and discharges specific locations on the drum. The discharged locations on the drum attract toner particles which are then attracted to the paper which has an opposite charge. The paper with the toner particles then passes through a fuser and pressure roller which melt and permanently affix the toner to the paper. Finally a blade or brush cleans any excess toner from the OPC.

Defects or artifacts in the printed output are typically caused by fluctuations in the angular velocity of the OPC, gear eccentricity, gear backlash, and polygon mirror wobble. These imperfections are directly tied to the electromechanical properties of the printer and create corresponding fluctuations in the developed toner on the printed page [16, 17]. Several methods can be employed to characterize a printer

¹OPC stands for organic photoconducting drum, however many photoconducting drums in use today are made of inorganic material. The term OPC is still used for historical purposes.

using these defects. The method chosen depends upon the content of the document being examined.

In addition, various process parameters can be controlled to “alter” the artifacts in the printed output to embed an extrinsic signature. The method by which these parameters are controlled determine the embedding capacity and whether the intrinsic signature remains intact when an extrinsic signature is embedded.

1.2 Printer Identification - Current Approaches

Current approaches to EP printer identification tend to rely on image quality measures or visually identifiable periodic structures in the document [12, 18]. Such measures include image sharpness, toner fusing characteristics, dot gain, and asymmetry of toner distribution [19]. Spatially periodic noise, such as that due to a damaged OPC, is visually identifiable and can also be used to identify a specific printer [20]. Other methods rely on specific test patterns. For example, one method uses pixel measures of a test pattern (black square) including histogram peaks and directional gradients that are classified using an artificial neural network [21].

Each of these methods relies on features that are not intrinsic to the printer, but are instead a function of the particular printer and consumables used by the printer. By changing the toner cartridge, for example, any spatially periodic noise that the previous cartridge exhibited may disappear.

1.3 Data Hiding in Printed Documents - Current Approaches

Current approaches to hiding data in a printed document tend to rely on standard “digital” watermarking techniques in which the information is embedded into the digital document before it is printed. The assumption is that the information is coded in such a way that it survives the printing process, and perhaps also the scanning process if the information is to be digitally extracted. Several techniques for this type of data hiding are described in the following sections.

1.3.1 Data Hiding in Text

Various methods for data hiding in text documents have been developed for securing documents. One of the earliest methods involves the shifting of elements in a text document as described in [22–24]. In this approach, copyright protection for text documents is provided by embedding information specific to the recipient or source in order to deter unauthorized distribution. The methods developed to encode this information into each page involve the shifting of textual elements in the document by amounts imperceptible to the human viewer to encode individual bits of data. These textual elements can be lines, words, or individual characters. With line shifting, every other line is shifted slightly up or down, approximately 1/600th inch, to encode a one or zero. To detect the shifts in a document, no prior information about the original is necessary since the information is differentially embedded. The baseline locations can be estimated in the scanned document and by measuring the relative distances between the baselines of adjacent text lines an estimate of the embedded data can be obtained. It is shown that this type of encoding is robust to scan-print attacks and photocopy generation loss, although some variability in the detection arises due to errors in the scan process such as rotation. Another detection method for line shifting that is more robust to imaging errors such as scan rotation is the use of the relative distance between centroids of adjacent lines of text. For this detection method the original document is needed in order to properly decode the information because the centroids are dependent on the content of each text line.

Word and character coding allow more data to be embedded into a page of text, but are not as robust as line coding due to the fact that each shift is encoded in a smaller portion of the printed page. In addition, most word processors will vary the spacing between adjacent words and characters, so estimation of the shifts using the differential method used for lines will not work in this case and the original document will be needed as a reference for decoding. These shifts could instead be used as a fragile watermark to detect alterations to a document.

One major drawback to these feature shifting methods is that they are easily defeated. An attacker can simply scan the document and use readily available tools to extract and reformat the text, remove any shifts, and reprint it without the encoded information.

More recent data hiding methods for text documents have focused on modulating the properties of individual text characters. Examples include full page watermarks [25], color quantization of individual characters [26–29] which requires that the text be printed in color and be halftoned, and perturbation of character outlines [30, 31] which require Bezier spline models of the text being printed.

1.3.2 Data Hiding in Halftone Images

The previous method deals with encoding information in text, but documents may also contain halftone images. Halftoning is the process of converting a continuous tone image into an image having only a finite number of levels, typically 2 for printed images [32]. When viewed from a distance, these images resemble the original. Numerous methods have been developed for watermarking of halftone images. Most of these methods involve modifying the halftone patterns used when printing the image. The three basic methods are the use of multiple dithering matrices, conjugate halftone screens, or angular variation of the halftone screen [33], or angular variation of the halftone dots themselves [34]. In the first method, the dithering matrix changes from tile to tile and can be used to encode information. Detection involves estimating the statistical properties of the halftone patterns and their variation across the image. The second method involves conjugate halftone screens where two screens are used to form two halftone images and the data are embedded in the correlations between the two screens [33]. The third basic method involves encoding the data in the angular orientation of the screen in each halftone cell. In this case, each angular orientation can represent multiple bits depending on the number of angles the halftone screen can be generated and detected at.

Another method of data hiding with halftone patterns is described in [35]; watermarks are embedded in duplex printed pages such that when the page is held up to a light source a fake watermark will appear. This method relies on halftone images that are printed opposite each other on either side of the page. This technique requires a high degree of control over the registration of each side of the document to make sure the halftone patterns line up. A similar technique is also described in [36] which can be used for one sided printing with verification using a transparent screen.

Data hiding in halftone images can also be performed using continuous tone image watermarking techniques [4, 5, 7]. These methods first embed a watermark into a continuous tone image and then print it at a high resolution to create the document. To detect the watermark, the document is scanned and transformed back into a continuous tone image after which an appropriate method for detecting the watermark is used. The type of watermark embedded has to be one that can survive the print-scan process.

The watermarking method described in [37–39] has been shown to be very robust against many types of attacks, including the print-scan attack and those involving global affine or local non-linear transformations. Being able to survive these two types of transformations is important because a document might be wrinkled or torn. First a compressed and encrypted message is encoded into a reference watermark block approximately 100 pixels square. The block is then flipped and mirrored to create a macroblock consisting of 4 copies of the original block. The purpose of the flipping is to visually decorrelate the embedding structure and also to assist in later estimation of local non-linear transformations. The macroblock is then tiled and embedded into the image which is then printed. An autocorrelation detection scheme as described in [40] can be used to estimate any global affine transformations that have been applied to the scanned document. Local non-linear transformations can then be estimated using a similar method within each macroblock.

Detectability of the embedded information and the printed image quality can both be improved by considering the printer’s halftoning process while embedding

the watermark [41]. The method described in [41] jointly optimizes the printed image quality and the detectability of the watermark. This approach uses direct binary search (DBS) halftoning [42] and a spread spectrum watermark [43]. The halftoned image is modeled as a bitmap. The detection process relies on a reconstructed continuous tone version of the document, and not the specific halftone patterns used in its creation. Therefore there are multiple bitmaps of the same watermarked image which will yield good detection results. Assuming that the bitmap image is $M \times N$ pixels, there are then 2^{MN} possible bitmaps, only a subset of which will visually resemble the original image data and simultaneously allow detection of the watermark. Each of these bitmaps will differ in visual quality and the goal is to pick the one which best maximizes both visual quality and watermark detectability. This is achieved by using a modified version of a direct binary search such that at each iteration both watermark detectability and perceptual image quality metrics are used jointly to optimize the halftone image. This method is shown to be more robust against many common image processing operations such as JPEG compression and histogram equalization when compared to prior methods which do not explicitly take into account the printing process.

A lower-level data hiding method for halftone images uses sub-pixel modulation to embed information into a halftone image by shifting individual halftone dots [44]. This method was developed specifically for printers with pulse width modulation (PWM) control of dot placement using AM/FM halftoning and is limited to use within highlight or mid-tone gray-levels.

1.4 Goals of this Research

Document security involves the prevention, detection, and possible deterrence of several possible attacks. The first is copying: is the document first or n^{th} generation? The second is forgery: has the document been altered in any way? The third is fingerprinting: where did the document originate, and through what devices did it

pass through to get to its current state? The fourth is authentication: is the document authentic? To achieve these the security features embedded into the document need to be content dependent and both robust and fragile.

Through the course of this research, techniques have been developed for both intrinsic characterization and extrinsic embedding that can together achieve these properties. It is shown that this can be accomplished by designing these techniques around the intrinsic properties of the printer.

As described in Section 1.2, current printer identification techniques are not based directly on any intrinsic properties of the printer, which means that they typically will not work if something as simple as a cartridge is changed in a printer. In Section 2 intrinsic signatures for EP printers are introduced that are shown to be directly related to the electromechanical properties of the printer itself.

Similarly, the data hiding techniques described in Section 1.3 all embed information into the document before it is printed. Several methods also exist which embed information into the document at the hardware level of the printer. These techniques exploit the way that the printer puts marks on the paper. This is different from modifying the document or image to be printed or the printer driver. Embedding at the hardware level allows access to a much larger marking domain and the potential for increased security. Additionally, circumvention of the embedding of such marks is more difficult since the embedding step exists in the hardware of the printing device instead of as a software module. Changing the document itself will not affect the embedding. Since the information is embedded by modulating print process parameters, the embedded data can be considered an extrinsic signature.

Two achievements are outlined in this thesis. The first is the development of a printer identification method based on texture features for documents containing text. This printer identification that uses graylevel co-occurrence texture features shows promising results. An adaptation of this method for ink-jet printers was presented in [45] with promising results. The second accomplishment in this research is the development of a coding technique for the embedding of extrinsic signatures in text

documents [46]. It has been shown that by treating the document as a communication channel, a coding technique allowing up to 3600 bits per page (12 point text) is achievable with very low bit error rate.

1.5 Contributions of The Research Described in This Thesis

In this thesis, techniques for intrinsic printer identification and extrinsic signature embedding are described. In this research, printer identification using texture features, and signature embedding in text documents have been investigated. The primary contributions in the area of intrinsic printer identification are as follows:

- The use of gray-level co-occurrence texture features as an intrinsic signature has been investigated. These features are shown to be robust to printer and consumable age as well as small changes in font size, font type, and paper type. In addition, the features are shown to exhibit behavior that may allow determination of consumable age.
- Combined use of gray-level co-occurrence features and banding features were investigated. These features when used together are shown to be robust against several attack modes that attempt to obscure or destroy the intrinsic signature of a printer.

The primary contributions in the area of extrinsic signature embedding and printer identification are as follows:

- A channel model for printed text documents was developed to aid in the embedding and detection of extrinsic signatures.
- Time and frequency domain embedding have been investigated with the use of both correlation, spectral analysis, and matched filter based detectors. It is shown that up to 3600 bits can be reliably embedded within a page of 12 point text.

- A method to detect counterfeit and content-tampering has been proposed that is based on a combinatorial group testing framework. It is shown that by using the proposed embedding system, document originality can be determined, and changes to key elements in a document can be identified.
- Parameters for a printer dot model that characterize printed edges in text have been obtained and have been shown to provide a good representation of printed output matching that of the physical embedding system.

2. INTRINSIC SIGNATURES FOR EP PRINTERS

An intrinsic signature is a set of features obtained from a printed document that can be used to identify the device used to create the document. Such features ideally are directly related to the electromechanical properties of the printing device, however this need not always be the case as will be seen in the following sections. Several of the goals outlined in the introduction can be achieved through use of the intrinsic signature. These include device identification and forgery detection.

2.1 Banding

In EP printing, a major artifact in the printed output is *banding* which is defined as the artifacts that are due to quasiperiodic fluctuations of process direction parameters in the printer. With reference to Figure 1.1, these are primarily due to fluctuations in the angular velocity of the OPC and result in non-uniform scan line spacing. This causes a corresponding fluctuation in developed toner on the printed page [16]. The appearance of the banding artifact is as alternating light and dark bands perpendicular to the process direction. This is most easily seen in halftone images containing large mid-level gray regions. Spectral analysis of the banding present in a printed document shows that the frequency content of the banding corresponds directly to the mechanical properties of the printer, specifically the gearing. An analysis of the rotation rates of the printer gears shown in Figure 2.1 reveals that the frequency of tonal variation in the printed image are primarily related to the rotations of the motor and OPC gears.

To estimate the banding frequencies of an EP printer, midtone graylevel patches created with a 25% line fill pattern were printed and analyzed [47]. This pattern was printed on a set of EP printers and then scanned at 2400dpi. Each scanned image

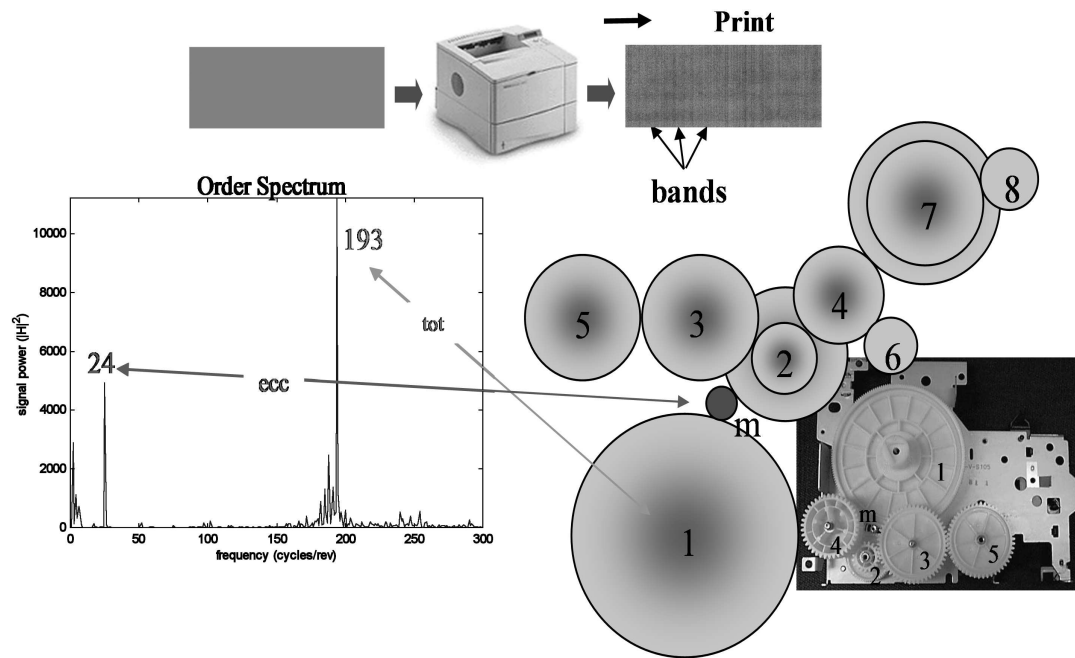


Figure 2.1. Banning frequencies from the image produced by the printer and the printer mechanism of an HP LaserJet 4050.

Table 2.1. Principle banding frequencies for several printers.

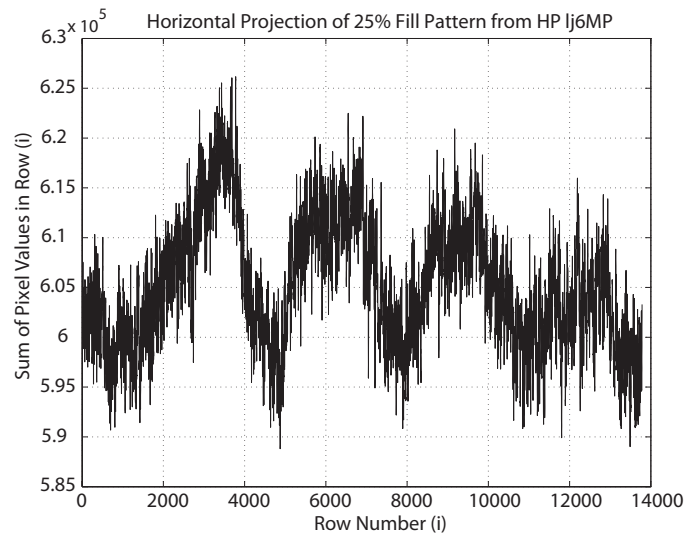
Printer Model	Principal Banding Frequencies (cycles/inch)
HP LaserJet 5MP	37, 74
HP LaserJet 6MP	132, 150
HP LaserJet 1000	27, 69
HP LaserJet 1200	69
HP LaserJet 4050	51, 100
Samsung ML-1450	16, 32, 100, 106

was then projected horizontally to produce the signal shown in Figure 2.2(a). The discrete Fourier transform of the projection, shown in Figure 2.2(b), shows peaks close to 132 cycles/inch and 150 cycles/inch which are the printer's principle banding frequencies. Table 2.1 is a list of printers and their principle banding frequencies as found by this method.

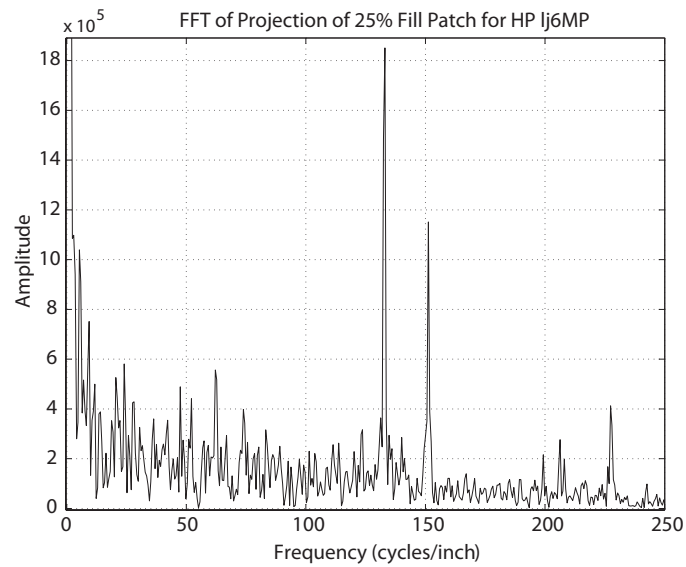
2.2 Texture Features

Detection and measurement of the banding signal in documents with large midtone regions, such as those with graphic art, can easily be performed by using methods similar to that used in Figure 2.2. However, detecting the intrinsic signature in small printed regions such as text requires a different processing technique due to the limited number of cycles of the banding signal which can be captured in the length of any one text character. For a 600 DPI print process, a 12 point character will have a maximum height of approximately 100 printer pixels or 1/6inch.

Intrinsic characterization of EP printers from printed text documents requires a feature set that can be obtained from individual printed characters. Features are obtained from an "image" of a document generated by scanning a text document at a



(a)



(b)

Figure 2.2. (a) Projection and spectral analysis of a 25% fill pattern printed on an HP LaserJet 6MP. (b) Spectrum of the projection (FFT) showing peaks at 132 and 150 cycles/inch.

sufficiently high resolution. Each character in the image is relatively small, typically only 200×200 pixels for 12 point text scanned at 2400 dpi, and is non-convex. These two properties make it difficult to perform meaningful filtering operations in either the pixel or transformed pixel domain if the area of interest is the printed region of each character.

Instead of attempting to estimate banding frequencies, the toner density fluctuations within printed areas of the document can be viewed as texture and modeled using graylevel co-occurrence texture features [48,49]. Graylevel co-occurrence texture features assume that the texture information in an image is contained in the overall spatial relationships among the pixels in the image. The graylevel co-occurrence matrix (GLCM) captures these pixel relationships within the printed regions of the page by providing an estimate of the second order probability density function of the pixels in the image. Features are then obtained as statistics of the GLCM which can then be used for classification.

Texture variation in a document is assumed to be predominantly in the process direction [50,51]. Figure 2.3 shows an idealized character, $I(i,j)$, from which features are extracted. The region of interest (ROI) is the set of all pixels within the printed area of the character. The determination of this region involves morphological filtering, specifically a morphological opening operation to close any holes inside the character and remove any noise outside the character due to toner scatter or the paper texture [14,52].

The set of pixels contained within the ROI, also called the support of the ROI, is denoted by the set Ω . The GLCM, defined as

$$C(n, m) = \sum_{(i,j),(i+dr,j+dc) \in \Omega} 1_{\{I(i,j)=n, I(i+dr,j+dc)=m\}}, \quad (2.1)$$

has entries $C(n, m)$ which are equal to the number of occurrences of pixels with graylevels n and m respectively with a separation of (dr, dc) pixels (see Figure 2.4). If the GLCM is normalized with respect to the number of pixels in the ROI, denoted

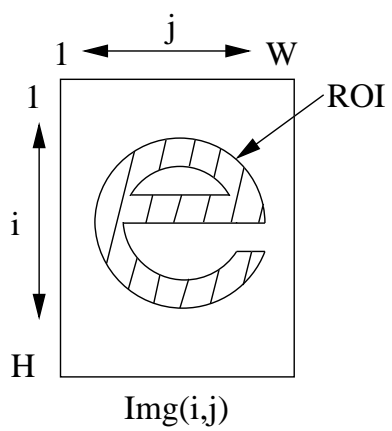


Figure 2.3. Idealized text character.

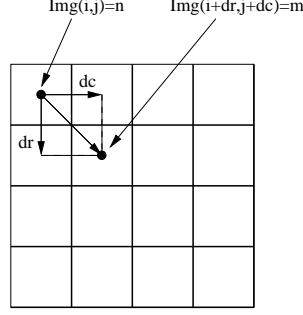


Figure 2.4. Generation of $C(n, m)$.

by $|\Omega|$, its entries then represent the probability of occurrence of pixel pairs with graylevels n and m with separation (dr, dc) and is defined as

$$p_{glcm}(n, m) = \frac{1}{|\Omega|} C(n, m). \quad (2.2)$$

To clarify how the graylevel co-occurrence matrix is obtained, the following examples are presented. For the first example, refer to the image represented in Figure 2.5. Each grid location represents a pixel with image coordinates (r, c) (row, column) in the image with the non-shaded region representing the region of interest. Furthermore, the pixels in these example images take values $I(r, c)$ from 1 to 5 such that the GLCM is a 5×5 matrix. To obtain the GLCM for this image, values for dr and dc must first be selected. For this example let $dr = 1$ and $dc = 0$. These values for (dr, dc) indicate that vertically adjacent pixel pairs are used to construct the GLCM. Start by initializing the GLCM to a 5×5 matrix of zeros. For each pixel (r, c) in the ROI, check if the pixel $(r + dr, c + dc)$ is also within the ROI. If $(r + dr, c + dc)$ is within the ROI, then add 1 to the $(I(r, c), I(r + dr, c + dc))$ th element of the GLCM. Continue the process with the remaining pixels in the ROI.

For Example 1, starting with the top-left pixel within the ROI at index $(1, 1)$ (row 1, column 1), first check if the pixel $(1 + dr, 1 + dc) = (2, 1)$ is also within the ROI. In this case the pixel $(2, 1)$ is within the ROI and the pixel values $I(1, 1) = 4$ and $I(2, 1) = 4$. The $(I(1, 1), I(2, 1)) = (4, 4)$ th element of the GLCM is therefore incremented by 1. Similarly for the pixel at $(1, 2)$, $I(1, 2) = 4$ and $I(1 + dr, 2 + dr) =$

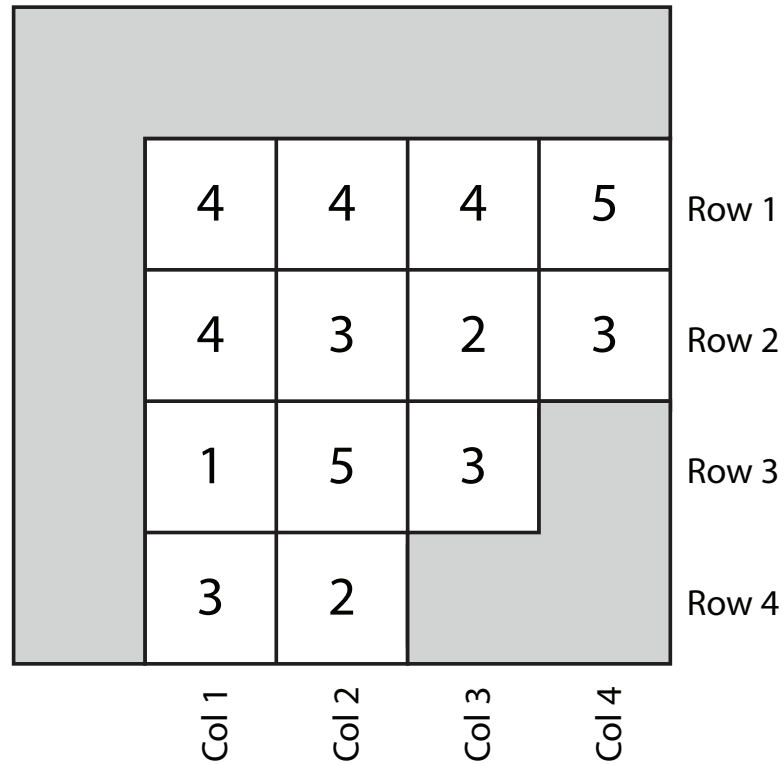


Figure 2.5. Image for GLCM Example 1. The non-shaded region is the region of interest from which the GLCM is obtained. Each number represents a graylevel pixel value between 1 and 5.

For the second example, consider the image represented in Figure 2.6. In this image, the pixel values are structured to represent banding in the vertical direction with a normalized frequency of 0.125 cycles/pixel (0.5 cycles/pixel would correspond to horizontal black and white stripes alternating every row). The GLCM for this image can be obtained in the same manner as for Example 1, and doing so with $(dr, dc) = (1, 0)$ results in

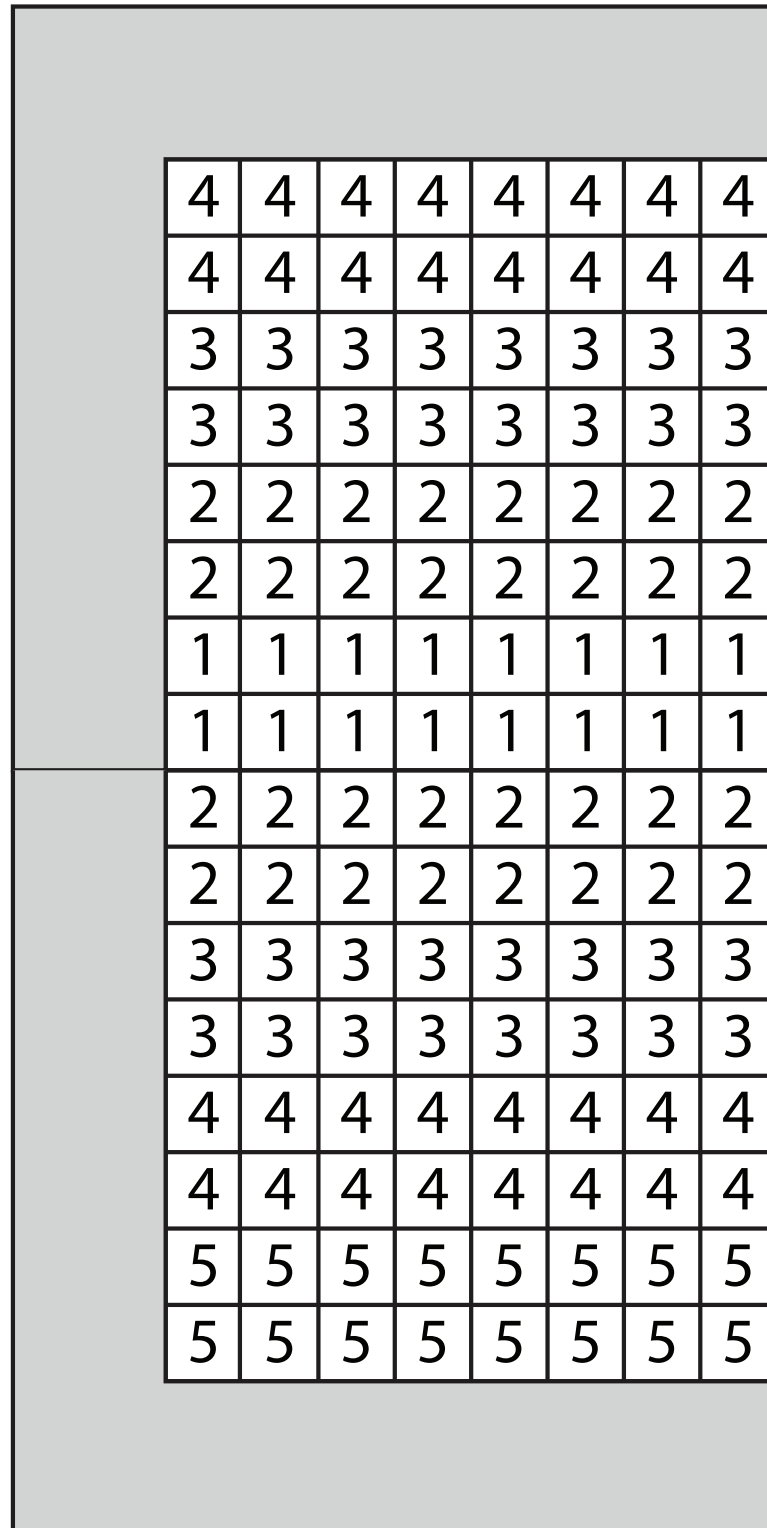
$$C = \begin{bmatrix} 0 & 8 & 0 & 0 & 0 \\ 8 & 0 & 8 & 0 & 0 \\ 0 & 8 & 0 & 8 & 0 \\ 0 & 0 & 8 & 0 & 8 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (2.4)$$

Note the diagonal symmetry and zeros on the diagonal. If the GLCM is obtained instead for horizontal pixel pairs, $(dr, dc) = (0, 1)$,

$$C = \begin{bmatrix} 7 & 0 & 0 & 0 & 0 \\ 0 & 14 & 0 & 0 & 0 \\ 0 & 0 & 14 & 0 & 0 \\ 0 & 0 & 0 & 14 & 0 \\ 0 & 0 & 0 & 0 & 7 \end{bmatrix}. \quad (2.5)$$

Note that the only non-zero elements in this case are on the diagonal, indicating that there are no pixel variations in the horizontal direction. In both cases above, the first order statistics of the image (graylevel histogram) can be obtained simply by summing the elements in each row or column and are independent of the second order statistics of the image.

For the third example, consider the image represented in Figure 2.7. In this image, the pixel values are structured to represent banding in the vertical direction with a normalized frequency of 0.0625 cycles/pixel. The GLCM for this image can



be obtained in the same manner as for Example 2, and doing so with $(dr, dc) = (1, 0)$ results in

$$C = \begin{bmatrix} 8 & 8 & 0 & 0 & 0 \\ 8 & 16 & 8 & 0 & 0 \\ 0 & 8 & 16 & 8 & 0 \\ 0 & 0 & 8 & 16 & 8 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (2.6)$$

Note that this GLCM is similar to the one obtained in Example 2, with the exception that the diagonal elements are now non-zero. In general, if the marginal probabilities of the GLCM have a high variance (non-zero elements far away from the diagonal) then the underlying image contains high frequency content in the direction (dr, dc) . If the marginal probabilities of the GLCM have a low variance (non-zero elements close to the diagonal) then the underlying image contains low frequency content in the direction (dr, dc) .

For the fourth example, consider the image represented in Figure 2.8. In this image, the pixel values are structured to represent banding in the vertical direction with a normalized frequency of 0.0625 cycles/pixel, as was the case for the image in Figure 2.6, however now the region of interest is non-rectangular.. The GLCM for this image can be obtained in the same manner as for Example 2, and doing so with $(dr, dc) = (1, 0)$ results in

$$C = \begin{bmatrix} 0 & 5 & 0 & 0 & 0 \\ 8 & 0 & 5 & 0 & 0 \\ 0 & 8 & 0 & 4 & 0 \\ 0 & 0 & 8 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (2.7)$$

Note that the structure of the GLCM is similar to the one obtained for Figure 2.6 with the exception that several elements of the matrix have decreased in value. In general, the shape of the ROI will affect the GLCM.

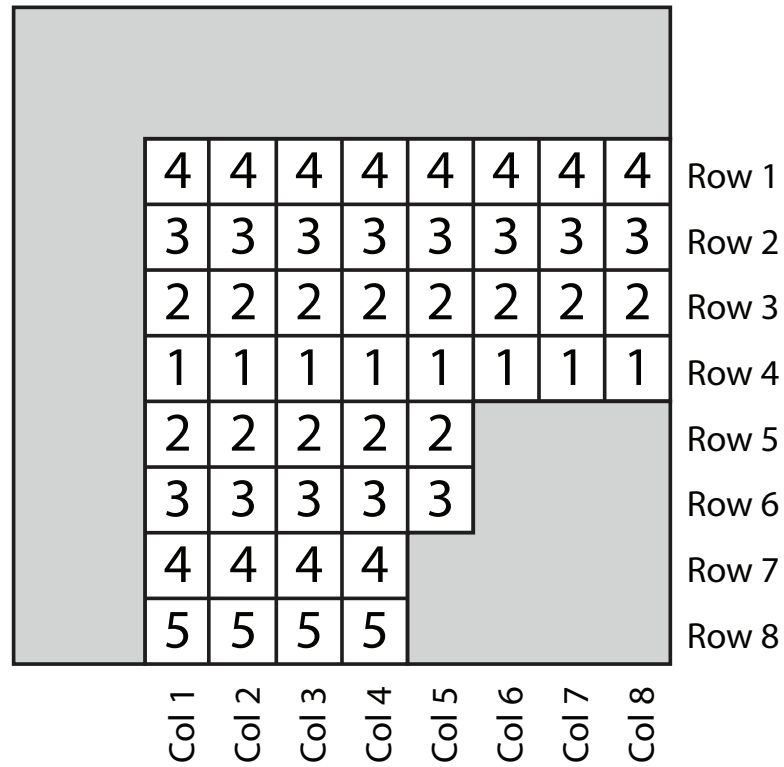


Figure 2.8. Image for GLCM Example 4. The non-shaded region is the region of interest from which the GLCM is obtained. Each number represents a graylevel pixel value between 1 and 5. This image has a structure imposed upon the pixel values that simulates banding in the vertical direction similar to the image in Figure 2.6, however the region of interest in this case is not rectangular.

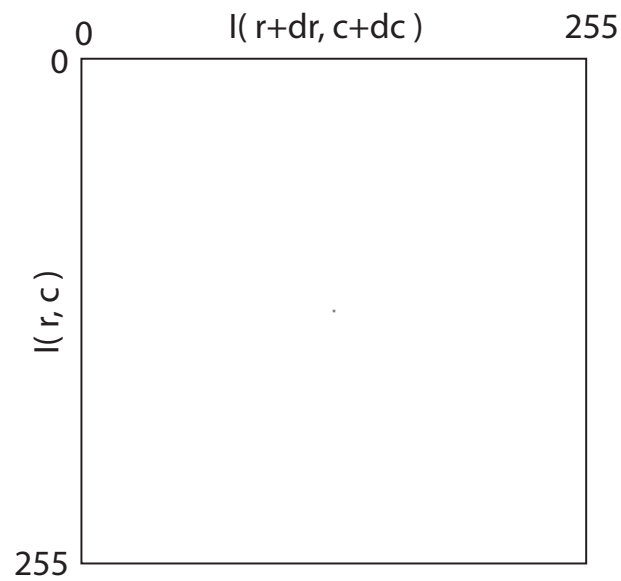
Several more examples of 8-bit grayscale images and their corresponding graylevel co-occurrence matrices are shown in Figure 2.9-2.12. The graylevel variation in the images follow a sinusoidal pattern

$$I(r, c) = 64 \sin(2\pi f \frac{r}{R_p}), \quad (2.8)$$

where f is a frequency in units of cycles/inch and $R_p = 600$ DPI is selected as the image resolution. Note how the shape of the GLCM changes as the frequency content in the image moves from 0 cycles/pixel (0 cycles/inch) toward the Nyquist rate 0.5 cycles/pixel (300 cycles/inch). Graylevel co-occurrence matrices for the same images with added Gaussian noise (mean 0 and variance 25.5) are shown in Figures 2.13-2.16.



(a)

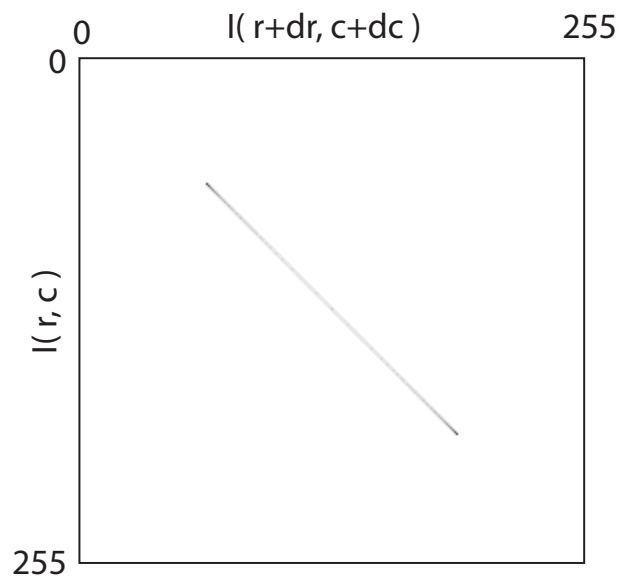


(b)

Figure 2.9. Graylevel co-occurrence Example 5. Subfigure (a) shows a 600×600 pixel 8-bit grayscale image with uniform graylevel. Subfigure (b) shows the corresponding graylevel co-occurrence matrix where darker pixels indicate larger values in the GLCM.

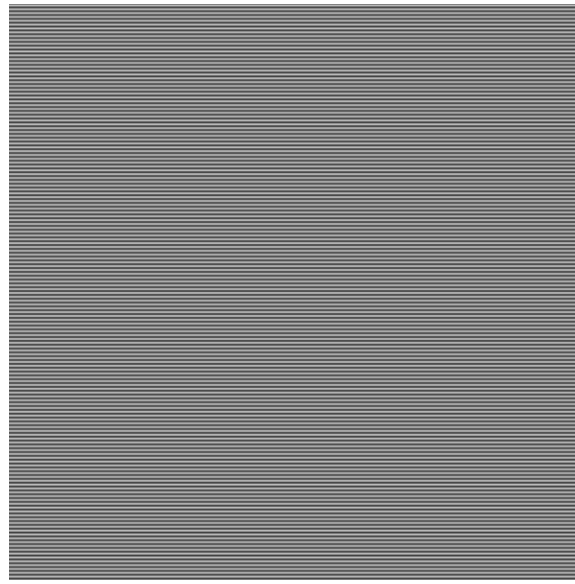


(a)

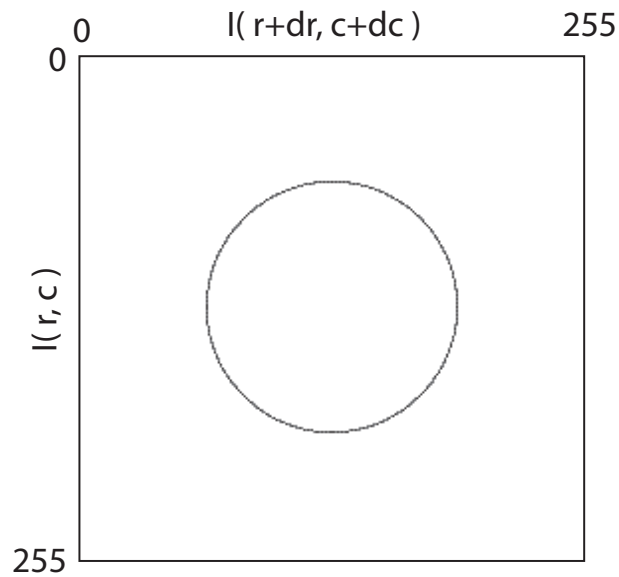


(b)

Figure 2.10. Graylevel co-occurrence Example 6. Subfigure (a) shows a 600×600 pixel 8-bit grayscale image with a 0.0017 cycle/pixel sinusoidal variation in the vertical direction representing 1 cycle/inch banding in a 600 DPI print process. Subfigure (b) shows the corresponding graylevel co-occurrence matrix where darker pixels indicate larger values in the GLCM.

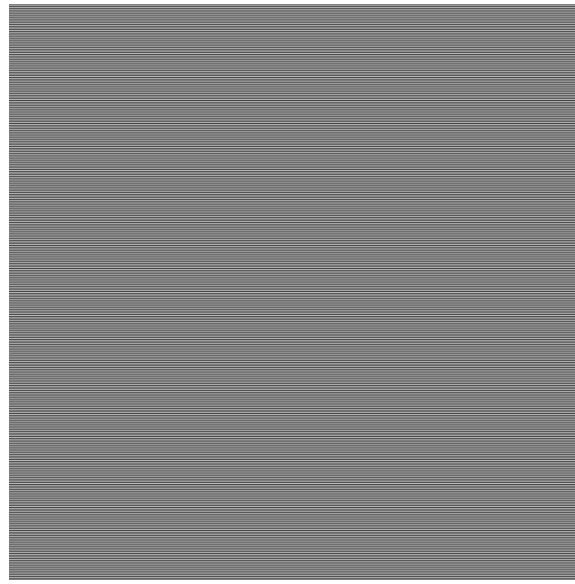


(a)

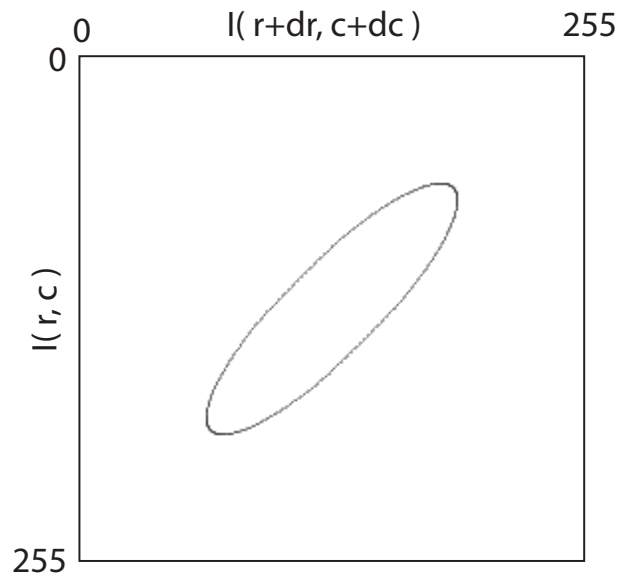


(b)

Figure 2.11. Graylevel co-occurrence Example 7. Subfigure (a) shows a 600×600 pixel 8-bit grayscale image with a 0.250 cycle/pixel sinusoidal variation in the vertical direction representing 150 cycle/inch banding in a 600 DPI print process. Subfigure (b) shows the corresponding graylevel co-occurrence matrix where darker pixels indicate larger values in the GLCM.

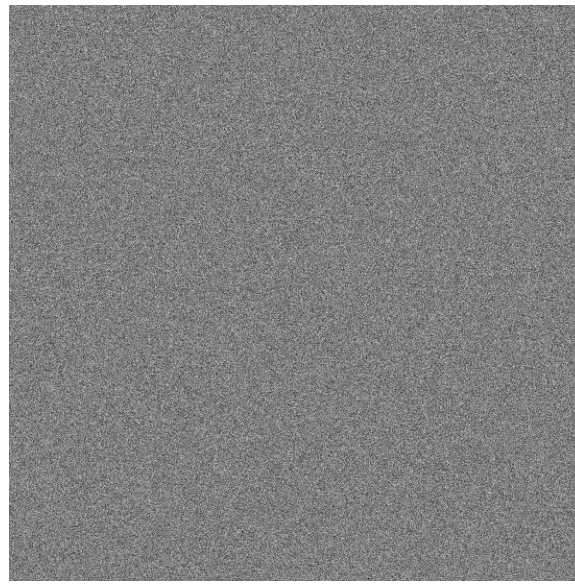


(a)

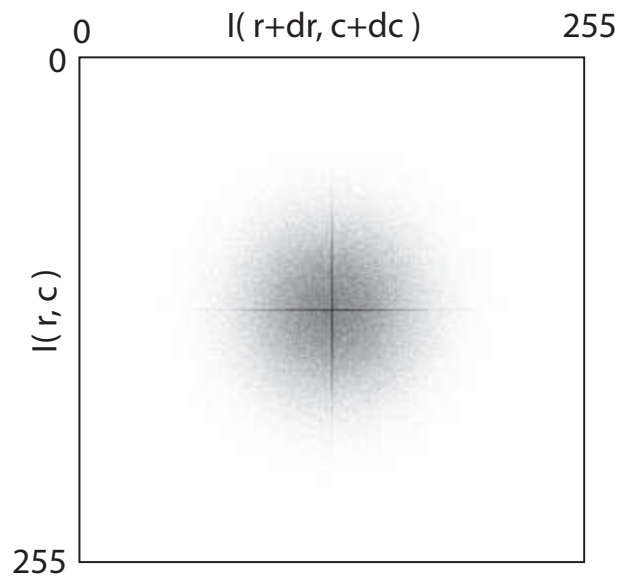


(b)

Figure 2.12. Graylevel co-occurrence Example 8. Subfigure (a) shows a 600×600 pixel 8-bit grayscale image with a 0.4167 cycle/pixel sinusoidal variation in the vertical direction representing 250 cycle/inch banding in a 600 DPI print process. Subfigure (b) shows the corresponding graylevel co-occurrence matrix where darker pixels indicate larger values in the GLCM.

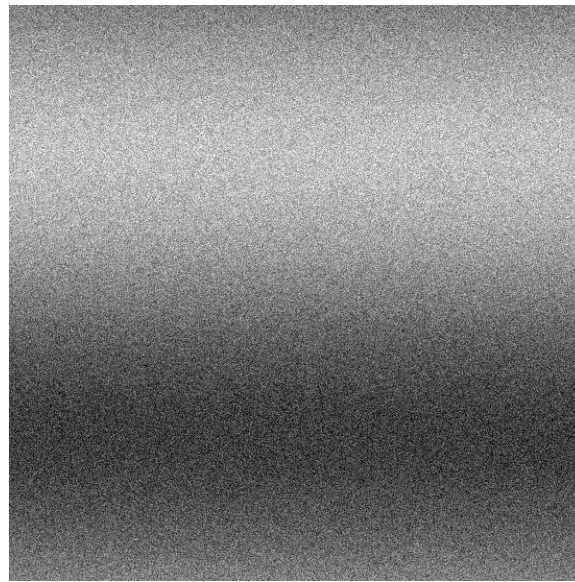


(a)

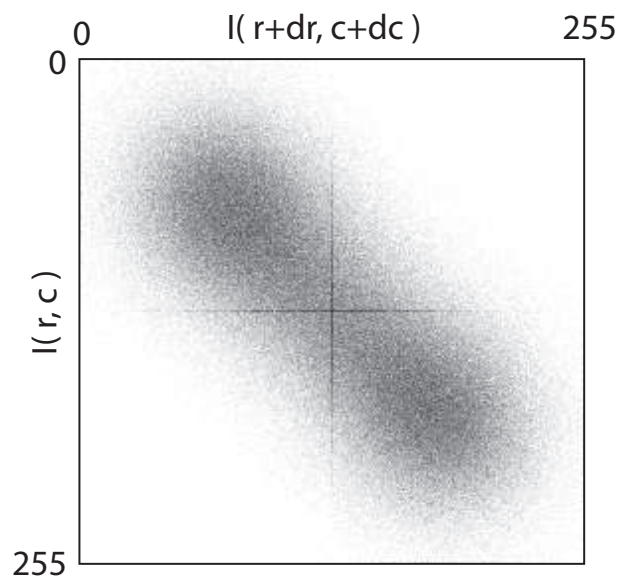


(b)

Figure 2.13. Graylevel co-occurrence Example 5 with Gaussian noise. Subfigure (a) shows a 600×600 pixel 8-bit grayscale image with uniform graylevel, plus Gaussian noise. Subfigure (b) shows the corresponding graylevel co-occurrence matrix where darker pixels indicate larger values in the GLCM.

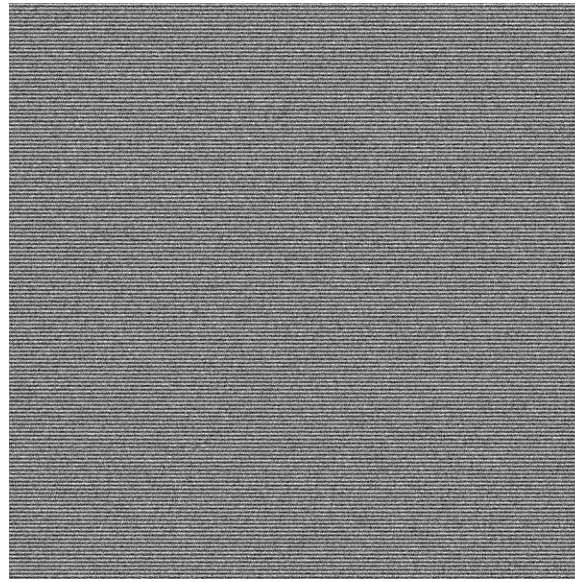


(a)

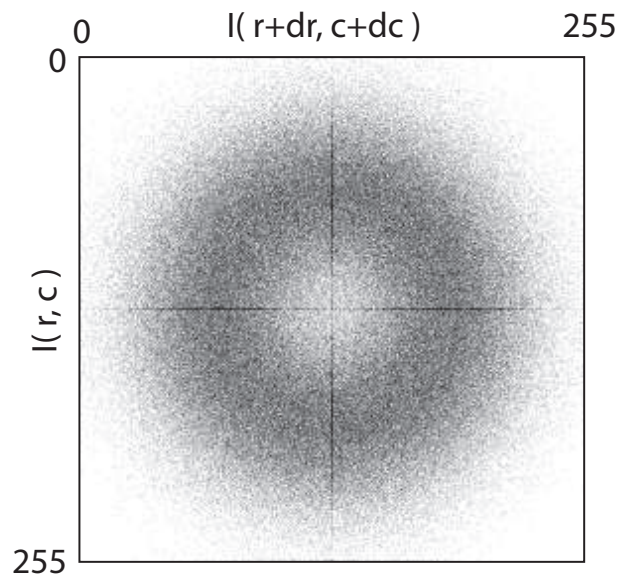


(b)

Figure 2.14. Graylevel co-occurrence Example 6 with gaussian noise. Subfigure (a) shows a 600×600 pixel 8-bit grayscale image with a 0.0017 cycle/pixel sinusoidal variation in the vertical direction, representing 1 cycle/inch banding in a 600 DPI print process, plus Gaussian noise. Subfigure (b) shows the corresponding graylevel co-occurrence matrix where darker pixels indicate larger values in the GLCM.

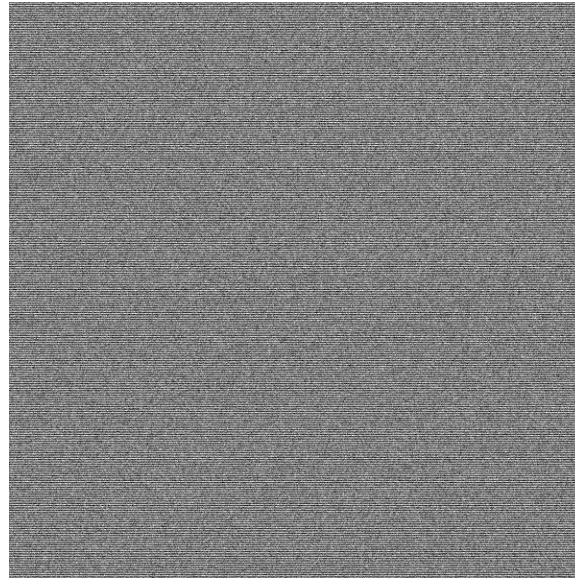


(a)

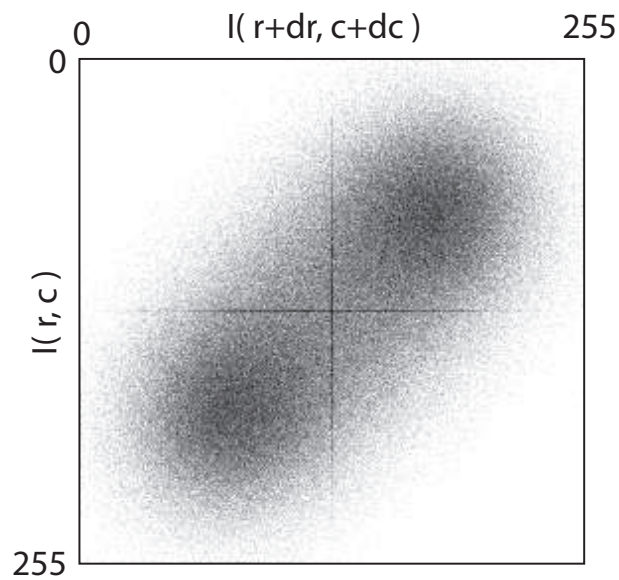


(b)

Figure 2.15. Graylevel co-occurrence Example 7 with Gaussian noise. Subfigure (a) shows a 600×600 pixel 8-bit grayscale image with a 0.250 cycle/pixel sinusoidal variation in the vertical direction, representing 150 cycle/inch banding in a 600 DPI print process, plus Gaussian noise. Subfigure (b) shows the corresponding graylevel co-occurrence matrix where darker pixels indicate larger values in the GLCM.



(a)



(b)

Figure 2.16. Graylevel co-occurrence Example 8 with Gaussian noise. Subfigure (a) shows a 600×600 pixel 8-bit grayscale image with a 0.4167 cycle/pixel sinusoidal variation in the vertical direction, representing 250 cycle/inch banding in a 600 DPI print process, plus Gaussian noise. Subfigure (b) shows the corresponding graylevel co-occurrence matrix where darker pixels indicate larger values in the GLCM.

Twenty features are obtained from the GLCM based on earlier work in [53] and originally proposed in [48]. Many of these features are standard statistical measures that are used to describe the distribution of non-zero elements within the GLCM, while others were added over the years after it was found that they provided better discrimination between certain classes of textures [48, 49, 54, 55]. When a features has an intuitive representation from a perceptual point of view it will be described below.

The first four features are the marginal means and variances defined by

$$\mu_r = \frac{1}{256} \sum_{n=0}^{255} n p_r(n), \quad (2.9)$$

$$\mu_c = \frac{1}{256} \sum_{m=0}^{255} m p_c(m), \quad (2.10)$$

$$\sigma_r^2 = \sum_{n=0}^{255} n^2 p_r(n) - \mu_r^2, \quad (2.11)$$

$$\sigma_c^2 = \sum_{m=0}^{255} m^2 p_c(m) - \mu_c^2, \quad (2.12)$$

which are obtained from the marginal probability densities

$$p_r(n) = \sum_{m=0}^{255} p_{glcm}(n, m), \quad (2.13)$$

and

$$p_c(m) = \sum_{n=0}^{255} p_{glcm}(n, m). \quad (2.14)$$

These features describe the distribution of the non-zero elements within the GLCM.

The next seven features are described by Equations 2.15-2.21.

$$E = \sum_{n=0}^{255} \sum_{m=0}^{255} p_{glcm}^2(n, m), \quad (2.15)$$

is the energy of the normalized GLCM; this value becomes larger as the GLCM becomes sparsely populated (fewer non-zero elements). The energy will be greatest when there is little to no graylevel variation in the image. Three entropy measurements

$$h_{xy1} = - \sum_{n=0}^{255} \sum_{m=0}^{255} p_{glcm}(n, m) \log_2(p_r(n)p_c(m)), \quad (2.16)$$

$$h_{xy2} = - \sum_{n=0}^{255} \sum_{m=0}^{255} p_r(n) p_c(m) \log_2(p_r(n) p_c(m)), \quad (2.17)$$

$$h_C = - \sum_{n=0}^{255} \sum_{m=0}^{255} p_{glcm}(n, m) \log_2 p_{glcm}(n, m), \quad (2.18)$$

describe the “randomness” of the GLCM and are larger if the GLCM has more uniformly distributed non-zero values (lots of graylevel variation in the image), and reaches a minimum when the GLCM is sparsely populated (little to no graylevel variation in the image). The maximum entry in the GLCM is

$$p_{glcm, max} = \max_{n, m} \{p_{glcm}(n, m)\}, \quad (2.19)$$

and

$$\rho = \sum_{n=0}^{255} \sum_{m=0}^{255} \frac{(n - \mu_r)(m - \mu_c) p_{glcm}(n, m)}{\sigma_r \sigma_c}, \quad (2.20)$$

$$\rho_d = \sum_{n=0}^{255} \sum_{m=0}^{255} |n - m| (n + m - \mu_r - \mu_c) p_C(n, m), \quad (2.21)$$

are two correlation metrics that capture linear dependencies within the image [48].

A similar set of four features, Equations 2.23-2.26, are obtained from the difference histogram defined by

$$D(k) = \sum_{\substack{0 \leq n < 255 \\ 0 \leq m < 255 \\ |n-m|=k}} p_{glcm}(n, m). \quad (2.22)$$

The energy and entropy of $D(k)$ are defined as

$$E_D = \sum_{k=0}^{255} D(k), \quad (2.23)$$

$$h_D = - \sum_{k=0}^{255} D(k) \log_2 D(k). \quad (2.24)$$

The inertia

$$I_D = \sum_{k=0}^{255} k^2 D(k), \quad (2.25)$$

can be thought of as a measure of contrast between pixels (dr, dc) apart. The homogeneity

$$L_D = \sum_{k=0}^{255} \frac{D(k)}{1 + k^2}, \quad (2.26)$$

describes how close the elements of the GLCM are to the diagonal.

The last five features, Equations 2.29-2.33, are obtained from the sum histogram defined as

$$S(k) = \sum_{\substack{0 \leq n \leq 255 \\ 0 \leq m \leq 255 \\ n+m=k}} p_{glcm}(n, m). \quad (2.27)$$

They are the energy, entropy, variance, cluster shade, and cluster prominence of $S(k)$, respectively defined as

$$\mu_S = \sum_{k=0}^{510} k S(k), \quad (2.28)$$

$$E_S = \sum_{k=0}^{510} S(k), \quad (2.29)$$

$$h_S = - \sum_{k=0}^{510} S(k) \log_2 S(k), \quad (2.30)$$

$$\sigma_S^2 = \sum_{k=0}^{510} (k - \mu_S)^2 S(k), \quad (2.31)$$

$$A_D = \sum_{k=0}^{510} \frac{(k - \mu_r - \mu_c)^3 S(k)}{(\sigma_r^2 - \sigma_c^2 + 2r\sigma_r\sigma_c)^{\frac{3}{2}}}, \quad (2.32)$$

$$B_D = \sum_{k=0}^{510} \frac{(k - \mu_r - \mu_c)^4 S(k)}{(\sigma_r^2 - \sigma_c^2 + 2r\sigma_r\sigma_c)^2}. \quad (2.33)$$

The cluster shade and cluster prominence “are believed to gauge the perceptual concepts of uniformity and proximity” [49, 56, 57].

In addition to the 20 GLCM features above, two simple features are also included; these are the variance

$$\sigma_I^2 = \frac{1}{|\Omega|} \sum_{(i,j) \in \Omega} (I(i, j) - \mu_I)^2, \quad (2.34)$$

and entropy

$$h_I = - \sum_{\alpha=0}^{255} p_I(\alpha) \log_2 p_I(\alpha), \quad (2.35)$$

of the pixel values in the ROI where

$$\mu_I = \frac{1}{|\Omega|} \sum_{(i,j) \in \Omega} I(i, j), \quad (2.36)$$

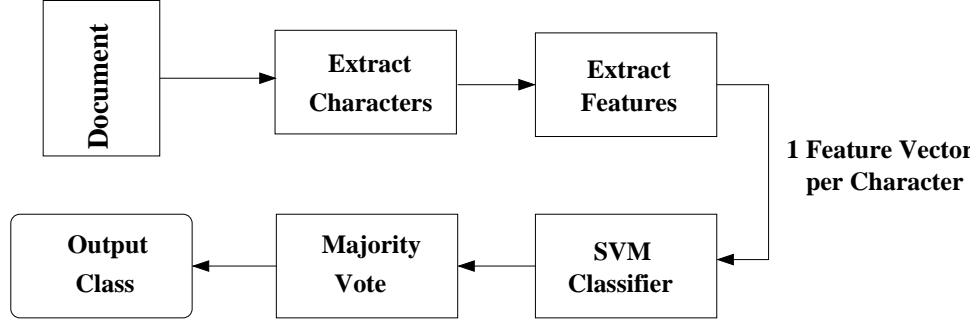


Figure 2.17. System diagram of intrinsic printer identification scheme using texture based features.

and

$$p_I(\alpha) = \frac{1}{|\Omega|} \sum_{(i,j) \in \Omega} 1_{\{I(i,j)=\alpha\}}. \quad (2.37)$$

2.2.1 System for Intrinsic Printer Identification

The forensic printer identification process using texture based features is shown in Figure 2.17. A random document generator, known as the Forensic Monkey Text Generator (FMTG), is used to generate full page text documents with known statistics which are used to test the system. The FMTG is a modified version of a Markov chain random text generator (the “Monkey” [58]) which is trained on text freely available from Project Gutenberg [59]. The FMTG first estimates the conditional probabilities of all the characters in the training data from order 0 to order 4. It then generates a user specified length of text following these transition probabilities and produces \LaTeX formatted output to allow easy formatting and conversion into PostScript (PS) and other page description languages.

The first step is to scan the document at a sufficiently high resolution. Next all the letter “e”s in the document are extracted. The reason for this is that “e” is the most frequently occurring character in the English language. A set of features are extracted from each character forming a feature vector for each letter “e” in the document.

Each feature vector is then individually classified using a 5-nearest-neighbor (5NN) classifier [60] or a support vector machine (SVM) classifier [61].

Let Ψ be the set of all printers $\{\alpha_1, \alpha_2, \dots, \alpha_n\}$. For any $\phi \in \Psi$, let $c(\phi)$ be the number of “e”s classified as being printed by printer ϕ . The final classification is decided by selecting $\phi = \arg \max_{\phi} c(\phi)$. In other words, a majority vote is performed on the resulting classifications from the SVM.

A support vector machine classifier is a type of binary classifier. A detailed description of SVM classification, which is summarized below, can be found in [62] and [63]. In its most basic form it is a linear binary classifier that finds an optimal separating hyperplane between two classes of data. Consider the data shown in Figure 2.18. In this case the data is separable by a single linear hyperplane, (\mathbf{w}, b) , giving the standard decision rule for a linear classifier

$$h(x) = \text{sgn}(\langle \mathbf{w}, \mathbf{x} \rangle + b), \quad (2.38)$$

where $\langle \mathbf{a}, \mathbf{b} \rangle$ denotes the inner product between vectors a and b , and $h(x) \in -1, 1$ with value -1 denoting class ‘X’ and value 1 denoting class ‘O’. The quantities γ_1 and γ_2 are the distances from the separating hyperplane to the nearest data points in each class. It is assumed that (\mathbf{w}, b) is found such that $\gamma_1 = \gamma_2 = \gamma$. γ is then called the margin of the training set.

A SVM chooses \mathbf{w} to maximize the margin between both classes. Specifically, the following problem is optimized:

$$\begin{aligned} & \underset{\alpha}{\text{minimize}} && \langle \mathbf{w}, \mathbf{w} \rangle, \\ & \text{subject to} && y_i (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1, \\ & && i = 1, \dots, l, \end{aligned} \quad (2.39)$$

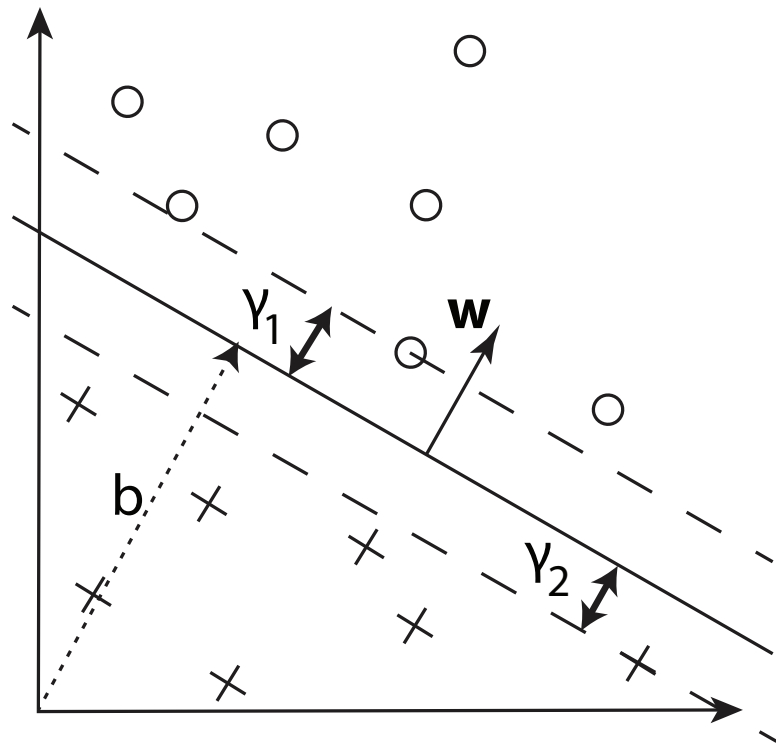


Figure 2.18. Binary classification problem solved using a single linear hyperplane.

where l is the number of training vectors or data points with known class labels used to construct the SVM. The dual of this problem is

$$\begin{aligned} \underset{\alpha}{\text{maximize}} \quad & W(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l y_i y_j \alpha_i \alpha_j < \mathbf{x}_i, \mathbf{x}_j >, \\ \text{subject to} \quad & \sum_{i=1}^l y_i \alpha_i = 0, \\ & \alpha_i \geq 0, \quad i = 1, \dots, l. \end{aligned} \tag{2.40}$$

The optimal \mathbf{w} is obtained through linear optimization with Lagrange multipliers α_i (see [62] for details), and the optimal hyperplane turns out to be

$$f(x) = \sum_{i \in SV} y_i \alpha_i < \mathbf{x}_i, \mathbf{x} > + b, \tag{2.41}$$

where $y_i \in -1, 1$ is the class label and α_i is the Lagrange multiplier corresponding to training vector \mathbf{x}_i . The set $SV = \{i : \alpha_i \neq 0\}$ is the set of training vectors closest to the decision boundary, also called the support vectors. The same decision boundary would be obtained if the training set was trimmed to only include the support vectors. The decision rule now becomes

$$h(x) = \text{sgn}(f(x)) = \text{sgn}\left(\sum_{i \in SV} y_i \alpha_i < \mathbf{x}_i, \mathbf{x} > + b\right). \tag{2.42}$$

Datasets in the real world are typically not linearly separable. SVM addresses this issue by first projecting the data into a high dimensional feature space in which they are linearly separable. The mapping of data point \mathbf{x} into this feature space is denoted by the operator

$$\phi(\mathbf{x}) : \mathbf{X} \rightarrow \mathbb{R}^m. \tag{2.43}$$

Solving the same optimization described above using the mapping ϕ for all feature points, the separating hyperplane becomes

$$f(x) = \sum_{i \in SV} y_i \alpha_i < \phi(\mathbf{x}_i), \phi(\mathbf{x}) > + b, \tag{2.44}$$

or alternatively

$$f(x) = \sum_{i \in SV} y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b, \tag{2.45}$$

where

$$K(\mathbf{x}_i, \mathbf{x}) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}) \rangle. \quad (2.46)$$

The function $K(\mathbf{x}_i, \mathbf{x})$ is called a kernel. Typically ϕ is not chosen explicitly or even known. Instead the kernel K is chosen such that it satisfies Mercer's Conditions [62]. By satisfying these conditions, Mercer's theorem states that a function $K(\mathbf{x}, \mathbf{z})$, symmetric on \mathbf{x} , is a dot product in some high dimensional space. One widely used kernel is the Gaussian radial basis function defined as

$$K(\mathbf{x}, \mathbf{z}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{z}\|^2}{\sigma^2}\right). \quad (2.47)$$

It turns out that this kernel function performs well for a wide variety of real world data.

Examples of the decision boundaries obtained using a SVM classifier are shown in the following two examples. The first example uses a linear SVM, that is $K(x, z) = \langle x, z \rangle$. Figure 2.19 shows two classes of data that are linearly separable with data points

$$\mathbf{x} = \begin{bmatrix} 1 & 2 & 2 & 1 & 3 & 4 & 3 & 4 \\ 1 & 1 & 2 & 2 & 3 & 3 & 4 & 4 \end{bmatrix}, \quad (2.48)$$

corresponding to class labels

$$y = [-1 \quad -1 \quad -1 \quad -1 \quad 1 \quad 1 \quad 1 \quad 1]. \quad (2.49)$$

The optimization problem in Equation 2.40 can be solved using \mathbf{x} as the set of training vectors with class labels y . Doing so gives the following Lagrange multipliers

$$\alpha = [0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0], \quad (2.50)$$

indicating that the third and fifth training vectors comprise the set of support vectors.

To write the decision rule, \mathbf{w} and b must first be obtained as

$$\mathbf{w} = \sum_{i=1}^l \alpha_i y_i \mathbf{x}_i = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad (2.51)$$

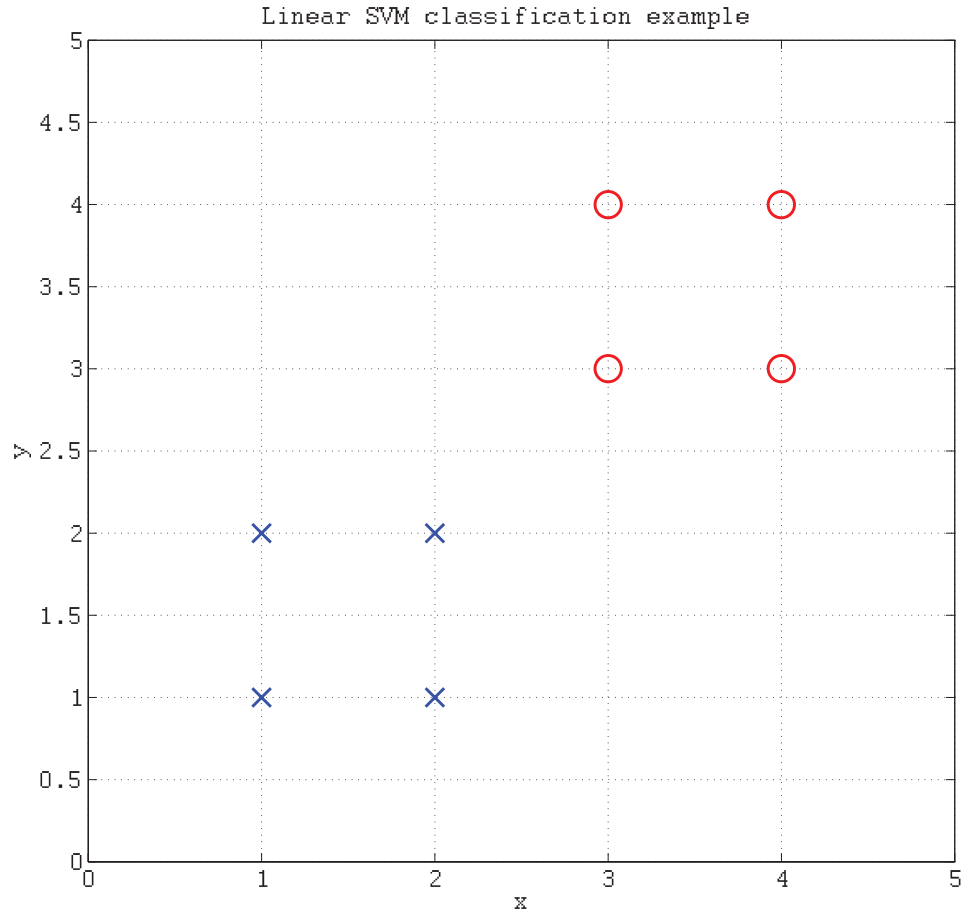


Figure 2.19. Linear support vector machine classification example. The two classes of data points represented by ‘X’ and ‘O’ are linearly separable.

and

$$b = \frac{1}{|SV|} \sum_{i \in SV} (y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle) = -2.5, \quad (2.52)$$

where SV is the set of support vectors and $|SV|$ is the number of support vectors which in this case is 2. The decision rule can then be written as

$$f(x) = \sum_{i \in SV} y_i \alpha_i \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}) \rangle - 2.5, \quad (2.53)$$

The result from this decision rule is shown in Figure 2.20. The support vectors are shown in bold.

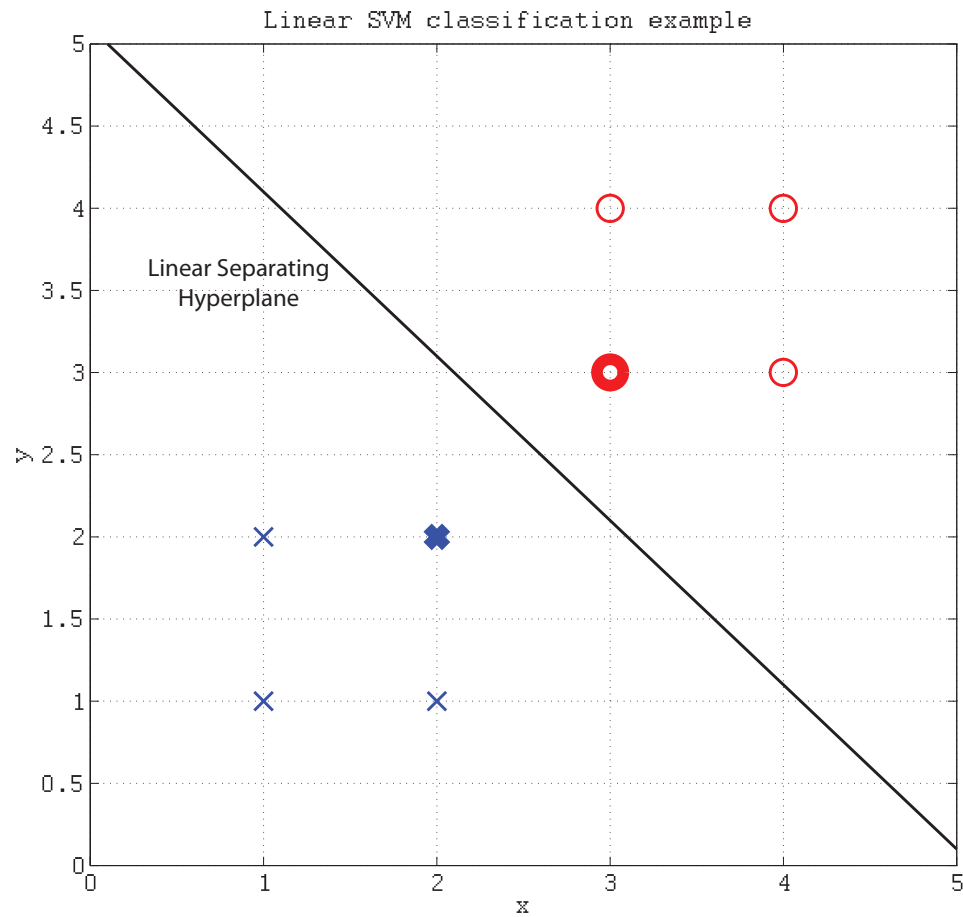


Figure 2.20. Linear support vector machine classification example. Two classes of data points represented by 'X' and 'O' are linearly separable in this case. The large 'X' and 'O' markers indicate the training data points, with those markers in bold indicating the two support vectors in this case. In this case there is a linear boundary separating the two training classes bisecting the line segment connecting the two support vectors.

The second example uses a non-linear SVM using the Gaussian radial basis function as the kernel, that is $K(x, z)$ is defined as shown in Equation 2.47. Figure 2.21 shows two classes of data that are not linearly separable. The optimization problem in Equation 2.40 is first modified by replacing all the inner products with the kernel $K(\mathbf{x}_i, \mathbf{x}_j)$, and then solved the same was as in the previous example. Doing so gives the following non-zero Lagrange multipliers

$$\alpha = \begin{bmatrix} 4.21 & 11.85 & 7.46 & 24.18 & 6.14 & 3.35 & 1.41 & 8.92 & 27.88 \end{bmatrix}, \quad (2.54)$$

corresponding to training vectors

$$\mathbf{x} = \begin{bmatrix} 5.82 & 2.87 & 7.26 & 2.79 & 7.73 & 6.65 & 3.74 & 3.16 & 2.41 \\ 7.28 & 6.20 & 5.55 & 5.42 & 5.00 & 7.96 & 7.17 & 7.01 & 5.35 \end{bmatrix}. \quad (2.55)$$

The original set of training vectors consisted of 41 training vectors, but only 9 of those data points, the support vectors, are necessary to construct the optimal decision boundary. To write the decision rule, \mathbf{w} and b must first be obtained as

$$\mathbf{w} = \sum_{i=1}^l \alpha_i y_i \mathbf{x}_i = \begin{bmatrix} -9.7316 \\ 2.4244 \end{bmatrix}, \quad (2.56)$$

and

$$b = \frac{1}{|SV|} \sum_{i \in SV} (y_i - K(\mathbf{w}, \mathbf{x}_i)) = \frac{1}{9}. \quad (2.57)$$

The decision rule can then be written as

$$f(x) = \sum_{i \in SV} y_i \alpha_i K(\phi(\mathbf{x}_i), \phi(\mathbf{x})) + \frac{1}{9}. \quad (2.58)$$

The result from this decision rule is shown in Figure 2.22. The support vectors are shown in bold. Finding a closed form solution for the decision boundary in this case is non-trivial, partially due to the fact that the mapping ϕ is unknown. Instead, points (a, b) , $0 < a < 10$, $0 < b < 10$, that lie closest to the decision boundary were searched for in increments of 0.01. These points are represented by black dots in Figure 2.22 and indicate the outline of the decision boundary.

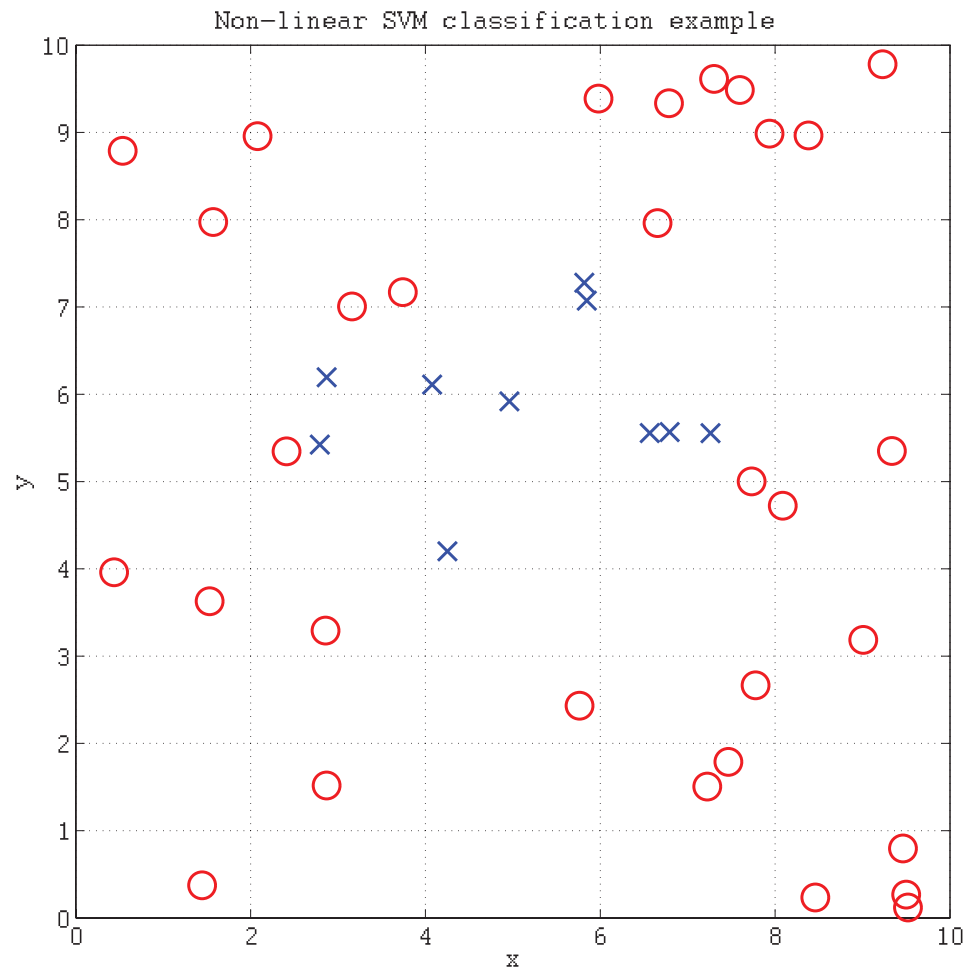


Figure 2.21. Non-linear support vector machine classification example. The two classes of data points represented by 'X' and 'O' are not linearly separable.

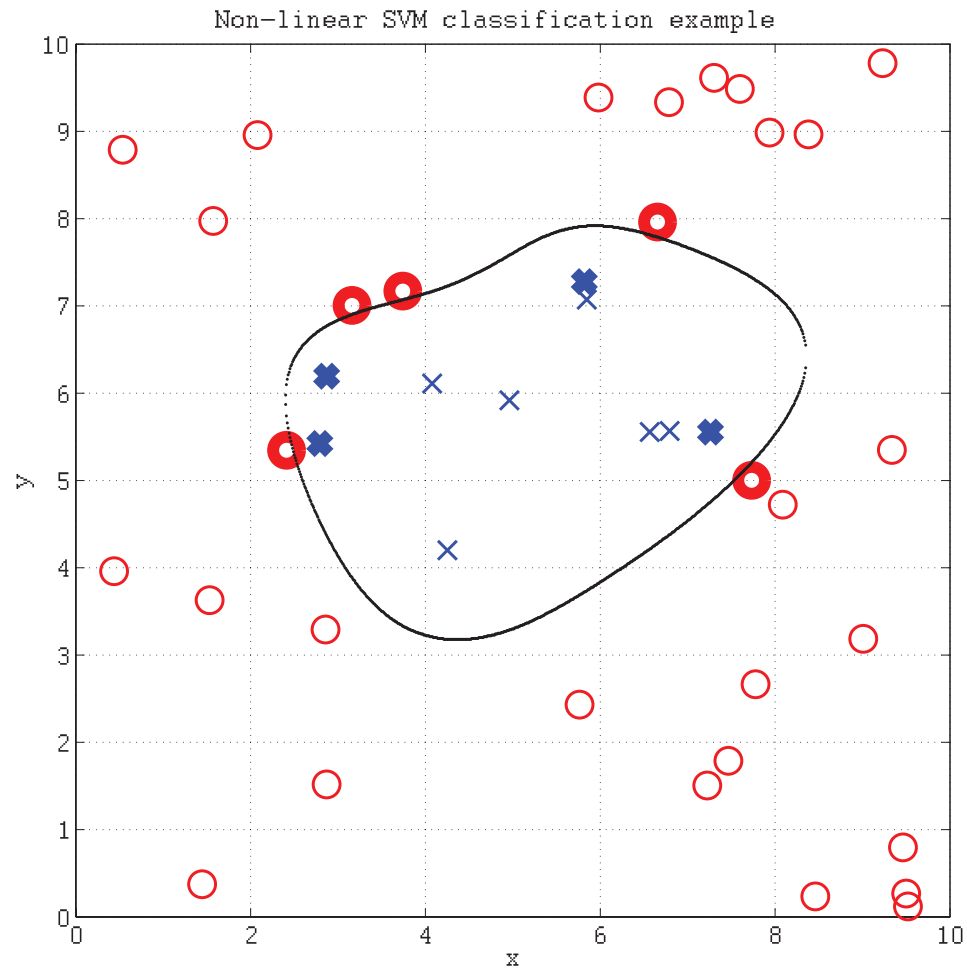


Figure 2.22. Non-linear support vector machine classification example. Two classes of data points represented by 'X' and 'O' are not linearly separable in this case. The large 'X' and 'O' markers indicate the training data points, with those markers in bold indicating the nine support vectors in this case. In this case there is a non-linear boundary separating the two training classes. Black dots mark the decision boundary obtained in this case.

Note that the SVM as described above is a binary classifier. In order to perform multi-class classification with more than two classes, multiple SVMs are generated between each pair of classes in a one-against-one approach [64]. For n classes there are $m = \binom{n}{2}$ unique pairs of classes, and therefore m SVMs are generated. To determine the class of a unknown vector \mathbf{x} , the decisions from all m classifiers are merged to determine the final output class.

Table 2.2. Printers used for classification.

Manufacturer	Model	DPI
Brother	hl1440	1200
HP	lj4050	600
Lexmark	e320	1200
HP	lj1000	600
HP	lj1200	600
HP	lj5M	600
HP	lj6MP	600
Minolta	1250W	1200
Okidata	14e	600
Samsung	ml1430	600
Samsung	ml1450	600

Using documents generated by the FMTG, an average of 458 “e”s per page is found for 12 point English text. Similarly, an average of 630 “e”s per page is found for 10 point English text. A test document containing 300 “e”s is used to test the classification technique. This document is printed on each printer listed in Table 2.2 and scanned in 8-bit grayscale at 2400dpi.

Since the texture is assumed to be varying primarily in the process direction, dc is set equal to 0 and only dr is set to a non-zero value in Equation 2.1 when generating the GLCM. To obtain the best value of dr , each scanned document was classified

using values of dr ranging from 1 to 10, providing 10 classification results for each document [14]. The system in this case used a 5NN classifier. 500 feature vectors per printer, generated independently of the feature vectors used for testing, were used to train the 5NN classifier. The percent correct classification after the 5NN classifier is plotted against dr in Figure 2.23. When using all 22 features, it was found that $dr = 2$ provides the best overall classification results [14]. The confusion matrix for this choice of dr is shown in Table 2.3. This specific type of matrix presents the classification results in a way that clearly shows how misclassifications (confusions) were made by the classifier. Each entry of the matrix is the number of “e”s out of the 300 in the test document which were classified as the printer listed at the heading of its column. For example, the first row of Table 2.3 shows that 207 “e”s printed by the Brother HL-1440 were classified correctly. The second highest number of “e”s were classified as being printed by the Minolta 1250W. A majority vote indicates that this document was most likely printed by the HL-1440, which is correct. If all the “e”s were classified correctly for each printer then only the diagonal elements of the confusion matrix would be non-zero.

Again let $c(\phi)$ be equal to the number of “e”s from any one unknown document classified as being printed by printer ϕ . Furthermore let $c(\phi_1)$ be the largest among all classes, and $c(\phi_2)$ be the second largest. Then it can be argued that the final classification resulting from the majority vote among the $c(\phi)$ has a higher confidence for larger ratios between $c(\phi_1)$ and $c(\phi_2)$. In this case, all the printers are classified correctly and with a relatively high confidence with the exception of the Okidata 14e, which is classified as being a Minolta 1250W.

The classification was repeated using 4 manually chosen features that showed large separation between classes. These features are σ_I^2 , h_I , μ_r , and E . From Figure 2.24 it is found that the best choice of dr in this case is 2. Scatter plots of these features show good separation between all ten printers and are shown in Figures 2.25 and 2.26. The HP LaserJet 4050 is a higher quality printer compared to the others in the test

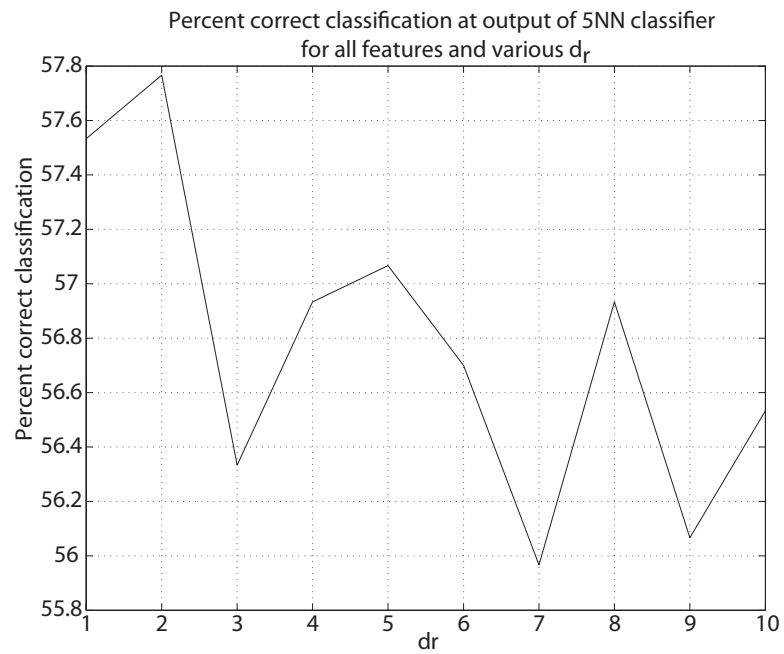


Figure 2.23. Percent correct classification of individual feature vectors versus d_r using all 22 features and 5NN classifier.

Table 2.3. Confusion matrix for GLCM based printer identification using 22 features, 5NN classifier, and $dr = 2$. (Blank indicates -0-)

train \ test	lj5m	lj6mp	lj1000	lj1200	E320	ml1430	lj4050	hl1440	1250w	14e	Output Class
lj5m	175	22	5	8		2		62	18	8	lj5m
lj6m	31	147	23	6	12	7		37	33	4	lj6m
lj1000	8	14	137	59	1	56		5	12	8	lj1000
lj1200	21	7	90	128	1	36		5	8	4	lj1200
E320				2	257			1	26	14	E320
ml1430	14	19	71	14	5	140		7	7	23	ml1430
lj4050							300				lj4050
hl1440	13	4	2		1	4		207	52	17	hl1440
1250w	8	1	2	1	40	3		34	179	32	1250w
14e	2	5	3	2	21	1		85	118	63	1250w

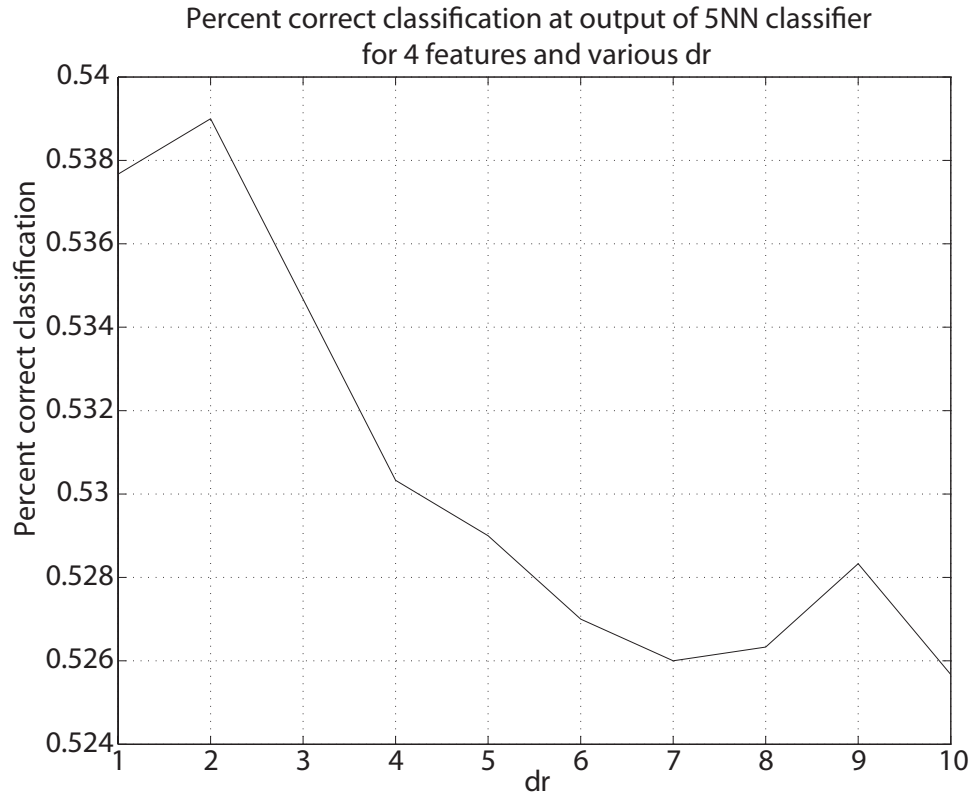


Figure 2.24. Percent correct classification of individual feature vectors versus dr using 4 selected features and 5NN classifier.

set, and the graylevel variance of its output is very low, moving all of its data points to the far left of Figure 2.25 off the graph.

The classification results using only these four features are shown in Table 2.4. All the printers are classified correctly with the exception of the HP LaserJet 1200 which is classified as an HP LaserJet 1000. This is reasonable since these two printers are mechanically similar. From these results it is concluded that the dimensionality of the original 22-dimensional feature vectors can be reduced while still preserving the capability to discriminate between the different printers. However, the confidence in the classification results using 4 features is in general lower than when using all the features. For example, the ratio between $c(\phi_1)$ and $c(\phi_2)$ for the Minolta 1250W when using 22 features is higher than in the 4 feature case. Similarly for the HP LaserJet 6MP, Minolta 1250W, and Okidata 14e, there is lower confidence in their all

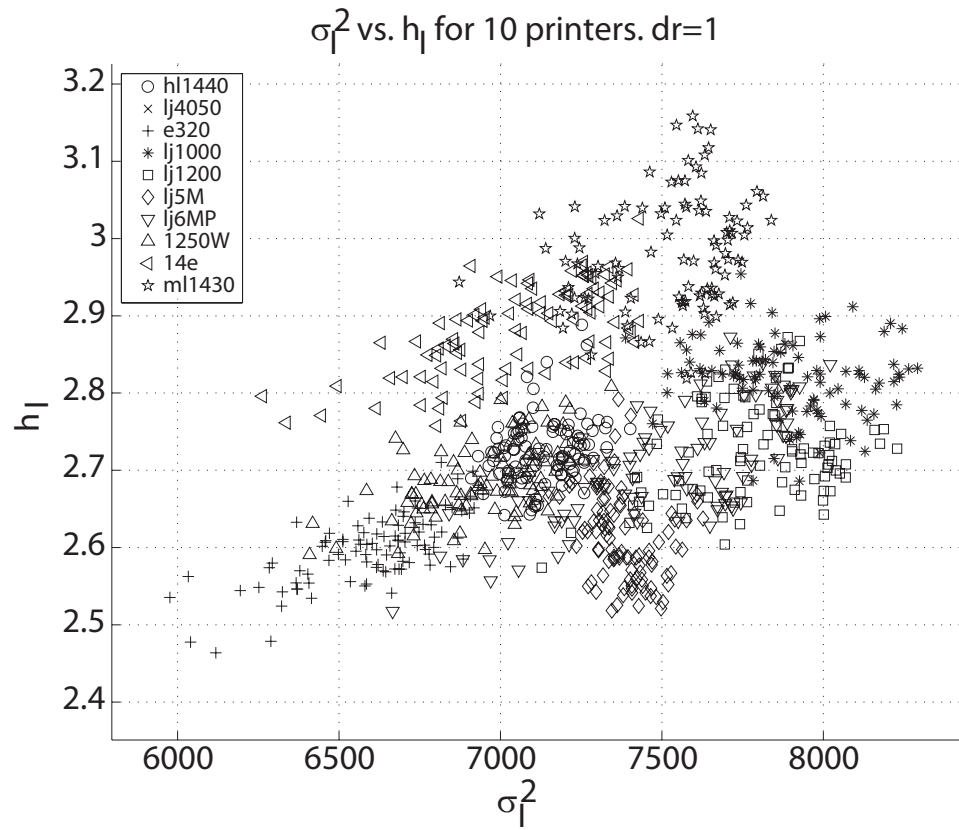


Figure 2.25. Scatter plot of features h_{Img} vs σ_{Img}^2 .

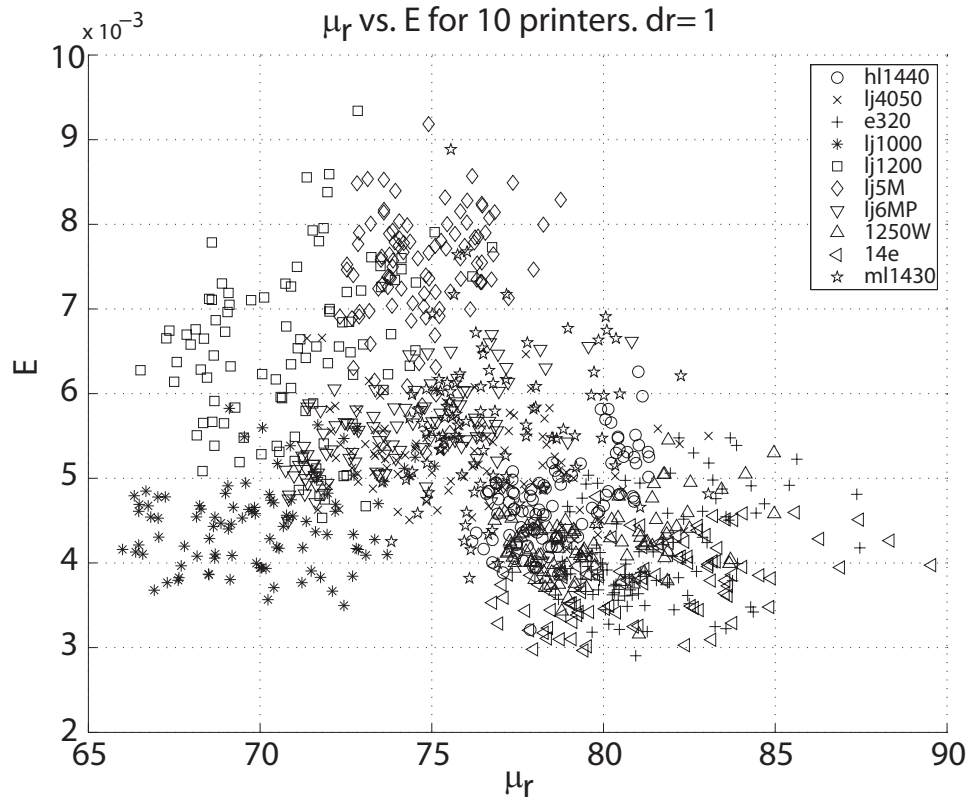


Figure 2.26. Scatter plot of features μ_r vs *Energy*.

the classification results despite the fact that they are classified correctly when using only 4 features.

One drawback to using a 5NN classifier is that it does not generalize well when the ratio of training vectors to dimensionality is relatively low. An SVM classifier is able to provide better generalization in this scenario. Significantly improved results can be achieved by using a SVM with all 22 features. The SVM in this case was trained using feature vectors from 500 “e”s generated independently of the test data. The results from this classification system are shown in Table 2.5.

2.3 Forensic Characterization and Variability in Document Content

In a typical forensic printer identification scenario, it may not be possible to obtain known documents from a set of printers with content matching that of the document in question. Paper type, font size, font type, and even the age of the consumables in the printer may vary. These scenarios are simulated in the following experiment [65].

Four cases are explored, one for each of the categories listed in Table 2.6. In each case the training set consists of 500 “e”s and the test set consists of 300 “e”s. The first case considered is where the printer identification system is trained using data of font size fs_{train} and tested using data of font size fs_{test} with all other variables held constant ($ft = \text{Times}$; $pt = \text{PT01}$). Age is held constant by printing the training and test data immediately after one another. In the second case the system is trained using data of font type ft_{train} and tested using data of font type ft_{test} with all other variables held constant ($fs = 12\text{pt}$; $pt = \text{PT01}$). In the third case the system is trained using data of paper type pt_{train} and tested using data of paper type pt_{test} with all other variables held constant ($fs = 12\text{pt}$; $ft = \text{Times}$). Finally the case where the system is trained on “new” data and tested on “old” is considered. Testing and training data sets printed 5 months apart are used. 10 sub-cases are considered by testing and training using data from the sets $\{fs_x, \text{Times}, \text{PT01}\}$ and $\{12\text{pt}, ft_x, \text{PT01}\}$. This is representative of a forensic scenario where the printing device that created a suspect

Table 2.4. Confusion matrix for GLCM based printer identification using 4 features, 5NN classifier, and $dr = 2$. (Blank indicates -0-)

train \ test	lj5m	lj6mp	lj1000	lj1200	E320	ml1430	lj4050	hl1440	1250w	14e	Output Class
lj5m	199	20	6	10	1			42	22		lj5m
lj6m	39	68	67	37	24	11		34	20		lj6m
lj1000	25	31	137	89		2		8	8		lj1000
lj1200	31	15	145	93	1	1		9	5		lj1000
E320	1				282				13	4	E320
ml1430		35	15	5	1	148		36	5	55	ml1430
lj4050							300				lj4050
hl1440	18	12	3	2	2	4		156	71	32	hl1440
1250w	24	11		1	105			43	114	2	1250w
14e		1			36	21		90	32	120	14w

Table 2.5. Confusion matrix for GLCM based printer identification using 22 features, SVM classifier, and $dr = 2$. (Blank indicates -0-)

train \ test	lj5m	lj6mp	lj1000	lj1200	E320	ml1430	ml1450	hl1440	1250w	14e	Output Class
lj5m	296	2		1		1					lj5m
lj6m	1	256	6		17			15	5		lj6m
lj1000	2	2	284	12							lj1000
lj1200	7	2	2	289							lj1200
E320					300						E320
ml1430	1					299					ml1430
ml1450							300				ml1450
hl1440		28				5	2	259	6		hl1440
1250w								3	292	5	1250w
14e								17	67	216	14e

Table 2.6. Four variables considered for forensic identification experiment.

Category	Sub-Types
Font Size (fs)	08 point 10 point 12 point 14 point 16 point
Font Type (ft)	Arial Courier Garamond Impact Times
Paper Type (pt)	PT01: 20lb., 84brt PT02: 28lb., 97brt PT03: 32lb., 100% cotton
Age (consumables)	-

Table 2.7. Percent correct classification for varying font size. (% after SVM)

$f_{s_{train}} \setminus f_{s_{test}}$	8pt	10pt	12pt	14pt	16pt
8pt	100 (87.6)	90 (82.9)	80 (61.0)	50 (43.0)	40 (35.1)
10pt	100 (78.3)	100 (95.3)	90 (72.9)	70 (56.3)	50 (47.9)
12pt	80 (58.3)	90 (73.3)	100 (93.0)	100 (84.1)	80 (66.0)
14pt	50 (43.6)	70 (62.7)	100 (88.9)	90 (89.7)	90 (81.2)
16pt	40 (37.6)	50 (48.1)	80 (74.4)	90 (84.2)	90 (89.5)

document needs to be identified given only the document in question and newly generated test and training data from the printer.

The results for Case 1 are shown in Table 2.7. The rows of the table correspond to the value of $f_{s_{train}}$ and the columns correspond to the value of $f_{s_{test}}$. Each entry contains two values. The first value is the percent correct classification of the system (i.e. the percentage of printers classified correctly from those listed in Table 2.2 with the exception of the lj4050). The second value, enclosed in parentheses, is the percent correct classification of the individual feature vectors immediately after the SVM. From the data in the table it is hypothesized that when the font sizes of the training and testing data are within 2 points of each other, at least 9 out of 10 printers are correctly classified.

The results for Case 2 are shown in Table 2.8. It is concluded from these results that the current feature set is font dependent. If $f_{t_{train}} = f_{t_{test}}$ then 9 out of 10 printers can be classified correctly. At most 7 out of 10 printers are classified correctly if $f_{t_{train}} \neq f_{t_{test}}$. Even though the font size is 12pt for each font type, the height of

Table 2.8. Percent correct classification for varying font type. (% after SVM)

$ft_{train} \setminus ft_{test}$	arial	courier	garamond	impact	times
arial	90 (84.1)	40 (35.0)	40 (26.0)	20 (17.8)	40 (34.7)
courier	20 (23.0)	90 (86.8)	50 (43.8)	0 (2.6)	50 (49.3)
garamond	10 (12.4)	40 (43.2)	90 (82.3)	10 (11.9)	20 (27.8)
impact	10 (16.8)	10 (10.4)	10 (11.4)	90 (82.9)	10 (17.9)
times	20 (30.1)	70 (57.0)	40 (33.0)	10 (6.6)	90 (84.0)

the “e” in each instance is different. It is possible that this implicit font size difference partly causes the low classification rates for different font types. The Times “e” and Courier “e” are approximately the same height and the classification rate for training on Times and testing on Courier is shown to be 70%.

The results for different paper types, Case 3, are shown in Table 2.9. 100% correct classification is achieved if both the training and testing sets use the same paper type. If paper type PT01 or PT02 is used for training, and PT01 or PT02 for testing, then at least 9 out of 10 printers are classified correctly. The same is not true with paper type PT03. Paper types PT01 and PT02 are both visually similar except that PT02 appears slightly smoother and brighter. PT03 has a visually rougher texture than the other two paper types. The features used for classification may be affected by the paper texture as well as textures from the printer itself.

In Table 2.10 are shown the results for the fourth case, training with new data and testing with old. At least 7 out of 10 printers are correctly identified in each sub-case. The individual SVM classifications show that in each of these sub-cases

Table 2.9. Percent correct classification for varying font size. (% after SVM)

$pt_{train} \setminus pt_{test}$	PT01	PT02	PT03
PT01	100 (93.0)	90 (83.3)	60 (47.2)
PT02	90 (75.2)	100 (93.2)	40 (32.4)
PT03	50 (40.4)	30 (28.1)	100 (93.0)

Table 2.10. Percent correct classification for varying age. Training and testing data generated 5 months apart.

$f_{S_{train}} \setminus f_{S_{test}}$	08pt	10pt	12pt	14pt	16pt
PT01	90 (66.0)	90 (76.3)	90 (72.3)	80 (66.9)	80 (67.8)
Train Test	Arial	Courier	Garamond	Impact	Times
PT01	70 (64.6)	70 (62.6)	80 (67.3)	80 (67.0)	80 (58.5)

the lj1200 was classified as an lj1000. This problem was observed in previous work and attributed to the fact that the two printers appear to have the same or similar mechanical structure.

2.4 Survivability of the Intrinsic Signature

There are many instances where existence of the intrinsic signature in the printed document would be undesirable to the creator of a document. This is useful, for example, in protecting the anonymity of people distributing printed documents during peaceful protest. On the other hand, some groups may want to hide the intrinsic signature for illegal purposes such as distribution of counterfeit currency. Such individuals may attempt to hide or obscure the intrinsic signature by modifying the document before printing.

The only way to achieve complete removal of the intrinsic signature is through extensive, non-trivial, and perhaps impossible hardware modifications of the printer. Instead of complete removal, we focus on attacking the graylevel co-occurrence matrix based intrinsic signature detection for text documents. The term “attack” in this case refers to any method of modifying or altering the printed document such that the printer identification system described in the previous sections fails to correctly identify the source printer. Specifically methods that modify the document before

it is printed are considered. The proposed method embeds false textures in each character of a text document. The changes in each character's texture are performed before the data is sent to the printer, so no hardware modifications are necessary. The false texture is designed to prevent correct printer identification by changing the graylevel co-occurrence features used as intrinsic signatures of the printer such that they do not lead to correct printer identification.

In addition to the GLCM and pixel features already described, 15 banding features are added to the feature vector. This seems counter-productive considering that banding is difficult to estimate from a saturated region such as a character of text. However, the attacks considered may cause the printer to perform halftoning when printing the text in which case banding features can be more easily estimated.

To obtain the banding features, a normalized projection of the character is first obtained as

$$b(i) = \frac{\sum_{(i,j) \in \Omega} I(i,j)}{\sum_{(i,j) \in \Omega} 1}. \quad (2.59)$$

The banding features are taken as the squared magnitudes of the first 15 points of the DFT spectrum defined as

$$P_b(n) = \frac{1}{N} \left| \sum_{k=0}^{N-1} b(k) e^{-j \frac{2\pi kn}{N}} \right|^2. \quad (2.60)$$

Choosing $N = 240$ and a sampling resolution of 2400DPI, the first 15 points correspond to bins centered at $\{10, 20, \dots, 150\}$ cycles/inch.

2.4.1 System Overview

Figure 2.27 shows the block diagram of the intrinsic printer identification scheme using the features described in the previous section. The first step is to scan the document at 2400 dpi with 8 bits/pixel (grayscale). Next all the letter “e”s in the document are extracted. Features are extracted from each character forming a feature vector for each letter “e” in the document. Each feature vector is then classified individually using a support vector machine (SVM) classifier [66]. The SVM classifier is

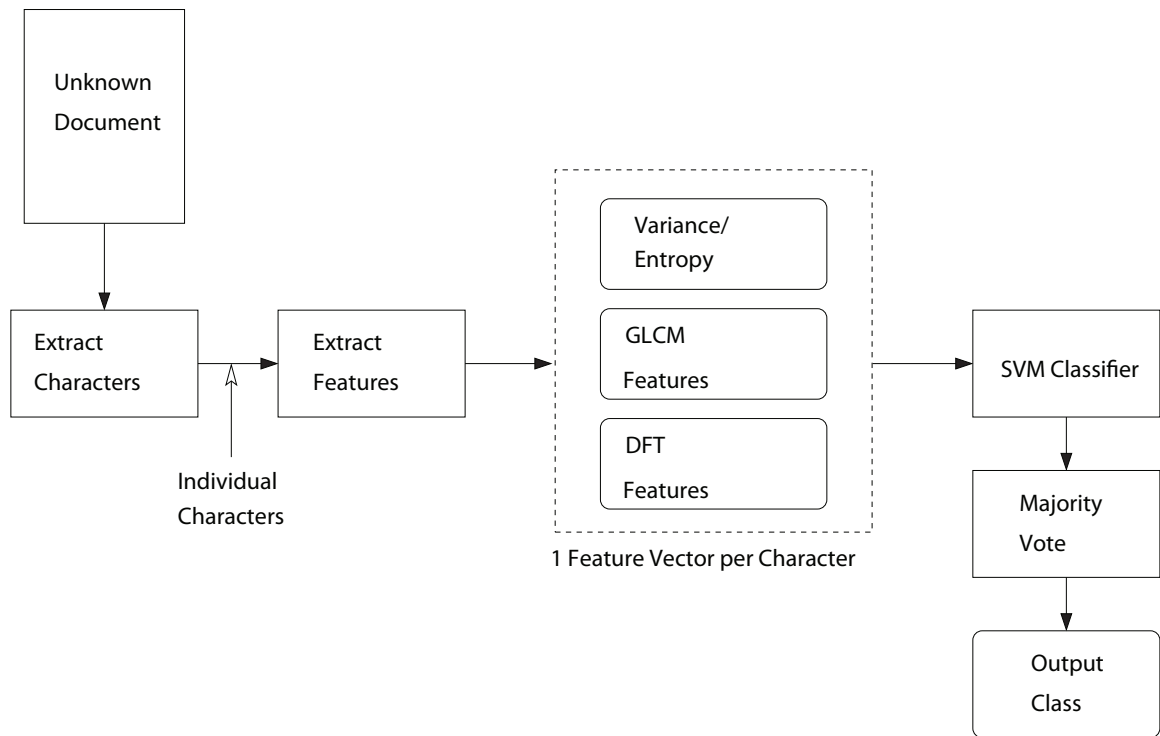


Figure 2.27. System diagram of printer identification scheme to handle attacks.

Table 2.11. Printers used in attack experiments.

Printer Identifier	Manufacturer	Model	DPI
P_1	HP	Color LaserJet 2605	1200
P_2	HP	Color LaserJet 3800	1200
P_3	Samsung	ML-1450	600
P_4	HP	LaserJet 4450	600

trained using 500 feature vectors from each of 4 printers listed in Table 2.11. The final classification is decided by a majority vote performed on the resulting classifications from the SVM classifier.

2.4.2 Attacks

In order to defeat the intrinsic printer identification system, attacks designed to mimic and obscure the underlying banding signals in the document. This is performed by preprocessing the document before it is sent to the printer with the addition of sinusoidal signals and gaussian noise to the text regions of the document [67].

The first attack adds a fixed amplitude fixed frequency sinusoidal signal

$$s_1(i) = \frac{A}{2} \left[1 + \cos \left(\frac{2\pi f i}{R_p} \right) \right], \quad (2.61)$$

to the printed regions of a document before sending it to the printer. Assume $I(i, j)$ is an 8-bit grayscale document image to be printed. In the case of a text document, I will only take on values 0 and 255 corresponding to printed text or background respectively. The attacked document image can then be written as

$$\tilde{I}(i, j) = \begin{cases} I(i, j) + s_1(i) & , \quad I(i, j) = 0 \\ 255 & , \quad else \end{cases}. \quad (2.62)$$

This attack is designed to mimic an intrinsic banding signal.

The second attack is a binarized version of the first. Since \tilde{I} above is grayscale, it will be automatically halftoned by the printer driver or the printer itself. If for some

reason it is not desirable to have the printer or printer driver perform the halftoning the attack function can be modified as follows. Let the sinusoidal signal define a threshold

$$s_2(i) = \frac{1}{16} \left[1 + \cos \left(\frac{2\pi f i}{R_p} \right) \right]. \quad (2.63)$$

For every black pixel in I , $I(i, j) = 0$, generate a random number $\gamma_{i,j}$ uniform in $[0, 1]$. Then define the attacked image as

$$\tilde{I}(i, j) = \begin{cases} 255 & , \quad I(i, j) = 0 \quad \text{and} \quad \{\gamma_{i,j} < s_2(i)\} \\ I(i, j) & , \quad \text{else} \end{cases}. \quad (2.64)$$

The third attack uses a frequency hopping sinusoidal signal defined by

$$s_3(i) = \frac{A}{2} [1 + \cos(\phi(i))], \quad (2.65)$$

$$\phi(i) = \phi(i-1) + \frac{2\pi f(i)}{R_p}, \quad (2.66)$$

where $f(i)$ is a randomly chosen frequency from from $[30, 120]$. Additionally, $f(i)$ remains fixed over α consecutive rows where α is randomly chosen from $[0, 100]$ with each frequency change. The attacked image then takes the same form as Equation 2.62 with s_3 in place of s_1 . This attack is designed to defeat the majority voting performed on the classifications of each individual “e”.

The fourth attack is a binarized version of the third using

$$s_4(i) = \frac{1}{2^\nu} [1 + \cos(\phi(i))], \quad (2.67)$$

as the decision threshold on whether or not to turn a pixel white (as in Equation 2.64). The parameter ν is an integer greater than zero that controls the threshold. Larger values of ν will result in fewer pixel being flipped, or equivalently lower noise power.

The fifth attack adds Gaussian noise to the printed regions of the document image. In this case the attacked image is

$$\tilde{I}(i, j) = \begin{cases} I(i, j) + \mathbf{X} & , \quad I(i, j) = 0 \quad \text{and} \quad \mathbf{X} \in [0, 3\sigma] \\ I(i, j) & , \quad \text{else} \end{cases}, \quad (2.68)$$

where $\mathbf{X} \sim N\left(\frac{3}{2}\sigma, \sigma^2\right)$.

2.4.3 Experiments

A test page was created consisting of approximately 2000 “e”s in 12 point Times Roman font. A 17 page text document was generated consisting of 16 attacked versions of this test page. The first page contained no attack. Four pages contained attack 1 with parameters $A = 50$ and $f = \{30, 60, 90, 120\}$ respectively for the four pages. One page contained attack 2 with parameter $f = 75$. Four pages contained attack 3 with parameter $A = \{25, 50, 75, 100\}$. Three pages contained attack 4 with parameter $\nu = \{2, 3, 4\}$. The last four pages contained attack 5 with parameter $\sigma = \frac{1}{3}\{25, 50, 75, 100\}$.

The document was printed on each of the four printers listed in Table 2.11. Each was then scanned in 8-bit grayscale at 2400 DPI. The training set consisted of 1500 feature vectors from the non-attacked page only. The feature vectors from the remaining 16 (attacked) pages were used for testing using the SVM classifier.

Since the SVM classifier is limited in mapping a feature vector only to one of the known training classes, it has limited scalability. If more printers are involved in training-testing then classification accuracies may drop significantly. Therefore, in addition to the SVM experiments, linear discriminant analysis (LDA) is performed to determine which features are the most significant in each attack scenario. Since it is expected that the DFT features will work best with non-saturated text (i.e. attacked text), and the GLCM will work with saturated text (i.e. non-attacked text), LDA is performed on the GLCM and DFT features independently.

2.4.4 Results

The graph in Figure 2.28 shows the classification accuracy of the individual “e”s in the test documents attacked with the single frequency sinusoidal signals of attack 1. This attack does not have a large effect on the classification accuracy except for P_2 when $f = 120$ cycles/inch.

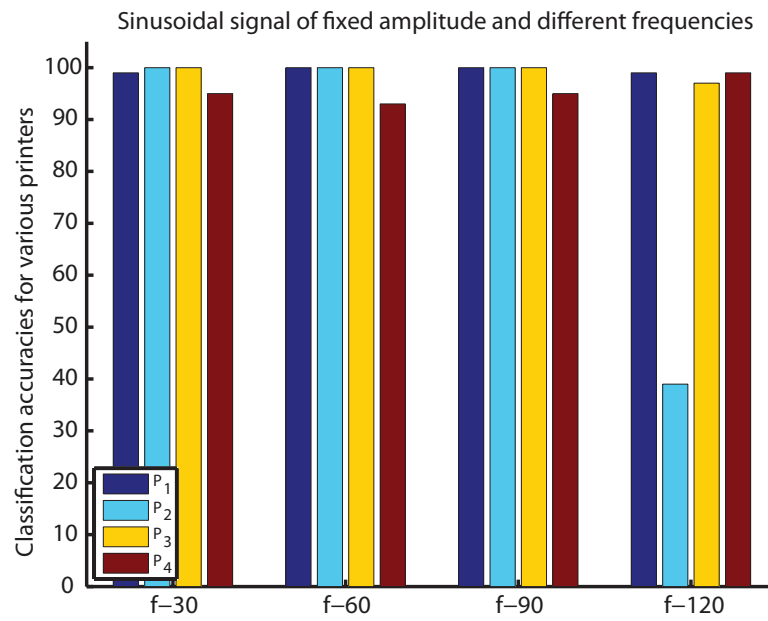


Figure 2.28. SVM classification results of individual “e”s after attacking with a single frequency sinusoid (attack 1).

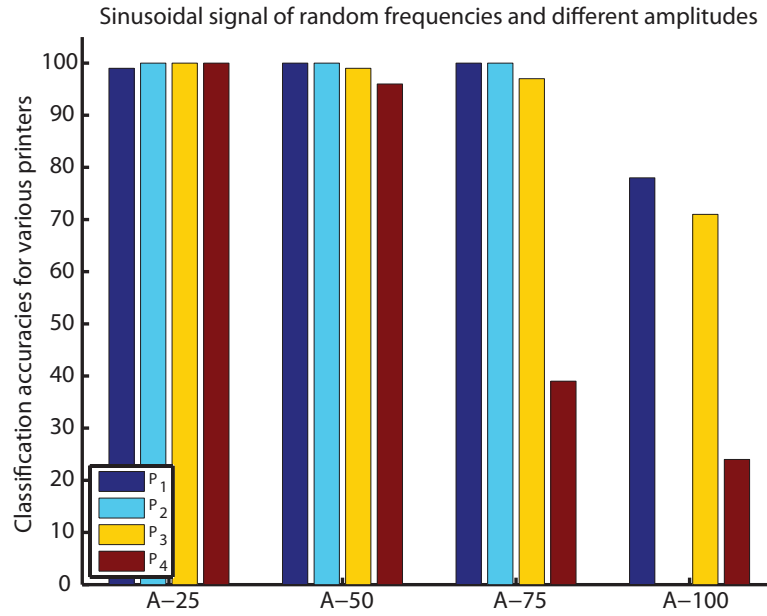


Figure 2.29. SVM classification results of individual “e”s after attacking with a random frequency sinusoid (attack 3).

The graph in Figure 2.29 shows the classification accuracy of the individual “e”s in the test documents attacked with the random frequency binarized sinusoidal signals of attack 3. Here we see classification accuracy drop as the amplitude A goes above 75. However the classification accuracy still remains relatively high for P_1 and P_3 .

The results for attacks 2 and 4, the binarized version of attacks 1 and 3, show all printers above 99% accuracy except for P_2 which has more than 90% of its “e”s classified as belonging to P_3 .

The graph in Figure 2.30 shows the classification accuracy of the individual “e”s in the test documents attacked with the gaussian noise of attack 5. Classification accuracy starts to drop off after $\sigma = 25$, at which point the visual quality of the text also begins to decline.

Scatter plots of feature pairs are shown in Figures 2.31 through 2.35 to gain a better understanding of how the features we have chosen behave in each of the attack scenarios. Each figure contains two subfigures each showing a scatter plot of the first two features from the reduced feature set obtained by applying LDA to the

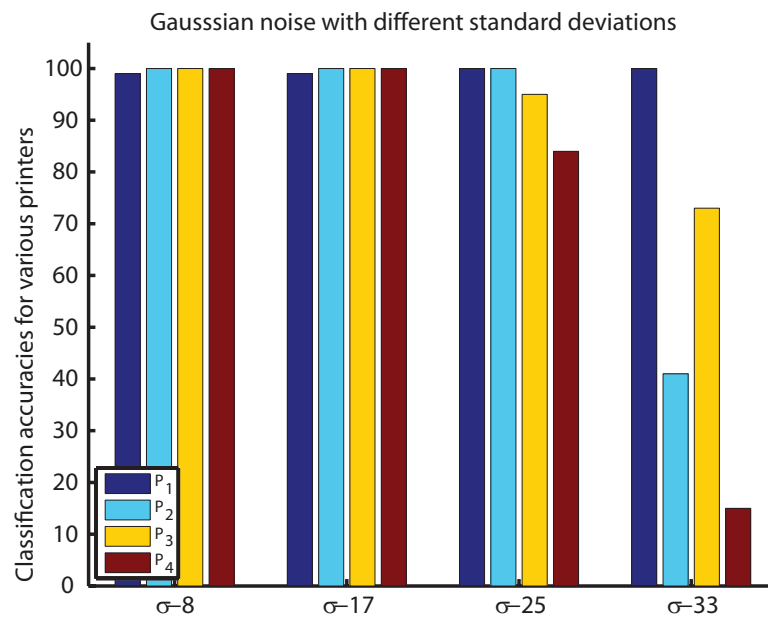


Figure 2.30. SVM classification results of individual “e”s after attacking with Gaussian noise (attack 5).

24 GLCM and pixel features (subfigure ‘a’), and to the 15 DFT features (subfigure ‘b’). Figure 2.31 shows the non-attacked training vectors. In each scatter plot, an oval is drawn for each class to represent the boundary within which 90% of that class’ training vectors lie. These ovals are superimposed on Figures 2.32 through 2.35 to show how the features from that attacked page correspond to the original non-attacked features. We see from these figures that there is a large separation between clusters in at least one of the two reduced feature spaces, which suggests that in all cases the correct printer can be identified even when the printed page is perceptually different from the original. Images of “e”s from for each of these attacks is shown in Table 2.12.

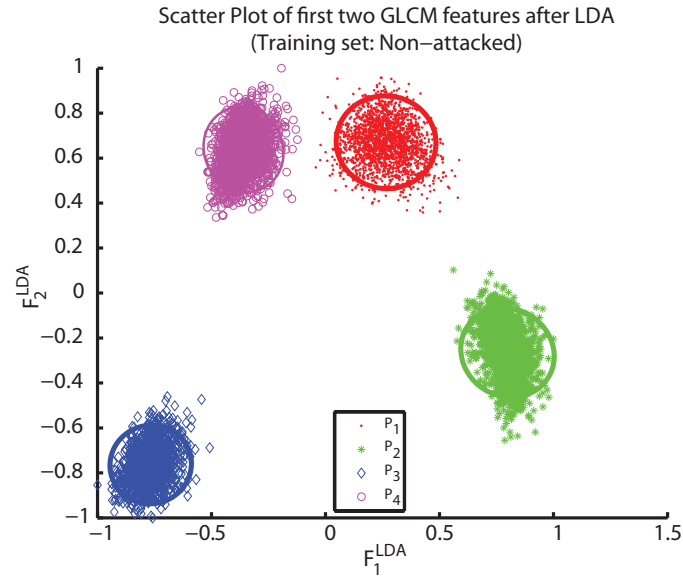
2.5 Time Variation of the Intrinsic Signature

The printer classification systems described up to this point do not perform printer identification, in the sense that they instead match an unknown document to the closest matching known printer. Consider the case where a document is from an unknown printer. Using a distance based classification method can help include the case where the document is outside the set of known printers.

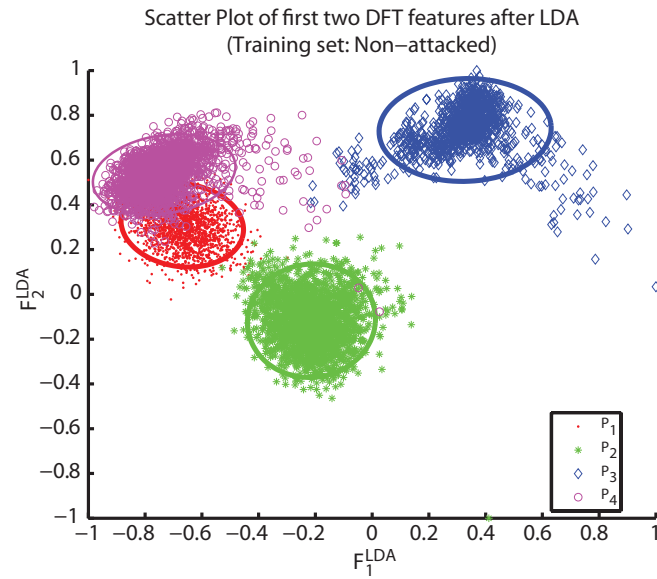
Figure 2.36 shows the block diagram of a distance based intrinsic printer identification scheme [68] using the features described in the previous section. Given a document with an unknown source, the printer that created it is to be identified as well as providing a metric signifying the confidence with which that decision was made.

The first step is to scan the document at 2400 dpi and 8 bits/pixel grayscale. Next all the letter “e”s in the document are extracted. Features are extracted from each character forming a feature vector for each letter “e” in the document. The system is trained using feature vectors from several known printers.

The Euclidean distance metric in high dimensional space is prone to errors introduced by non-discriminative features. It is therefore important to perform feature

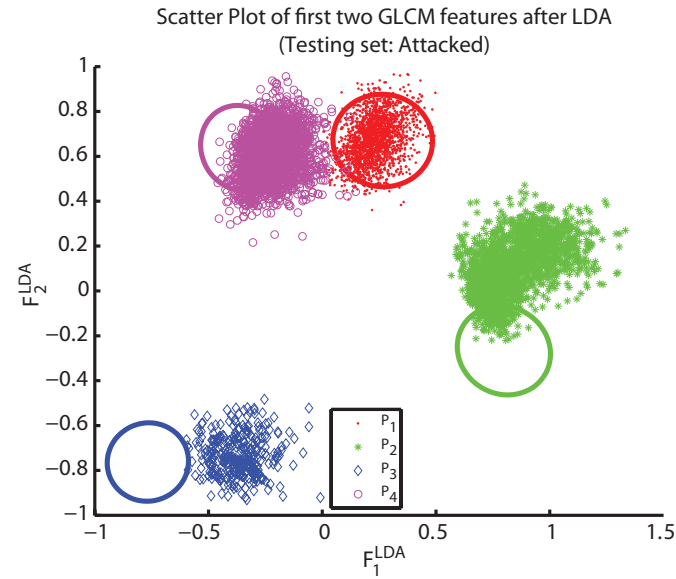


(a)

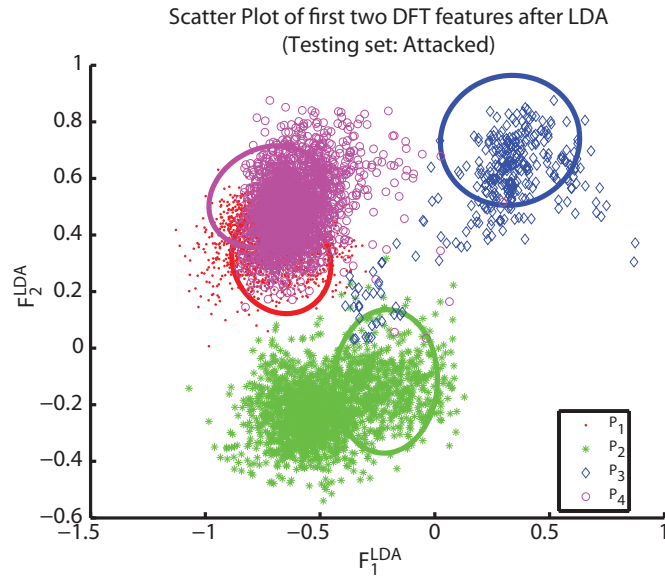


(b)

Figure 2.31. Scatter plot of first two LDA features from non-attacked “e”s. Ellipses identify the boundary of the training clusters. All points are from the non-attacked document for that printer and are used to define the training cluster. (a) LDA on GLCM Features. (b) LDA on DFT Features.

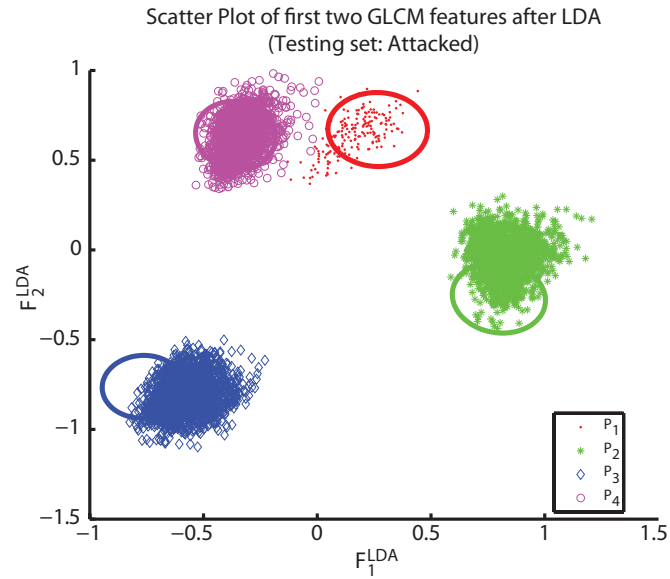


(a)

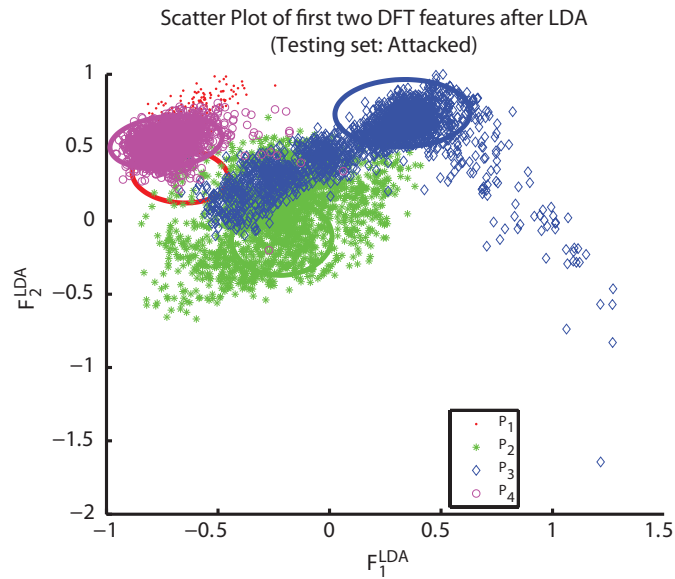


(b)

Figure 2.32. Scatter plot of first two LDA features for attack 1 (fixed sinusoid: $A = 50$, $f = 90$). Ellipses identify the boundary of the training clusters. All other points are from the attacked document for that printer. (a) LDA on GLCM Features. (b) LDA on DFT Features.

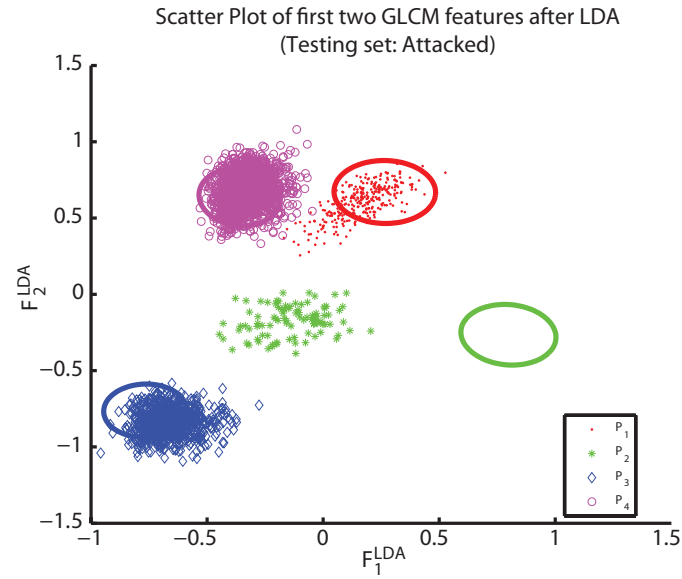


(a)

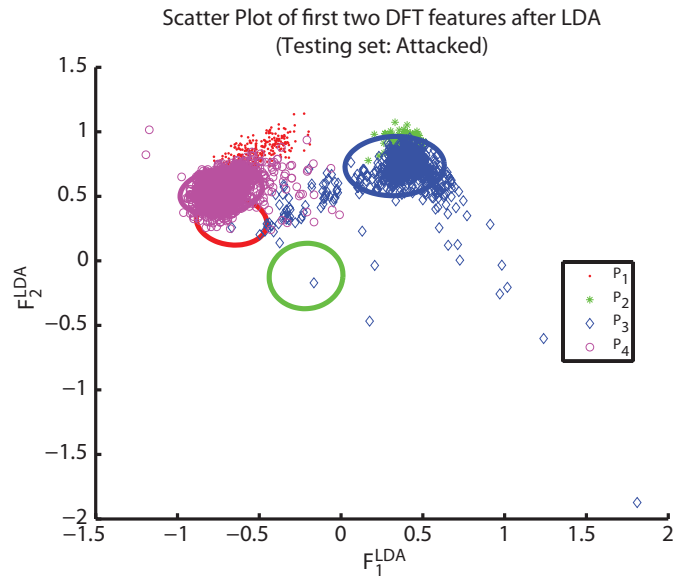


(b)

Figure 2.33. Scatter plot of first two LDA features for attack 3 (random frequency sinusoid: $A = 25$). Ellipses identify the boundary of the training clusters. All other points are from the attacked document for that printer. (a) LDA on GLCM Features. (b) LDA on DFT Features.

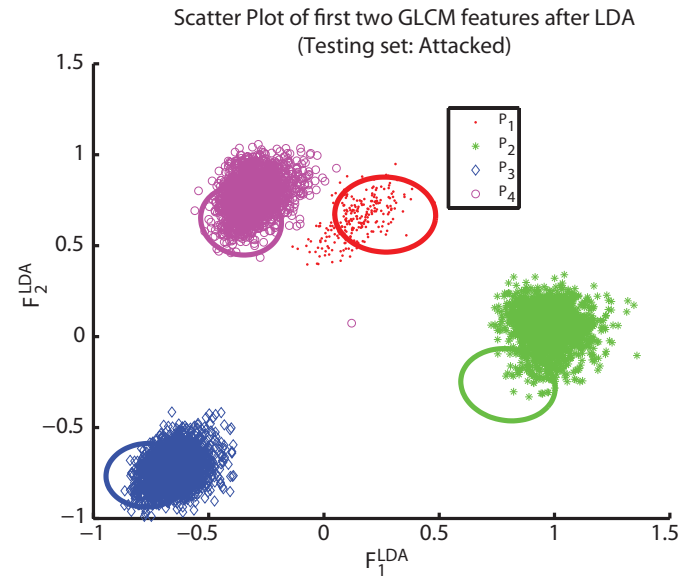


(a)

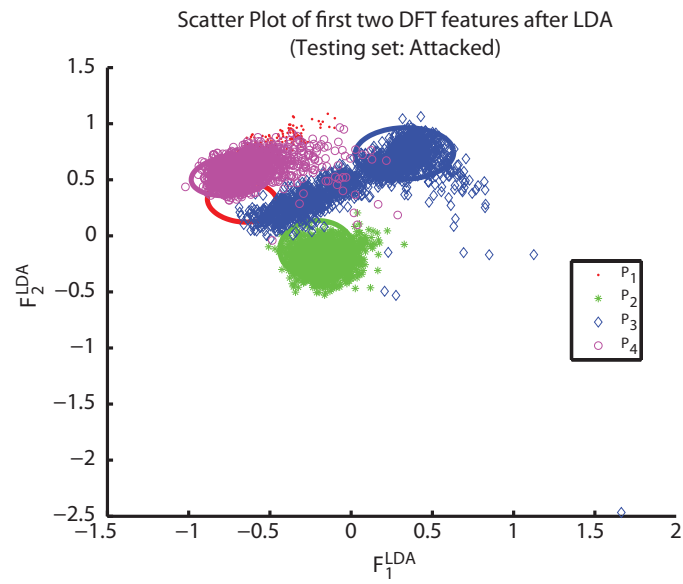


(b)

Figure 2.34. Scatter plot of first two LDA features for attack 4 (random frequency binarized sinusoid: $\nu = 4$). Ellipses identify the boundary of the training clusters. All other points are from the attacked document for that printer. (a) LDA on GLCM Features. (b) LDA on DFT Features.



(a)



(b)

Figure 2.35. Scatter plot of first two LDA features for attack 5 (Gaussian noise: $\sigma = 8$. Ellipses identify the boundary of the training clusters. All other points are from the attacked document for that printer. (a) LDA on GLCM Features. (b) LDA on DFT Features.

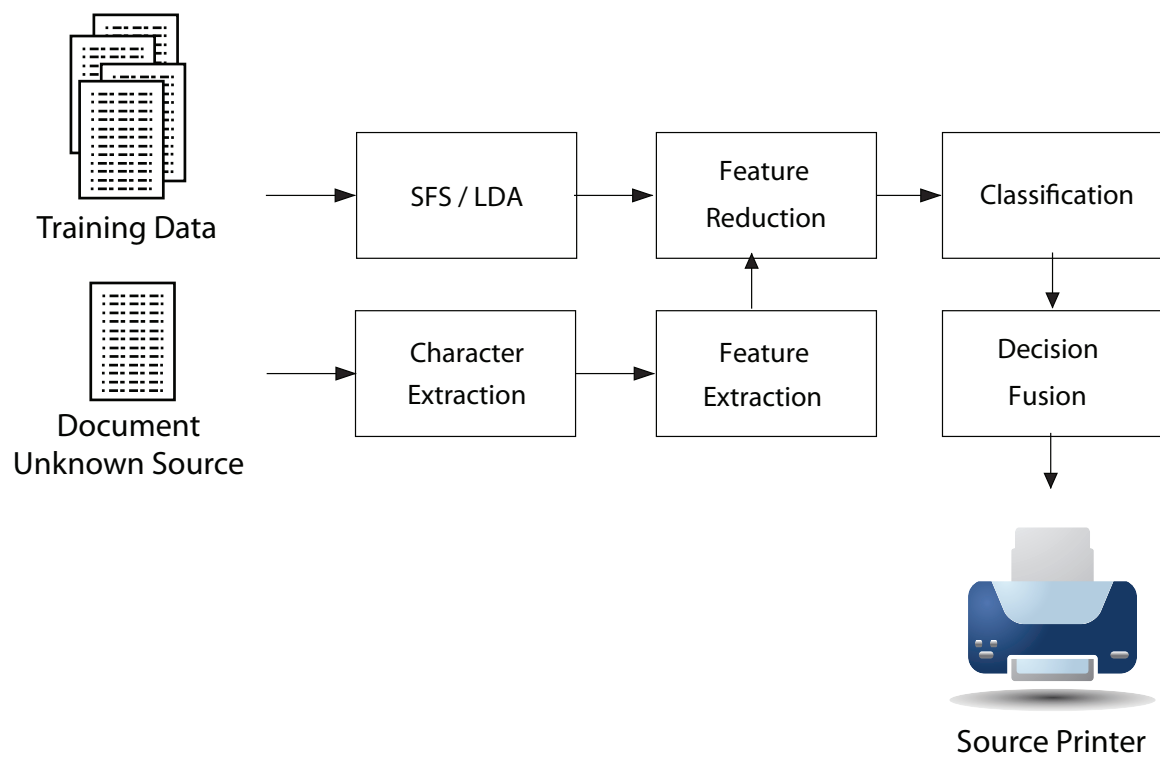

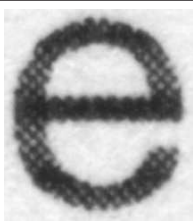
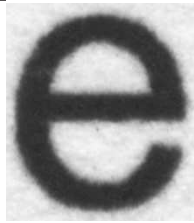

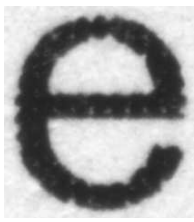
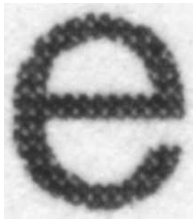









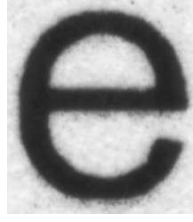

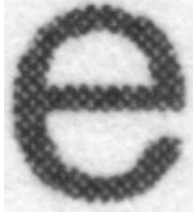
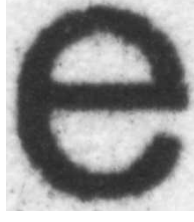



Figure 2.36. System diagram for forensic printer identification.

Table 2.12. Images of attacked “e”s.

Attack\ Printer	P_1	P_2	P_3	P_4
-				
1 $A = 50, f = 90$				
3 $A = 25$				
4 $\nu = 4$				
5 $\sigma = 8$				

reduction before estimating the thresholds and error probabilities for the distance based classifier. Training data is used to perform two-level feature reduction. First a subset of the most discriminating features is selected by using sequential forward selection (SFS) [69]. The SFS feature selection process works as follows. Let Υ represent the set of features selected by the SFS algorithm, and let the set of all features be represented by Γ . The procedure starts with $\Upsilon = \{\emptyset\}$ (the empty set). The feature that provides the best separation between classes by itself is added to the set Υ . This is then iterated by adding the feature from the remaining set of features, $\Gamma \setminus \Upsilon$, that together with Υ provides the best separation between classes. The stopping criterion is typically when a certain number of features has been selected.

SFS is only step-optimal, that is it does not guarantee that the final set of n features is the best set of n features even though each feature when added provided the best improvement in class separation. A slightly improved performance can be obtained by using more sophisticated feature selection techniques such as sequential floating forward selection (SFFS), sequential floating backward selection (SFBS), and their variations [69–71].

Feature reduction is then performed by using linear discriminant analysis (LDA) [72] to find an optimal linear combination of features. Consider the class distribution shown in Figure 2.37. The classes are not linearly separable on either of the two feature axes alone. LDA finds an optimal combination of the two features on which the two classes have the greatest linear separation. In this case, the optimal combination of features is represented by the diagonal axis with slope approximately equal to 1. When the data points are projected on this axis, a linear separation plane is easily found to separate the two classes of data.

In the proposed method, SFS is used to select a subset of features from the initial 22 dimensional feature set. LDA then projects this subset into a three dimensional feature space. Each of the resulting features is a linear combination of the SFS reduced features. The dimensionality of the LDA reduced feature space is chosen based on the

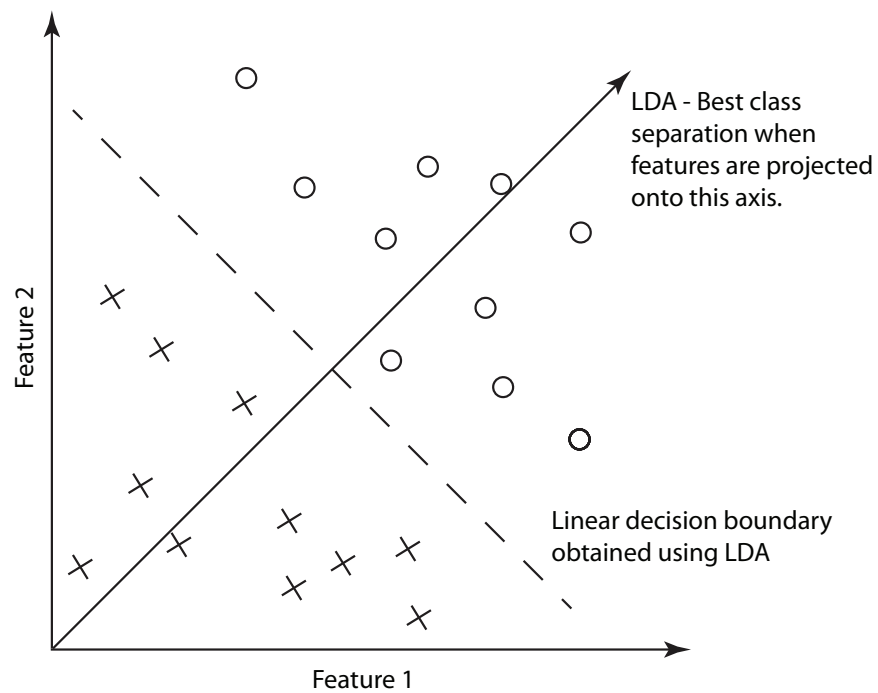


Figure 2.37. Linear discriminant analysis applied to a two class problem and two-dimensional feature space. The two classes are not linearly separable using either of the two original features. Once the data is projected onto the LDA axis it becomes linearly separable.

ratio of all eigenvalues as compared to the highest eigenvalue of the LDA projection matrix.

Finally, each of the reduced features are scaled appropriately so that they lie in the interval $[-1, 1]$. Each training class is represented by its cluster center (mean of all the reduced feature vectors from that class) and a threshold. Threshold values are independently chosen for each class/cluster based on the trade-off between false acceptance and false rejection rates as given by

$$\epsilon = \lambda FRR + (1 - \lambda) FAR, \quad (2.69)$$

for some choice of λ .

To determine what printer generated a document, feature vectors from the document are projected into the lower dimension feature space using the reduction matrix generated during the training phase and are identified as belonging to a particular printer or an unknown source. Each “e” in the document will have an individual classification result which are then merged to give the source printer.

Table 2.13. Printers used in our experiments.

Printer Identifier	Manufacturer	Model	DPI
<i>P01</i>	HP	LaserJet 1000	1200
<i>P02</i>	Samsung	ML-1450	1200
<i>P03</i>	HP	LaserJet 1200	1200
<i>P04</i>	HP	LaserJet P1006	1200
<i>P05</i>	Okidata	B220n	600
<i>P06</i>	Brother	HL 2140	600
<i>P07</i>	HP	LaserJet 1505	1200
<i>P08</i>	HP	LaserJet 1200	1200
<i>P09</i>	Samsung	ML-1430	1200

2.5.1 Experiments and Results

The printers listed in Table 2.13 are subject to heavy daily use. The experimental data set consists of a test page printed monthly by each printer over the course of two years. Each test page contains approximately 600 “e”s in 12 point Times Roman font. Several times during the course of data generation, the consumables within each printer (toner cartridge) reached end-of-life and were replaced.

Several experiments are performed by varying the set of data used to train the system. This training data consists of at least 300 “e”s from each printer. In all the experiments that follow, the features selected by SFS include the following 6 features: h_I , μ_r , h_{xy1} , ρ , h_D , and E_S as defined in Section 2.2. LDA is applied on the set of selected features and projects them into a three dimensional feature space.

The first experiment trains the system using data from a set of two dates for all printers and then tests using all other dates. By varying the date of the training data the average accuracy varies over the range 37% to 56%. Two confusion matrices from this experiment are shown in Tables 2.14 and 2.15, one using the training set going the highest average accuracy and the other using the worst performing training set. Each entry (i, j) denotes the percentage of feature vectors (“e”s) in the testing set that are from printer i and classified as belonging to printer j . The last column denotes percentage of feature vectors that were classified as belonging to none of the known classes. Printers P03 and P08 are the same printer model, and P01 is mechanically the same as P03 and P08. Printers P04 and P07 are also mechanically similar. For a given date, the age of each printer’s consumables may differ, so it is not possible to state whether training data from newer or older consumables are more desirable. The interclass and intraclass distances obtained during one iteration of this first experiment are shown in Figure 2.38. Each plot shows the distribution of distances from class n ’s mean for intraclass (solid line) and interclass (dashed line) feature vectors. The top-left plot shows the distances for class 1 and continuing in row order the bottom-right plot shows the distances for class 9.

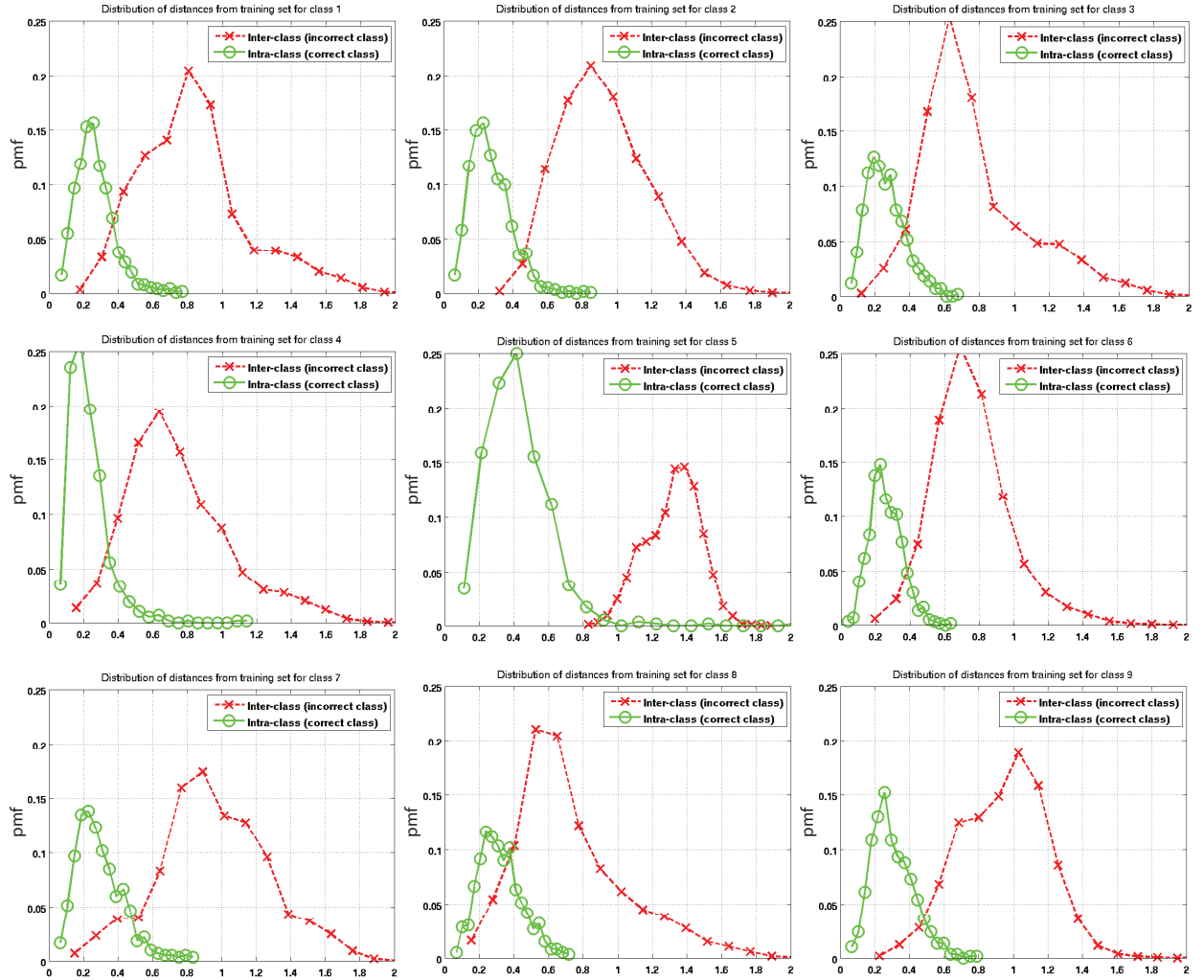


Figure 2.38. Each subplot shows the intraclass (solid line) and interclass (dashed line) distances for a single class. The top-left plot is for class one with numbering continuing in row order to class 9 in the bottom right plot.

Table 2.14. Confusion matrix for Experiment 1. Best training set with 56% average accuracy.

	P01	P02	P03	P04	P05	P06	P07	P08	P09	Unknown Class
P01	37.6	1.7	35.0	5.9	-	0.1	1.7	2.5	-	15.5
P02	0.6	63.1	6.3	0.2	-	14.2	-	0.7	-	14.8
P03	6.8	5.6	55.5	6.7	-	0.7	-	9.6	-	15.1
P04	13.5	-	2.8	65.6	-	0.4	6.9	6.1	-	4.7
P05	-	-	-	-	100.0	-	-	-	-	-
P06	-	0.8	-	3.4	-	87.5	-	0.1	3.8	4.5
P07	1.1	-	-	32.5	-	-	60.7	-	-	5.8
P08	7.8	7.9	41.5	7.7	-	3.6	-	9.0	0.1	22.4
P09	-	0.2	0.1	10.6	-	49.5	0.4	0.7	21.9	16.7

Table 2.15. Confusion matrix for Experiment 1. Worst training set with 37% average accuracy.

	P01	P02	P03	P04	P05	P06	P07	P08	P09	Unknown Class
P01	24.5	11.7	0.2	16.3	-	0.6	6.0	1.0	-	39.6
P02	1.0	40.7	0.1	-	2.9	12.1	0.3	0.7	0.4	41.8
P03	41.7	37.4	0.3	2.0	-	2.1	-	0.9	-	15.6
P04	5.3	6.5	-	40.9	0.1	18.6	4.0	0.1	1.1	23.2
P05	-	-	-	-	88.5	-	-	-	-	11.5
P06	-	5.5	-	0.2	-	56.1	0.3	0.1	9.4	28.5
P07	0.2	0.1	-	31.2	-	0.8	55.0	-	-	12.8
P08	21.7	35.0	0.3	1.7	-	6.5	-	7.1	0.2	27.7
P09	0.3	1.3	-	2.2	0.7	36.8	0.1	0.1	22.0	36.3

The second experiment treats P01/P03/P08 as a single class and P04/P07 as a single class since these printer groups are mechanically similar. Instead of trying to

Table 2.16. Confusion matrix for Experiment 2. Best training set with 71% average accuracy.

	P01/P03/P08	P02	P04/07	P05	P06	P09	Unknown Class
P01/P03/P08	78.7	1.7	8.8	-	0.8	0.1	9.9
P02	9.1	64.4	0.5	0.2	12.7	0.6	12.6
P04/P07	24.8	-	71.7	-	0.2	0.2	3.1
P05	-	-	-	99.7	-	-	0.3
P06	-	2.8	5.2	-	88.6	2.9	0.5
P09	0.6	0.3	6.9	-	49.7	25.2	17.3

Table 2.17. Confusion matrix for Experiment 2. Worst training set with 46% average accuracy.

	P01/P03/P08	P02	P04/07	P05	P06	P09	Unknown Class
P01/P03/P08	34.4	32.8	6.4	-	0.8	0.1	25.5
P02	2.5	55.7	0.1	-	14.6	0.7	26.4
P04/P07	24.2	5.4	48.6	-	0.3	0.1	21.2
P05	-	-	-	99.7	-	-	0.3
P06	0.6	1.8	12.7	-	20.1	15.9	48.9
P09	5.3	1.6	6.6	0.1	12.1	16.4	58.0

identify the particular printer, this experiment tries to identify the specific printer model. Confusion matrices for this experiment are shown in Table 2.16 and 2.17 for the best and worst performing training sets respectively. The average accuracy in this case is between 46% and 71% depending on the date chosen for the training data.

The third experiment treats all classes separately but uses training data distributed randomly across all dates in the dataset. The average accuracy of this experiment over 500 iterations was 52%. A confusion matrix generated over the 500 iterations of this experiment is shown in Table 2.18.

Table 2.18. Confusion matrix for Experiment 3. 52% average accuracy.

	P01	P02	P03	P04	P05	P06	P07	P08	P09	Unknown Class
P01	57.5	0.7	10.2	8.2	-	-	3.6	0.8	-	18.9
P02	3.3	57.6	1.9	0.2	0.9	8.5	-	0.2	0.1	27.3
P03	41.2	4.8	24.3	4.2	-	0.3	-	2.4	-	22.8
P04	19.4	-	0.2	55.3	-	2.7	15.4	1.7	0.6	4.7
P05	-	-	-	-	96.6	-	-	-	-	3.4
P06	-	5.8	-	2.6	-	66.5	1.5	-	11.4	12.1
P07	1.1	-	-	17.6	-	-	72.4	-	-	9.0
P08	29.1	9.1	18.1	5.2	-	2.8	0.1	8.9	0.1	26.7
P09	0.5	1.4	-	5.8	0.3	39.4	1.4	0.1	29.5	21.6

Table 2.19. Confusion matrix for Experiment 4. 65% average accuracy.

	P01/P03/P08	P02	P04/07	P05	P06	P09	Unknown Class
P01/P03/P08	73.7	0.9	5.9	-	0.7	-	18.8
P02	7.7	59.0	0.2	1.0	8.5	-	23.6
P04/P07	12.5	0.1	64.2	-	1.6	2.2	19.5
P05	-	-	-	95.5	-	-	4.5
P06	0.1	5.7	4.6	0.1	68.5	11.7	9.3
P09	0.7	1.0	5.7	0.3	40.5	30.3	21.5

The fourth experiment distributes training data randomly across all dates in the dataset and combines classes as was done in Experiment 2. The average accuracy in this case over 500 iterations was 65%. A confusion matrix generated by averaging the outcomes of the 500 iterations of this experiment is shown in Table 2.19.

The reason for better performance when training on all the dates becomes apparent when looking at how the distribution of certain features change over the life of

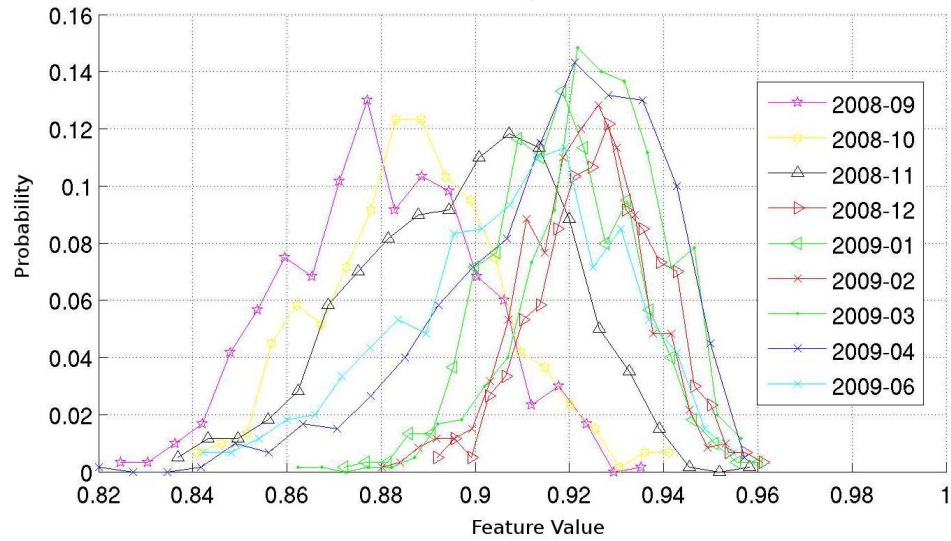


Figure 2.39. Distributions (pmf) of h_{xy1} for P01 over all dates.

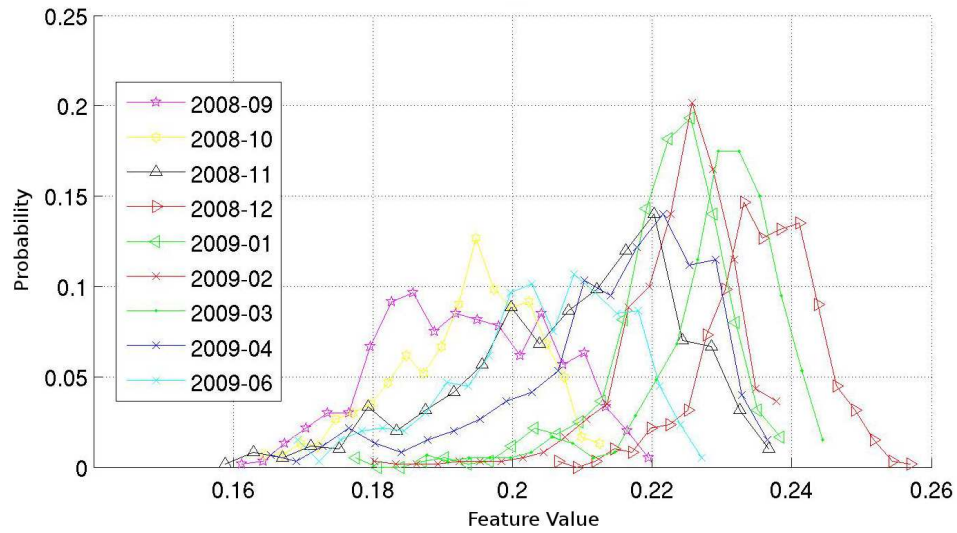


Figure 2.40. Distributions (pmf) of ρ for P01 over all dates.

the consumables in the printer. A plot of the feature h_{xy1} for all dates available in class 1 is shown in Figure 2.39. The entropy appears to increase on average as the consumable age increases. Similarly ρ_{glcm} increases with consumable age as shown in Figure 2.40 for class 1. When training using data from a single date, the data may correspond to different consumable age for each printer.

3. EXTRINSIC SIGNATURES FOR EP PRINTERS

Extrinsic signatures allow arbitrary information to be embedded into a document in addition to or replacing the intrinsic signature that would normally be present. Such an extrinsic signature could contain a variety of information such as the printer's serial number, time and date of printing, or a checksum of the content of the document itself. Several goals outlined in the introduction can be met by using an extrinsic signature including printer identification and forgery detection.

3.1 EP Embedding Techniques

In an EP print process, artifacts are created in the printed output due to electromechanical imperfections in the printer such as fluctuations in the angular velocity of the OPC drum, gear eccentricity, gear backlash, and polygon mirror wobble. In Section 2 it was shown that these imperfections are directly related to the electromechanical properties of the printer. This property allows the corresponding fluctuations in the developed toner on the printed page to be treated as an intrinsic signature of the printer.

The most visible print quality defect in the EP process is banding, which appears as cyclic light and dark bands most visible in midtone regions of the document. Many banding reduction techniques have been successfully demonstrated. These techniques, three of which are laser intensity/timing/pulse width [73], motor control [74], and laser beam steering [75], actively modulate certain process parameters in an effort to reduce the banding present in the printed document. These same techniques could be used to embed information into a document. The information could for example be embedded as artificially generated banding using one of these methods.

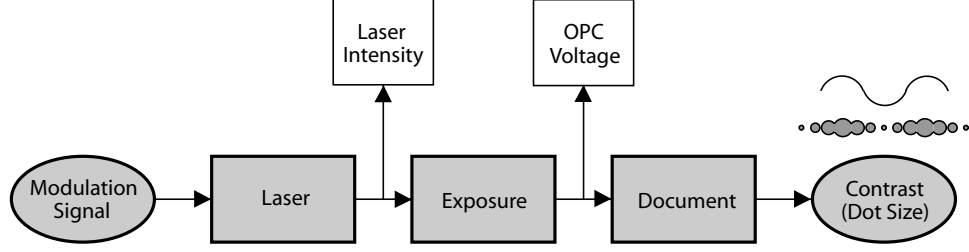


Figure 3.1. Process block diagram for embedding of extrinsic signature using laser intensity modulation.

It is desirable to embed signals with high spatial frequency where the human visual system has relatively low contrast sensitivity. Among the methods mentioned above, motor control has difficulty controlling the EP process at high spatial frequencies due to inherent electro-mechanical limitations. Laser beam steering requires additional process capability that is not found in typical EP engines. However, modulating various laser parameters such as laser power to affect exposure is common practice in typical EP process controls.

As shown in [46], these techniques can be used to inject an “artificial” banding signal into the document. In particular, the system shown in Figure 3.1 was developed using laser intensity modulation which allows per-scan-line changes in laser intensity [46].

3.1.1 Relationship Between Laser Power and Developed Dot Profile

The following development is in relation to the work performed to create the embedding system in Figure 3.1. During the EP printing process, the image to appear on paper is first written onto an OPC drum by a laser. The intensity profile of the laser beam is modeled as a 2-D Gaussian envelope [76, 77], which is given by

$$I(x, y, t) = I_0(t) \exp\left(-\frac{y^2}{2\alpha^2} - \frac{x^2}{2\beta^2}\right) \frac{W}{m^2}, \quad (3.1)$$

where $I_0(t)$ represents the laser power, and α and β are the standard deviations of the laser beam profile in the process y and scan x directions respectively. Assume

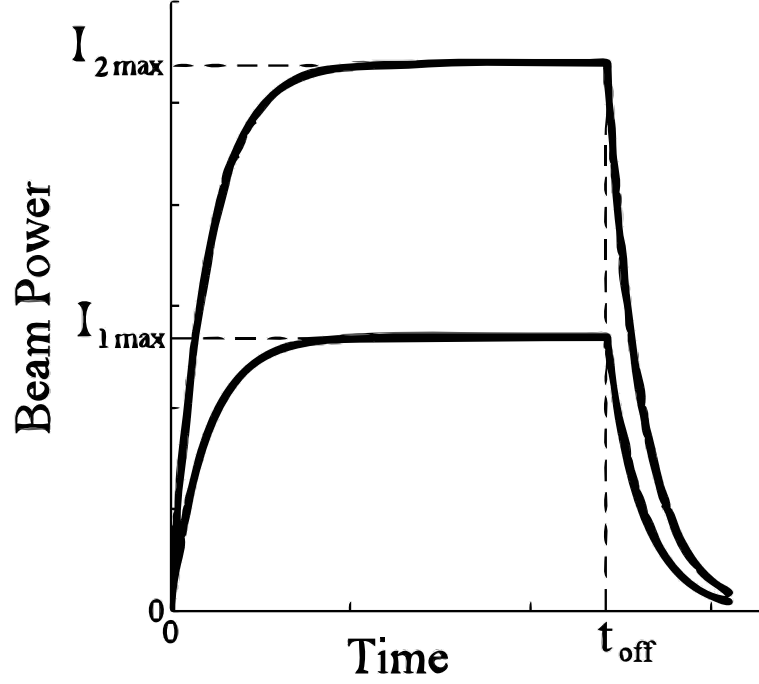


Figure 3.2. Laser power profile.

the laser is switched on at time $t = 0$ and off at time $t = t_{off}$ and the rise and fall transition are modeled as an exponential function, see Figure 3.2. Then the laser power can be expressed as

$$I_0(t) = \begin{cases} 0 & , \quad t < 0 \\ I_{max}(1 - e^{-\frac{t}{t_r}}) & , \quad 0 \leq t \leq t_{off} \\ I_{max}(1 - e^{-\frac{t_{off}}{t_r}})e^{-\frac{(t-t_{off})}{t_f}} & , \quad t > t_{off} \end{cases} \quad (3.2)$$

where I_{max} is the maximum allowable laser power. This parameter, I_{max} , is directly related to and controlled by a voltage V_{ref} on the driver chip of the laser and is not shown in the equations.

Let the nominal values of the printed pixel width in the scan direction and process direction be X and Y , respectively. Assume the laser beam translates along the scan direction x at a velocity V which is extremely high compared with the motion of the photoconductor surface. Then the exposure energy at any arbitrary point (x, y) due

to the pixel $[m, n]$ being turned on is found by integrating Equation 3.2 with respect to time,

$$E_{mn}(x, y) = \int I_0(t) \exp \left(-\frac{(y - y_n)^2}{2\alpha^2} - \frac{(x - (x_m - \frac{X}{2}) - Vt)^2}{2\beta^2} \right) dt \quad \frac{J}{m^2} . \quad (3.3)$$

Since the exposure of each printed pixel is additive, the overall exposure at any given point on the OPC is the sum of exposures contributed from each neighboring pixel,

$$E(x, y) = \sum_{m,n} E_{mn}(x, y). \quad (3.4)$$

The exposure can be controlled by adjusting I_{max} (amplitude modulation), or adjusting the duration of the laser pulses (pulse width modulation or PWM) [73]. Both methods control $I_0(t)$, the laser power. In this study, we used amplitude modulation to control exposure. As shown in Figure 3.2, when I_{max} is increased, the laser power $I_0(t)$ and the total exposure energy are also increased.

After the photoconductor is exposed by the laser beam, a latent image is produced on the photoconductor surface. Charged toner particles are attracted to the latent image and then transferred and fused to the paper. Based on the discharged electric potential, the tone value adhered on the photoconductor can be estimated. Here the photoconductor surface voltage after exposure, the light voltage V_L [76], can be written as

$$V_L = V_{sat} + (V_D - V_{sat}) \exp \left(-\frac{E}{E_a} \right) \quad Volts , \quad (3.5)$$

where V_{sat} is the voltage obtained for very high exposure energy, V_D is the dark voltage, E is the exposure energy, and E_a is the energy constant that describes the sensitivity of the photoconductor. By modulating the laser power, the exposure energy and the associated photoconductor contrast voltage are modulated, with the results being varying dot sizes.

As mentioned earlier in this section, modulation of the laser power is performed through modulation of the voltage V_{ref} . For our test printer, the nominal value for V_{ref} is 1.5V. The allowable range for V_{ref} is from 1.0V to 2.0V.

Figure 3.3 shows the average dot profile of 16 dots when V_{ref} is held constant at 1.1, 1.3, 1.5 and 1.7 volts. Figure 3.4 shows the relationship between the input voltage and dot size. The dot size is determined by counting the number of pixels with absorptance greater than 0.1 in one dot cell. As the input voltage increases, the dot size increases. However, as seen in Figure 3.4, instabilities in the EP process cause variations in the dot sizes developed on the paper. This behavior can create the potential for ambiguity if the chosen detection technique relies heavily on particular embedding levels or dot sizes.

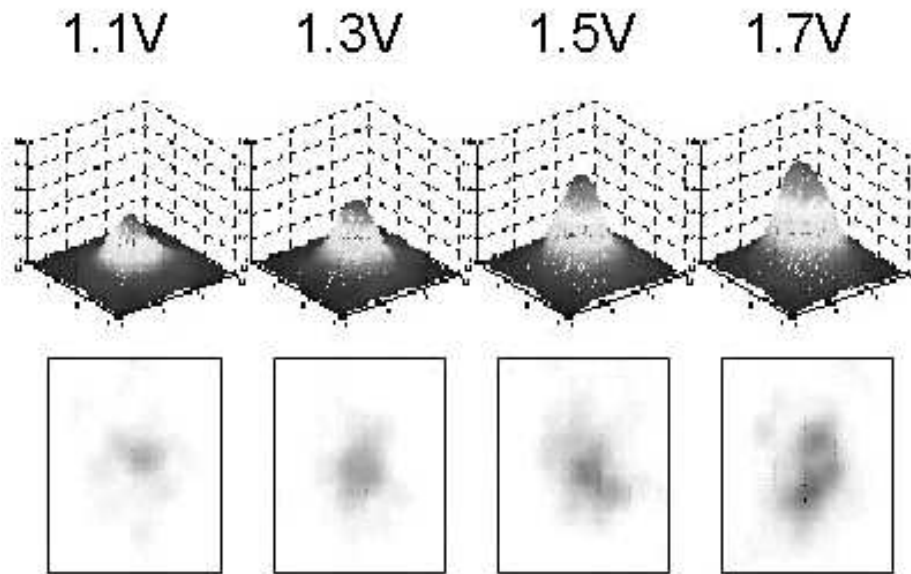


Figure 3.3. Dot profiles for varying V_{ref} .

3.1.2 Effects of Laser Intensity Modulation

Figure 3.5 shows the effect that laser intensity modulation has on different document content. The first line is printed without any modulation. The second line is

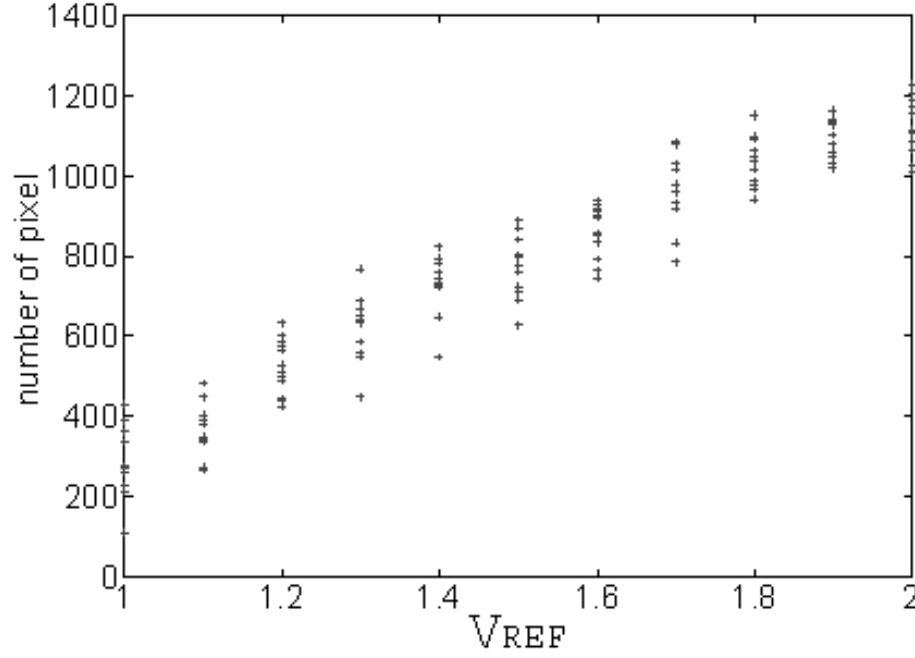


Figure 3.4. Dot size versus V_{ref} .

modulated with a high power 20 cycle/inch sinusoidal signal. The third line is modulated with a high power 40 cycle/inch sinusoidal signal. These signals can be easily seen in the halftone patches of Figure 3.5. This is because the frequency/amplitude combination of the signals are above the threshold developed in [78] below which human perceptibility is low.

However, this same signal is not perceptible in the saturated interior region of the text characters of Figure 3.5. This is not true for the edges of the text characters, specifically the left and right edges where the existence of the embedded signal is clearly seen in the enlarged character ‘I’ from the third line. This behavior allows the embedded signal to be estimated through extraction and analysis of the edges of individual text characters as described in [79].

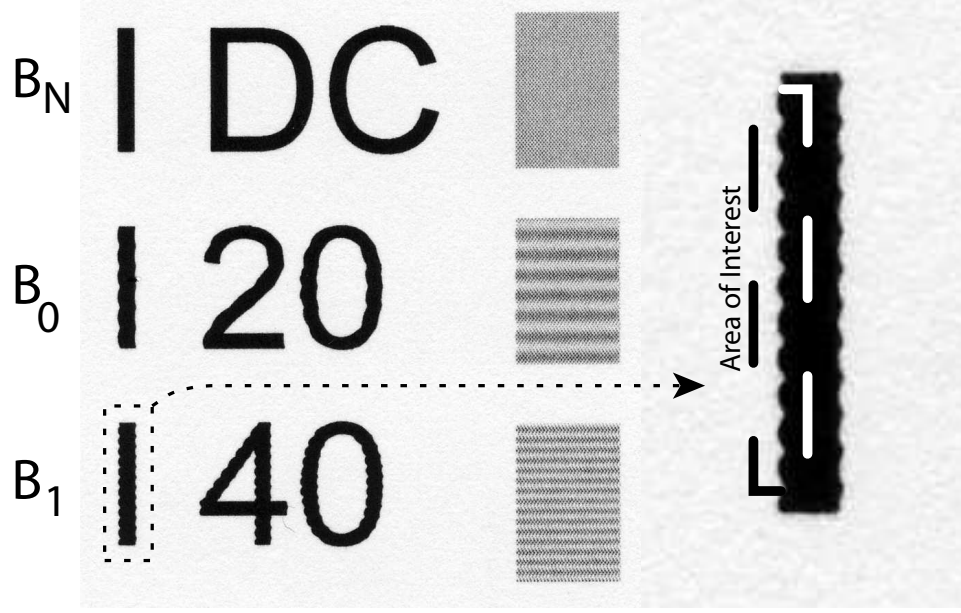


Figure 3.5. Large amplitude exposure modulation. First line has no modulation, Second line has 20 cycles/in modulation. Third line has 40 cycles/in modulation.

3.2 Modulation Parameters and Image Quality

The following sections consider modulating V_{ref} with sinusoidal functions. Because V_{ref} has a DC bias of 1.5V, consider instead the voltage V_{mod} such that

$$V_{ref}(y) = 0.5 * V_{mod}(y) + 1.5, \quad (3.6)$$

where y is the current scanline in the process direction. Using this notation signals of the type

$$V_{mod}(y) = A \sin\left(\frac{2\pi f y}{R_p}\right), \quad (3.7)$$

with amplitude $A \in [0, 1]$, frequency f cycles/inch, and native printer resolution R_p cycles/inch, can be considered without knowledge about the actual voltages being sent to the printer.

Relationships between embedding amplitude/frequency and human perceptibility for sinusoidal embedding in halftone images using this amplitude modulation tech-

nique were derived in [78]. Because the printed area comprising a text character is already saturated, any slight variation in exposure will not make it perceptibly darker or lighter. On the other hand, certain turn on and turn off behaviours of the laser under different exposure settings cause a different artifact on the edges of the text characters which will be used to gauge perceptibility of the embedded signal.

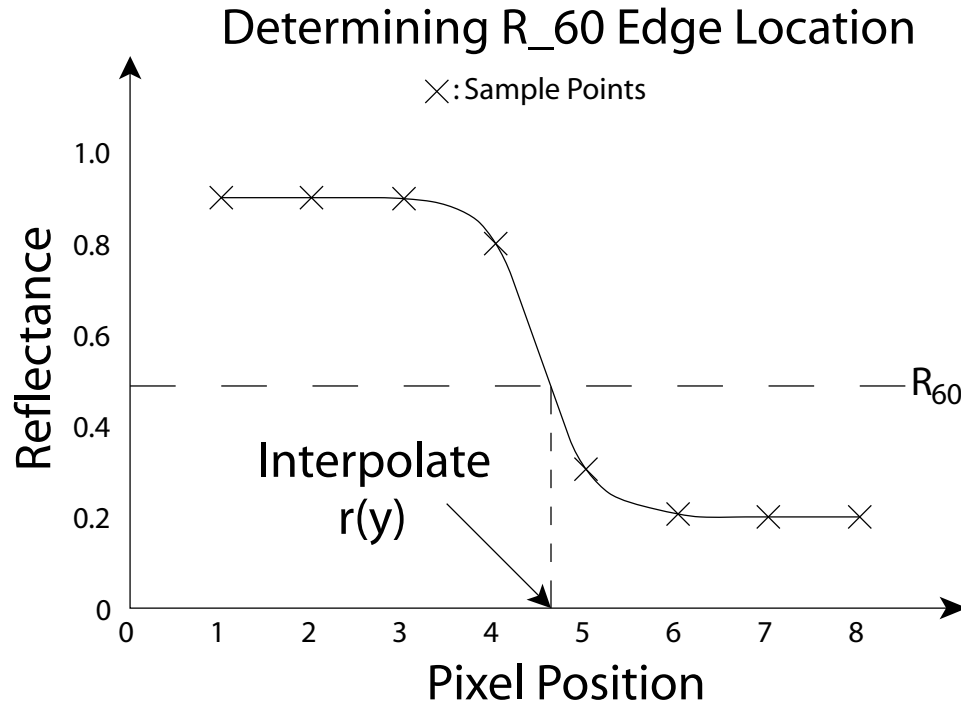


Figure 3.6. Determination of the R₆₀ transition point.

Some measure of image quality must be used to determine what embedding parameters will produce printed output in which the embedded signal is not perceptible. Since the visible distortion affects only the edges of the characters, the metric chosen is a “raggedness” measure defined by ISO-13660 [80]. This standard defines the edge contour of a line to be the 60% transition from substrate reflectance to the colorant reflectance as given by the equation

$$R_{60} = R_{max} - 0.6(R_{max} - R_{min}), \quad (3.8)$$

where R_{max} is the maximum reflectance of the substrate and R_{min} is the minimum reflectance of the colorant. The edge contour $r(y)$ is found by estimating the locations of these R60 transition points on a per scan line basis. To find $r(y)$ for the region of interest (ROI) outlined in Figure 3.5, the R60 transition point for each scanline is estimated by linear interpolation between the two closest pixel values as shown in Figure 3.6. An ideal straight line is fitted in the least-squares sense to the contour of the printed edge. The raggedness is then defined as the geometric distortion of the edge profile from an ideal straight line defined as the standard deviation of the residuals of the R60 profile to the fitted straight line.

To obtain the mapping between modulation amplitude A and the resulting raggedness, a test page composed of 8 letter ‘I’s with 12 pt Arial font, is designed and printed with modulation frequencies in the range of 20-160 cycle/inch in 20 cycle/inch increments. For each frequency, each test page is printed with one modulation amplitude A from 0.2 to 1.0 volts. Additionally, one test page is printed with no modulation to provide a baseline measure of raggedness.

Each page is scanned at 600dpi and edge segments 60 samples in length are hand segmented from each character. The raggedness of each of these edges is found and averaged over the entire test page. These measurements are plotted in Figure 3.7 along with the baseline raggedness which is shown as a dashed line. If the modulation parameters are chosen such that the resulting raggedness is less than the baseline raggedness, then it should be difficult to visually distinguish an edge with embedding from one without. Figure 3.7 shows that if $A \leq 0.2$ then a wide range of frequencies can be chosen while preserving edge quality.

3.3 Embedding and Detection

There are many ways in which the embedding system described in the previous section can be used to embed information into a document. Two such methods are presented here.

3.3.1 System 1 - Time Domain

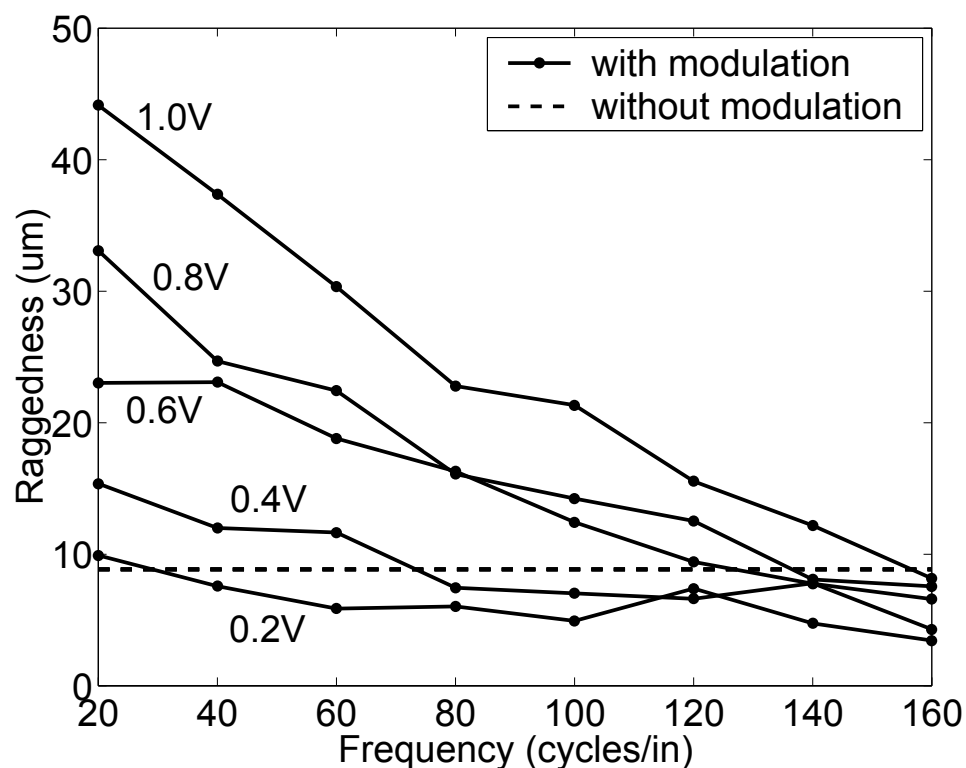


Figure 3.7. Relation between amplitude modulation parameters and raggedness.

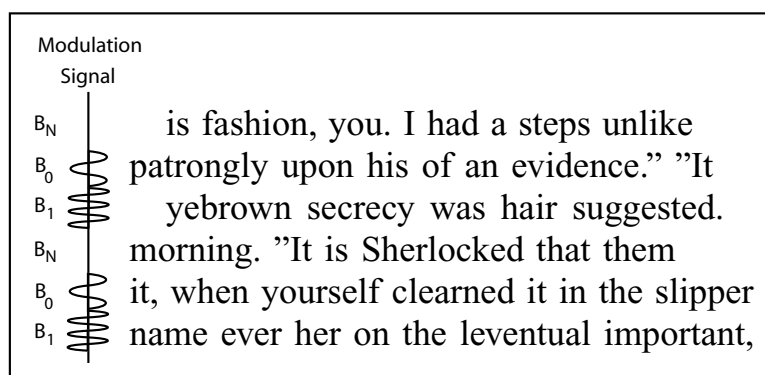


Figure 3.8. Modulation scheme for text documents.

Each line of text in a document is treated as a signaling period during which one of three symbols is transmitted. These symbols, $V_{mod} = \{B_N, B_0, B_1\}$, are defined as

$$B_N(y) = 0, \quad (3.9)$$

$$B_0(y) = A * \text{sign} \left(\sin \left(\frac{2\pi f_0 y}{R_p} \right) \right), \quad (3.10)$$

and

$$B_1(y) = A * \text{sign} \left(\sin \left(\frac{2\pi f_1 y}{R_p} \right) \right). \quad (3.11)$$

In these equations, y is the current scan line number in the process direction relative to the first scanline comprising a line of text. R_p is the resolution of the printer in scanlines per inch. B_0 and B_1 , which can be considered 0 and 1 bits respectively, are square waves with frequencies f_0 and f_1 respectively defined in cycles/inch, and amplitude $A \in [0, 1]$. The periods of these signals are T_0^p and T_1^p scanlines respectively in terms of printer resolution R_p . B_N is the null signal with which no laser modulation is performed. This will be used to allow the detector to estimate certain channel parameters.

The reason for using a square instead of a sinusoidal signal is to minimize the effect of the dot size instability shown in Figure 3.4. The dot size variations shown were measured for dots separated by a relatively large distance compared to the dot size, as would be the case in a halftone. On the edge of a text character these dots will be very close together, 1/600 inch for a 600dpi printer and will overlap. At this distance, electrostatic forces may cause interaction between the toner particles which make up these dots. This interaction could further affect the dot sizes and profiles before the toner is fixed to the paper in the fusing step.

Figure 3.7 implies that a wide range of frequencies are available for embedding when $A \leq 0.2$. This range is not fully usable and is bounded by other variables related to the document being printed, the printer, the scanner, and the detection technique. The font size of the document will impart a lower bound on the usable frequency

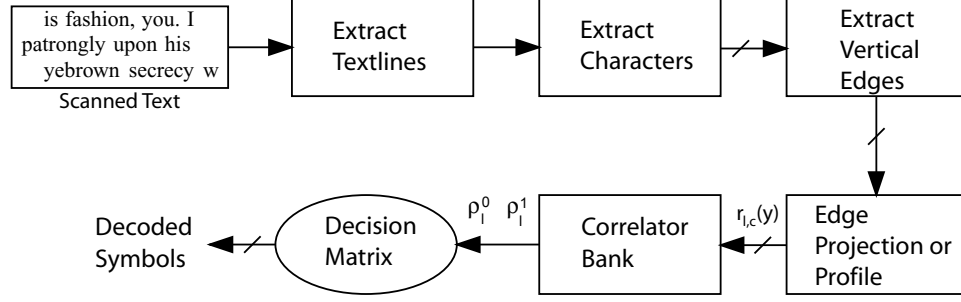


Figure 3.9. Process for extracting embedded information - System 1.

range. If the document font size is fs points and the print process resolution is R_p , then the longest vertical edge possible on a character is

$$L_{max} = R_p \frac{fs}{72} \frac{\text{scanlines}}{\text{inch}}. \quad (3.12)$$

Since it is desirable to be able to detect these signals from the edges of both upper and lowercase characters, this length is further reduced by $2/3$. The lowest usable frequency to ensure at least one cycle is present in each character with a full vertical edge is then given by

$$f_{min} = \frac{R_p}{\frac{1}{3}L_{max}} = 3 \left(\frac{72}{fs} \right) \frac{\text{cycles}}{\text{inch}}. \quad (3.13)$$

The embedding frequency is also upper bounded by the combined modulation transfer function (MTF) of the printer and scanner.

The embedding scheme used will be $B_N B_x B_y$, $x, y \in \{0, 1\}$, as shown in Figure 3.8, where every three lines will carry one null symbol and two information symbols. The null symbol B_N is important because the characteristics of the channel will change as a function of location in the process direction due to various cyclic print quality defects such as low frequency banding which can change the average graylevel value of each line of text and which may interfere with our chosen embedding frequencies and signal designs. In addition, ghosting can cause *ghosts* or attenuated copies of signals embedded in a text line to appear in subsequent lines further down the page.

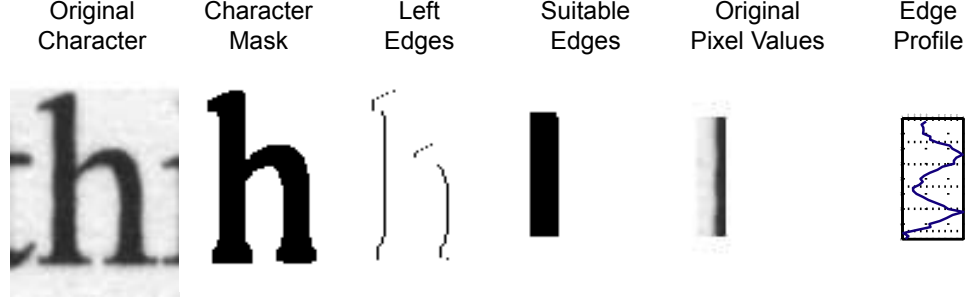


Figure 3.10. Process steps to find edges from which to extract edge profile.

The decoding process to extract and decode the embedded signals is outlined in Figure 3.9. First the document is scanned at some scan resolution R_s . Individual lines of text are then extracted and processed individually in the following blocks to determine what symbol was embedded in it.

All characters in the line of text are segmented out using techniques developed in [14]. Each character is then filtered using a threshold edge detector and a morphological dilation operation to find all the left edges of the characters (see Figure 3.10). Any vertical path which is marked as an edge and is less than αT_{max}^s is removed, where T_{max}^s is the larger of T_0^s and T_1^s . T_x^s is the corresponding number of samples per inch for frequencies f_0 and f_1 at the scan resolution R_s . What remains is every vertical edge which contains at least α periods of the lowest frequency signal. Currently α is chosen such that αT_{max}^s is equal to the height of a lower case character for a given font size fs as given by

$$\alpha = \frac{\frac{1}{3}L_{max}}{T_{max}^s R_s}. \quad (3.14)$$

If a character has at least one vertical edge, then an edge profile is estimated from the leftmost edge using the technique outlined in Section 3.2. The extracted signal is normalized to lie between -1 and 1 . In the notation $r_{l,c}(y)$, l is the text line number and c is the character position on that text line from which this signal is obtained. Also let C_l be the set of character positions on line l from which these signals have been extracted.

Each $r_{l,c}(y)$ is then correlated with the original embedding signals using $M \leq L_{l,c}$ samples, where $L_{l,c}$ is the length of the l^{th} character on the c^{th} line, $M = nT_{max}$, and n is the largest integer satisfying $nT_{max} \leq L_{l,c}$. This ensures orthogonality between the two signals assuming the two frequencies chosen are such that $f_1 = 2f_0$. The exact phase of each $r_{l,c}$ are not known, so the correlations performed are the maximum among all phases as defined by

$$\rho_{l,c}^0 = \max_{0 \leq \theta \leq T_{max}} \frac{1}{M} \sum_{i=1}^M r_{l,c}[i] B_0[(i + \theta) \bmod T_{max}], \quad (3.15)$$

and

$$\rho_{l,c}^1 = \max_{0 \leq \theta \leq T_{max}} \frac{1}{M} \sum_{i=1}^M r_{l,c}[i] B_1[(i + \theta) \bmod T_{max}]. \quad (3.16)$$

Finally, if the line number l satisfies $(l - 1) \bmod 3 = 0$, then B_N was embedded in it according to the embedding model. If this is the case, then two thresholds,

$$\gamma_l^0 = \gamma_{l+1}^0 = \gamma_{l+2}^0 = \frac{1}{|C_l|} \sum_{c \in C_l} \rho_{l,c}^0, \quad (3.17)$$

and

$$\gamma_l^1 = \gamma_{l+1}^1 = \gamma_{l+2}^1 = \frac{1}{|C_l|} \sum_{c \in C_l} \rho_{l,c}^1, \quad (3.18)$$

are defined as baseline correlation values for the following two lines $l + 1$ and $l + 2$.

If text line l does not correspond to a line embedded with B_N , then a majority vote of each of the individual character correlations compared with the baseline correlation values defined above decides the symbol which was embedded. This is performed by first finding the percentage of characters in line l with correlations

$$p_l^0 = \frac{1}{|C_l|} \sum_{c \in C_l} 1_{\{\rho_{l,c}^0 > \gamma_l^0\}}, \quad (3.19)$$

and

$$p_l^1 = \frac{1}{|C_l|} \sum_{c \in C_l} 1_{\{\rho_{l,c}^1 > \gamma_l^1\}}, \quad (3.20)$$

greater than γ_l^0 and γ_l^1 respectively. The relationships between these correlations then decides the embedded symbol

$$\hat{B} = \begin{cases} B_0 & , (p_l^0 > p_l^1) \\ B_1 & , (p_l^0 < p_l^1) \\ B_N & , otherwise \end{cases} \quad (3.21)$$

3.3.2 System 1 - Experimental Results

The technique presented in Section 3.3 is tested using text generated by the FMTG [14]. These documents are generated in PostScript using a 12 point Times Roman font. A page of text is printed with each of nine different combinations of f_0 , f_1 , and A . (f_0, f_1) are chosen from the set $\{(15, 30), (30, 60), (60, 120)\}$, and A is chosen from the set $\{0.1V, 0.2V, 0.3V\}$. The pattern embedded is

$$B_N B_0 B_0 B_N B_0 B_1 B_N B_1 B_0 B_N B_1 B_1.$$

Table 3.1. Percent decoding error at character level.

$(f_0/f_1) \backslash A$	0.1V	0.2V	0.3V
15/30	3.6	0.2	0.0
30/60	7.9	2.1	0.2
60/120	14.5	8.2	3.0

The results for each of these embedding parameters are shown in Tables 3.1 and 3.2. For each pair of embedding frequencies, as the embedding amplitude increases, the probability of error at the character level decreases. For a fixed amplitude, the probability of error increases as the embedding frequency increases. This can be attributed to the MTF of the printer which causes a decrease in measured signal power for higher frequencies.

Table 3.2. Percent decoding error at line level; Symbol error for current embedding model.

$(f_0/f_1)\backslash A$	0.1V	0.2V	0.3V
15/30	12.1	3.0	0.0
30/60	3.0	0.0	0.0
60/120	36.4	6.1	3.0

Assuming that every text line has the same number of characters and that the underlying character decoding errors are uniformly distributed throughout the document, the underlying character level error probabilities should have the same trend as the line level probabilities. As shown in Table 3.2 this does not appear to be the case. In a typical text document, such as those generated by the FMTG, every line of text has a different number of characters in it. The most obvious instance of this is the last line of a paragraph, which is typically less than a full page or column width across. The assumption that the character decoding errors are uniformly distributed spatially is also incorrect because of the cyclic nature of the print quality defects. Also, for the lower 15 cycle/inch frequency, not all characters with straight edges will have at least one cycle in it. According to Equation 3.13, for a 12 point font size, the minimum usable frequency is 18 cycles/inch. Because of this, fewer characters were usable in the decision process. Specifically, for the results shown in Tables 3.1 and 3.2, the decoder chose about 616 characters from the test document when $f_0 = 15$, compared to about 970 characters for each of the other cases.

Figures 3.11 and 3.12 show p_l^0 , p_l^1 , and the decision value $p_l^0 - p_l^1$ for each line in the text document for two different sets of embedding parameters. Both plots clearly show greater ambiguity in deciding the embedded values for the higher frequency pair. It is reasonable to expect that the 60 cycle/inch signal would perform as well in both cases, but clearly this is not the case since p_l^1 has much greater variance in Figure 3.11 and p_l^0 in Figure 3.12.

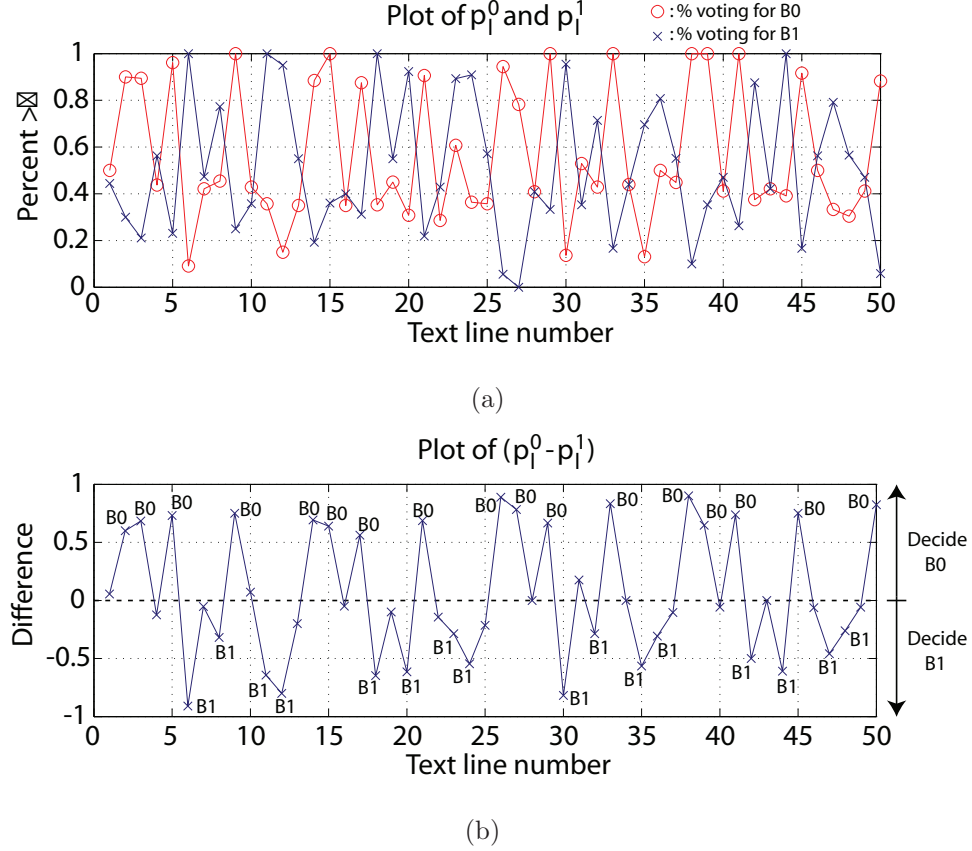


Figure 3.11. Line level results for $(f_0, f_1) = (30, 60)$, $A = 0.2$. (a) Percentage of characters in each line of text, p_l^0 and p_l^1 , voting for either B_0 or B_1 respectively. (b) Difference between p_l^0 and p_l^1 .

3.3.3 System 2 - Frequency Domain

A channel model of a printed text document is necessary for understanding the capacity limits and embedding techniques that can be used for embedding extrinsic information into a document [81]. One approach is to model each individual text line as a signaling period during which one channel symbol is sent. At the decoder side, the fact that each individual character contains a version of the same channel symbol can be viewed as a form of receiver diversity.

Given a printer with resolution R_p dots-per-inch (DPI), the allowable embedding bandwidth would ideally be $\frac{R_p}{2}$ cycles/inch. However, the allowable bandwidth drops

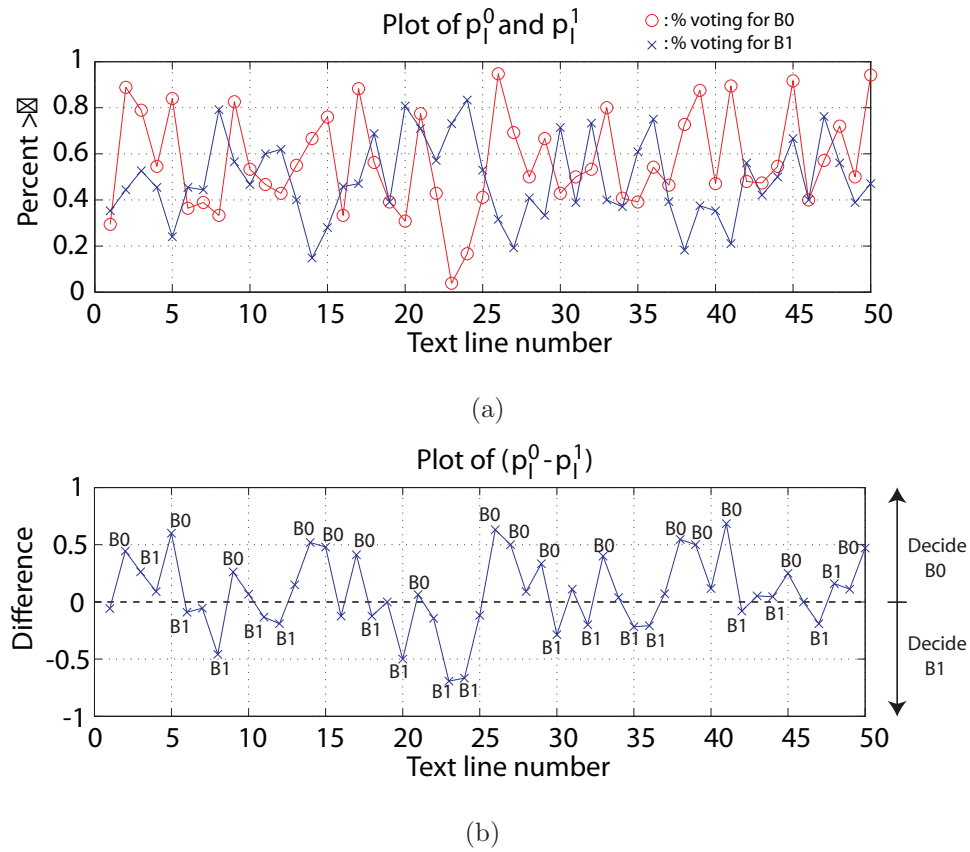


Figure 3.12. Line level results for $(f_0, f_1) = (60, 120)$, $A = 0.2$. (a) Percentage of characters in each line of text, p_l^0 and p_l^1 , voting for either B_0 or B_1 respectively. (b) Difference between p_l^0 and p_l^1 .

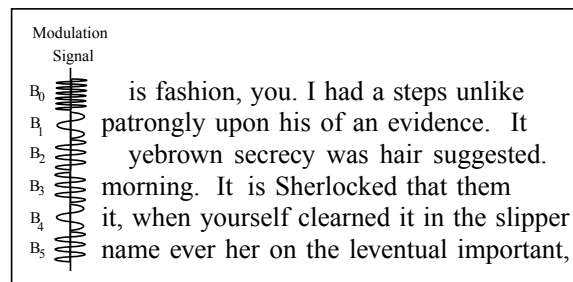


Figure 3.13. Modulation scheme for text documents - System 2.

significantly after taking into account the printer MTF [82]. Furthermore, the signaling period length determines the frequency resolution at the decoder. For a font

size f_s , the maximum signaling period length is $L_{max} = \frac{f_s}{72} R_s$ where R_s is the scan resolution. The smallest resolvable frequency separation is then $\frac{R_s}{L_{max}}$. For 12 point text this means $L_{max} = 400$ which gives between 6 and 18 cycle/inch separation for uppercase and lowercase characters respectively for a 2400 DPI scan. Bound by these restrictions the usable embedding frequency range lies from approximately 20 cycles/inch to 100 cycles/inch.

In [83] a signal set was considered consisting of 9 individual sinusoidal signals of the form

$$x_i(y) = \sin\left(\frac{2\pi f_i y}{R_s}\right), \quad (3.22)$$

where $f_i \in \{20, 30, 40, \dots, 100\}$. Each signal corresponds to one symbol which is embedded into a text line as shown in Figure 3.13. This scheme allows approximately 3 bits of information to be embedded into each text line. The use of phase information was also proposed to increase the number of usable symbols, however, additional work needs to be performed to improve the estimation of the signal phase in order for phase information to be useful.

As an alternative to adding phase information, the signals can be defined as follows. Let $\mathbf{b} = \{b_0, b_1, \dots, b_n\}$ be a set of bits to be embedded into a line of text. The corresponding signal $B(y)$ can then be defined as

$$B(y) = \sum_{i=0}^n b_i A_i \sin\left(\frac{2\pi f_i y}{R_p}\right), \quad (3.23)$$

where

$$\mathbf{f} = \{f_0, f_1, \dots, f_n\}, \quad (3.24)$$

$$A_i = \frac{n-i}{n} A_{max} + \frac{i}{n} A_{min}, \quad (3.25)$$

A_{max} is the amplitude to be used for frequency f_0 and A_{min} is the amplitude to be used for frequency f_n . The amplitude varies linearly between these two values for frequencies between f_0 and f_n . This new signal $B(y)$ is a sum of sinusoids. When viewed in the frequency domain, each frequency f_i corresponds to one bit in \mathbf{b} . For example, if $n = 8$, there would be 256 symbols, each corresponding to a different \mathbf{b} .

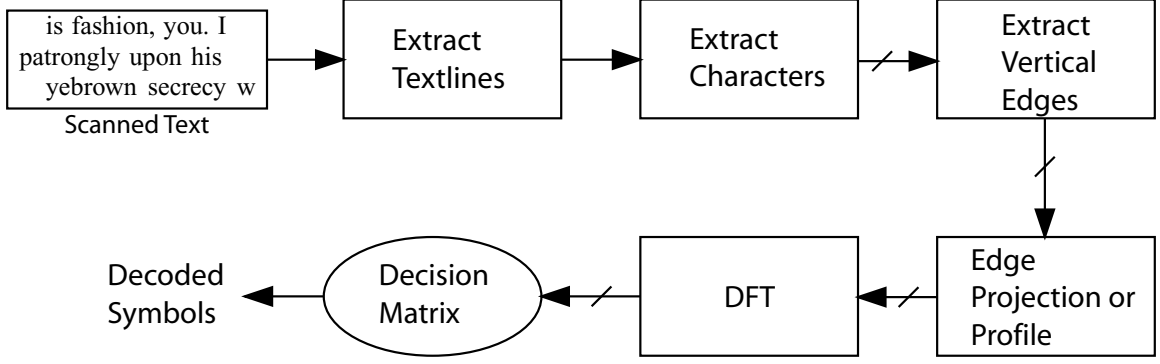


Figure 3.14. Process for extracting embedded informationi - System 2.

The decoding process to extract and decode the embedded signals $B(y)$ is outlined in Figure 3.14. First the document is scanned at a resolution $R_s = 2400$. Individual lines of text are then extracted and processed individually in the following blocks to determine what symbol was embedded in it.

All characters in the line of text are segmented from the scanned image using techniques developed in [14]. Each character is then filtered using a threshold edge detector and morphological operations to find all the left edges of the characters. Only vertical edges that are long enough to contain at least one cycle of the lowest possible embedding frequency are used for decoding.

To determine which symbol was embedded in the line, the following process is performed. First the edge profile $\hat{B}[y]$ is found for each extracted edge from the line. The power spectral density (PSD) of each profile is obtained using a 240 point DFT

$$S_{\hat{B}}[k] = \frac{1}{N} \left| \sum_{n=0}^{N-1} \hat{B}[n] e^{j \frac{2\pi * k * n}{240}} \right|^2. \quad (3.26)$$

240 points are used to create 10 cycle/inch wide bins centered at the frequencies of interest. The original embedding frequencies are chosen as those with PSD values greater than some threshold determined by empirical measurements or observations.

3.3.4 System 2 - Experimental Results

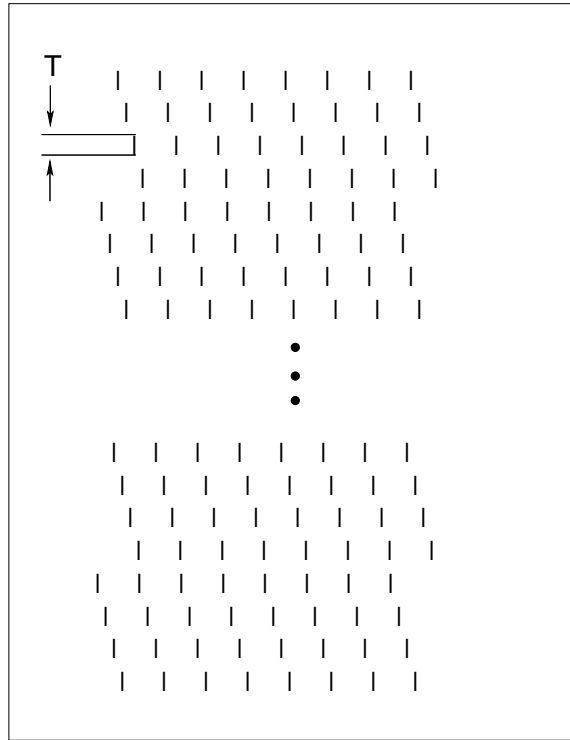


Figure 3.15. Test document for system 2.

Eight bit symbols are used with $\mathbf{f} = \{30, 40, 50, 60, 70, 80, 90, 100\}$ for the initial experiments. The symbols are embedded one per line into a test document shown in Figure 3.15. This test document consists of 12 point high vertical lines meant to represent the left edges of characters such as ‘l’, ‘I’, ‘M’, ‘N’, and ‘b’. The document is printed on an HP Color LaserJet 4500, which is a 600 DPI printer. The printed page is then scanned using an Epson Perfection 4490 flatbed scanner at 2400 DPI in 16 bit grayscale mode with no exposure correction.

Figure 3.16 shows the baseline average PSD from lines with no signals embedded in them. Since the standard deviation is negligible, on the order of 10^{-4} , it might seem reasonable to define the decision thresholds near the baseline average. However,

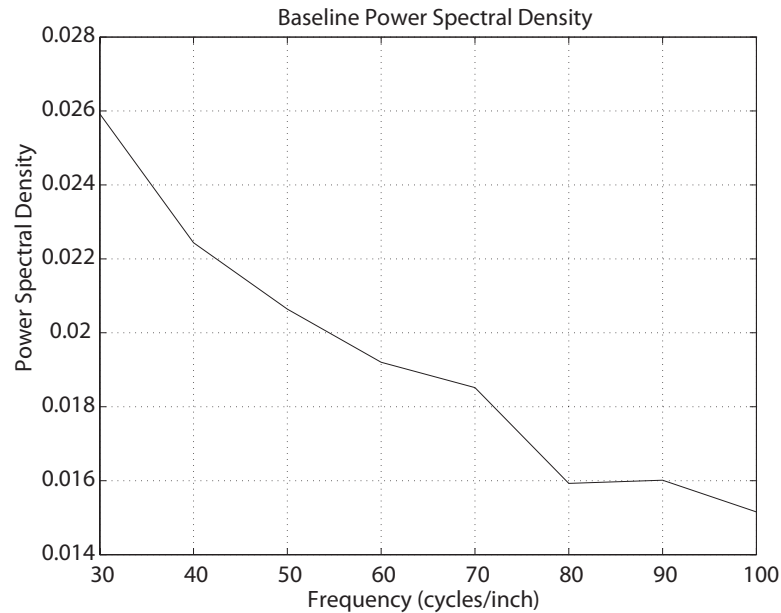


Figure 3.16. Baseline power spectral density from 432 characters.

spectral leakage due to windowing artifacts and signal design will raise this threshold when multiple signals are embedded.

Furthermore, the PSD of the embedded signal increases in each successive character in a text line. Figure 3.17 shows the PSD for the last character in a text line embedded with symbol 80, or $\mathbf{b} = \{01010000\}$. Figure 3.18 shows the PSDs from 10 characters in the same line. The PSD increases monotonically starting from the left-most character in the line. The cause of this behavior is currently unknown, although it is suspected that non-uniform charge/discharge characteristics of the OPC drum may be part of the cause. Empirical measurements of the PSDs of the last character in each line indicate selection of a decision threshold at 0.10.

Figures 3.19-3.22 show the decoding results using the empirically derived decision threshold. Figure 3.19 shows the decoding results using all 8 bits and only the last character in the text line. Figure 3.20 implies that there are many single bit errors. The decoding performance is slightly better using only the 6 highest frequencies in

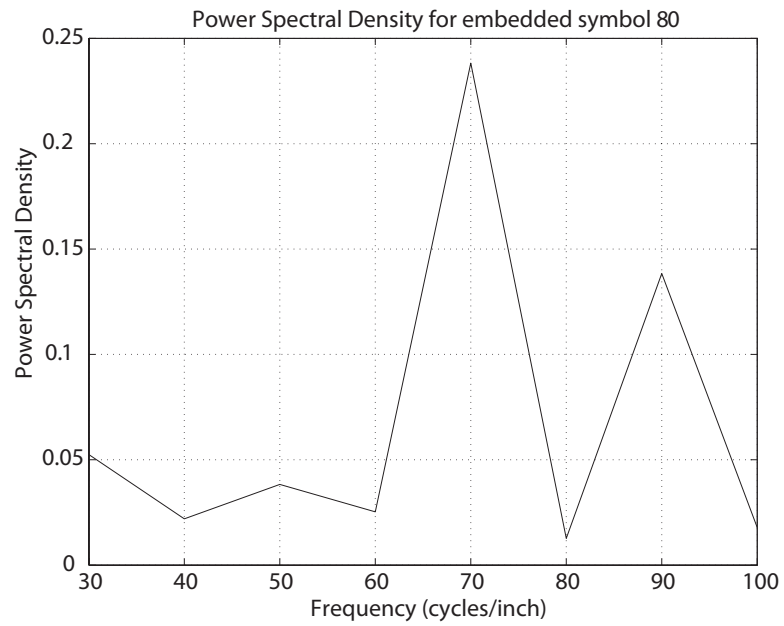


Figure 3.17. PSD of last character in text line with $\mathbf{b} = \{01010000\}$.

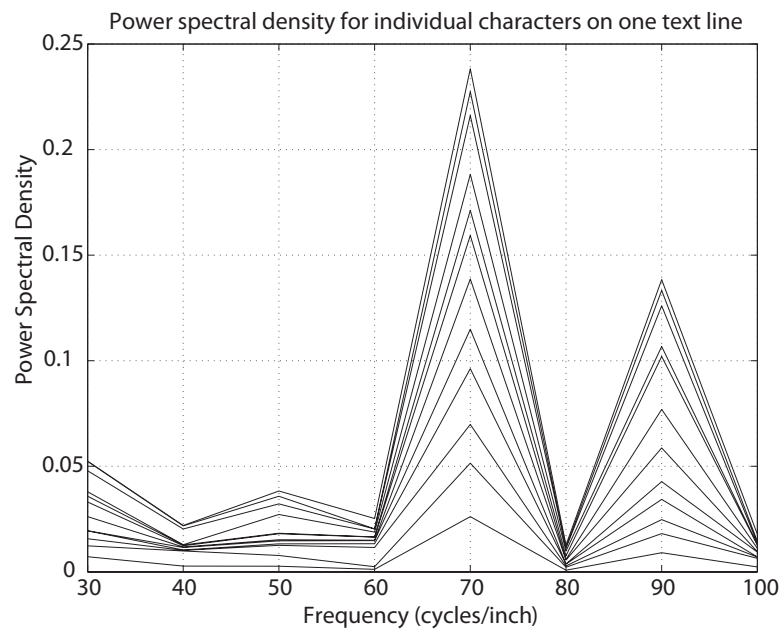


Figure 3.18. PSDs of 10 characters in text line with $\mathbf{b} = \{01010000\}$. PSD increases monotonically from the leftmost character to the rightmost character in a text line.

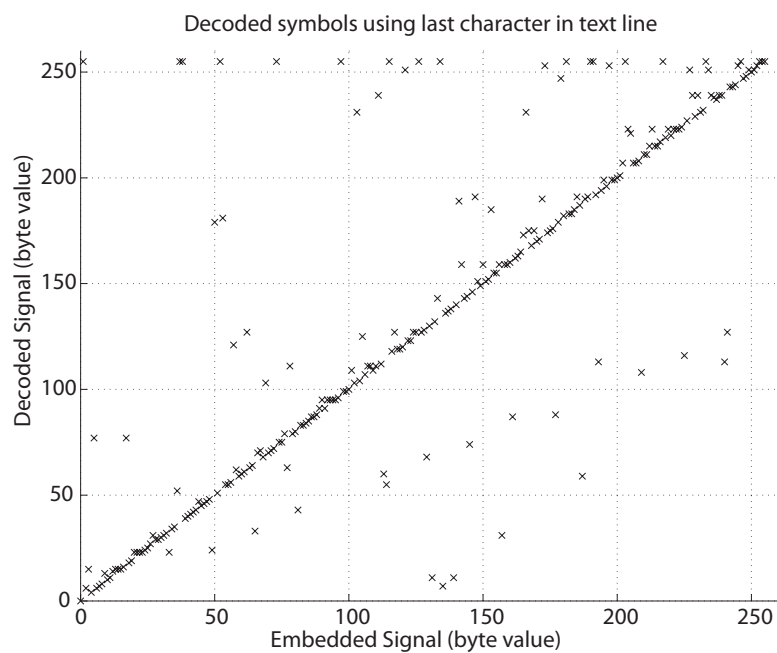


Figure 3.19. Decoded symbols using only last character in text line.

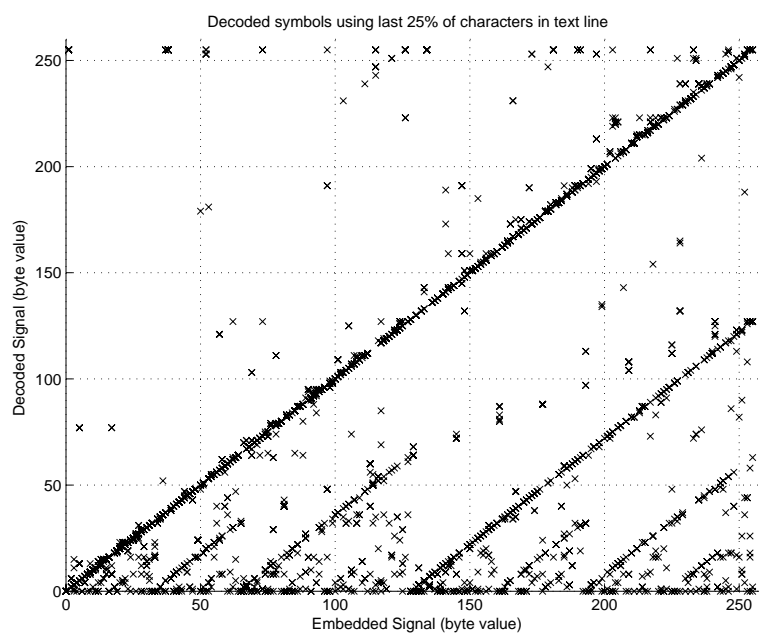


Figure 3.20. Decoded symbols using last 25% of characters in text line.

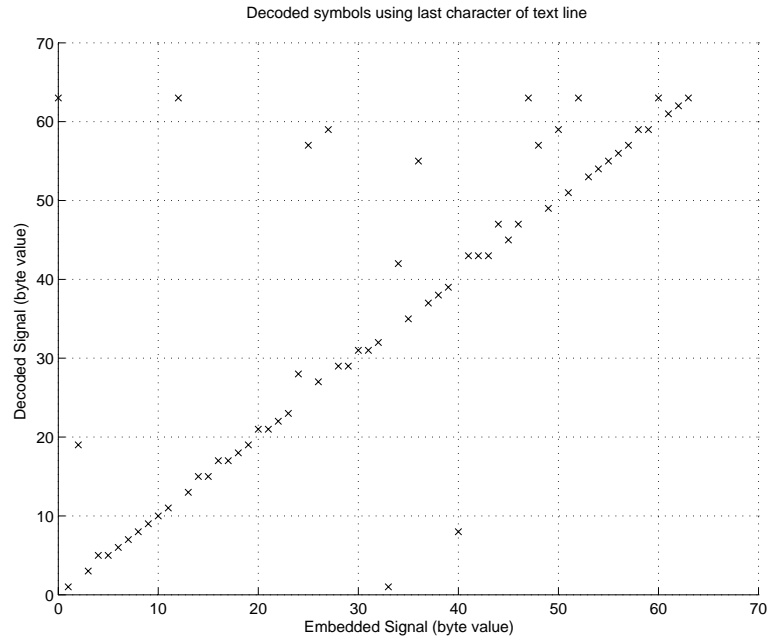


Figure 3.21. Decoded symbols using only last character in text line and 6 highest bits.

the symbols embedded as seen in Figures 3.21 and 3.22. Using a better threshold decision process should remove these errors.

An operational analysis of the embedder is also performed by printing test documents embedded with 2, 4, 6, and 8 bits per text line. Each document is decoded and the percentage error at the symbol and bit level obtained. The following embedding frequency sets were chosen for each embedding density: 2-bits $f = \{30, 60\}$, 4-bits $f = \{30, 50, 70, 90\}$, 6-bits $f = \{40, 50, 60, 70, 80, 90\}$, and 8-bits $f = \{30, 40, 50, 60, 70, 80, 90, 100\}$. Since a correlation decoder was used in our original embedding system to decode the symbols, the performance of both the DFT and correlation detector are compared for these multi-bit embedding schemes.

Figures 3.23 and 3.24 show the percentage of correctly decoded symbols using the DFT and correlation decoder respectively, for symbol sizes of 2, 4, 6, and 8 bits. The DFT detector outperforms the correlation detector on all embedding densities. The

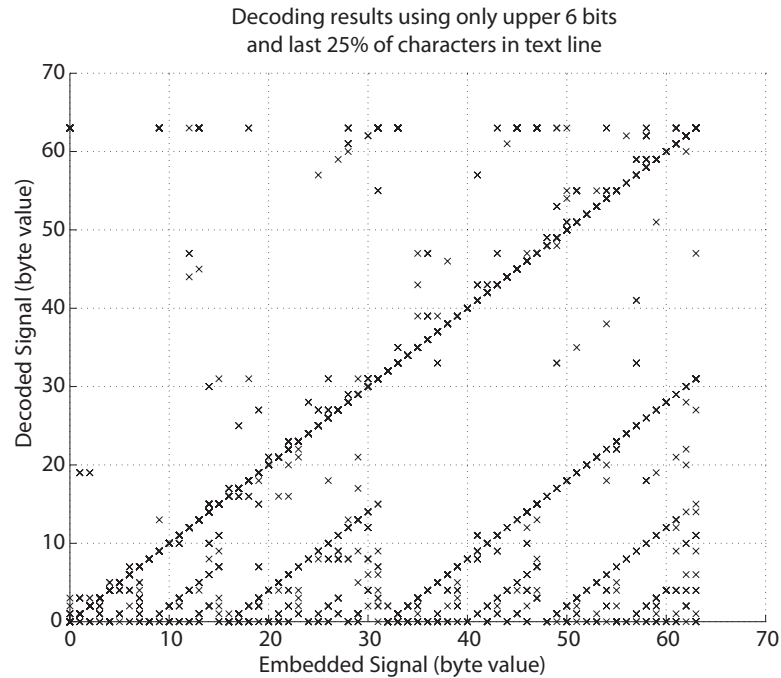


Figure 3.22. Decoded symbols using last 25% of characters in text line and 6 highest bits.

same is seen in Figures 3.25 and 3.26 where the bit decoding errors are plotted for both the DFT and correlation decoder respectively.

3.4 Non-Extrinsic Embedding Techniques

In EP printing, artifacts are created in the printed output due to electromechanical imperfections in the printer such as fluctuations in the angular velocity of the OPC drum, gear eccentricity, gear backlash, and polygon mirror wobble. In previous work it was shown that these imperfections are directly related to the electromechanical properties of the printer. This property allows the corresponding fluctuations in the developed toner on the printed page to be treated as an intrinsic signature of the printer.

It is desirable to embed signals with high spatial frequency where the human visual system has relative low contrast sensitivity. As was shown in [46], several techniques

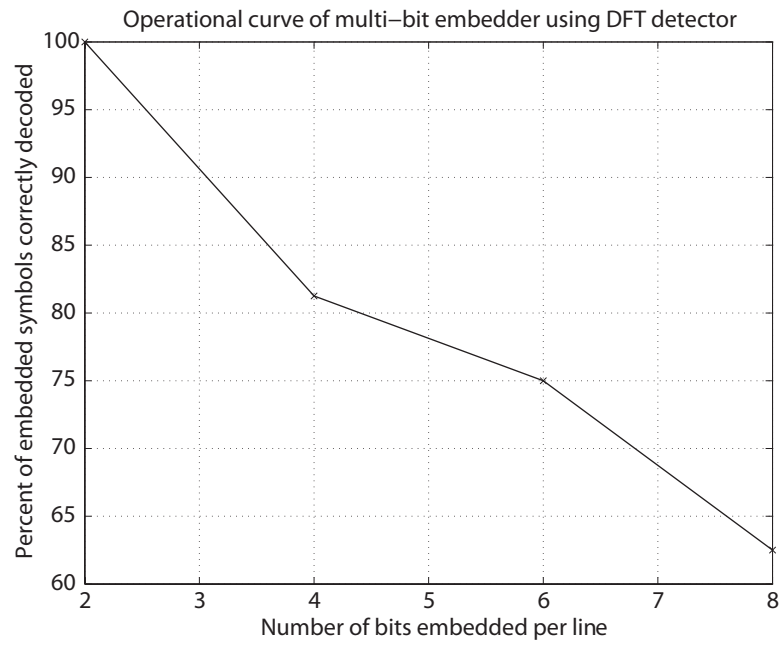


Figure 3.23. Symbol level operational curve using DFT decoder.

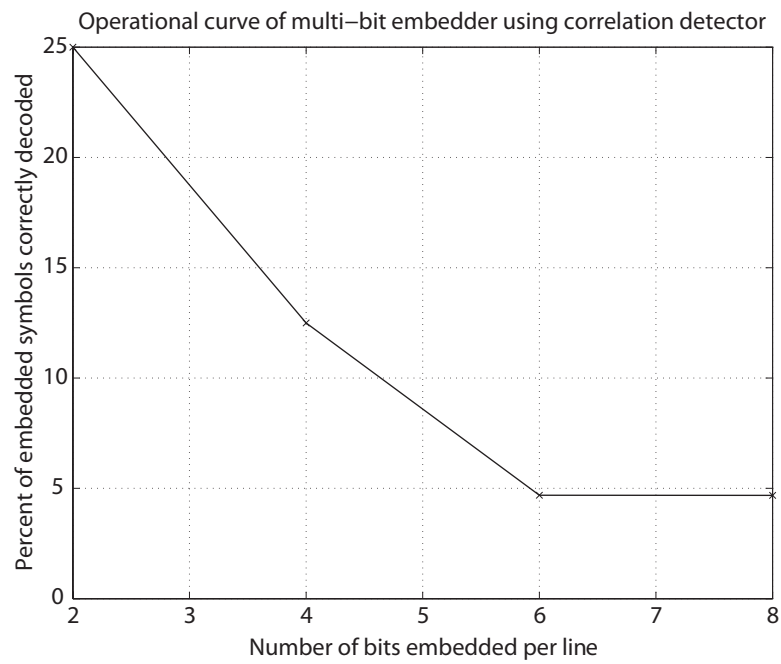


Figure 3.24. Symbol level operational curve using correlation decoder.

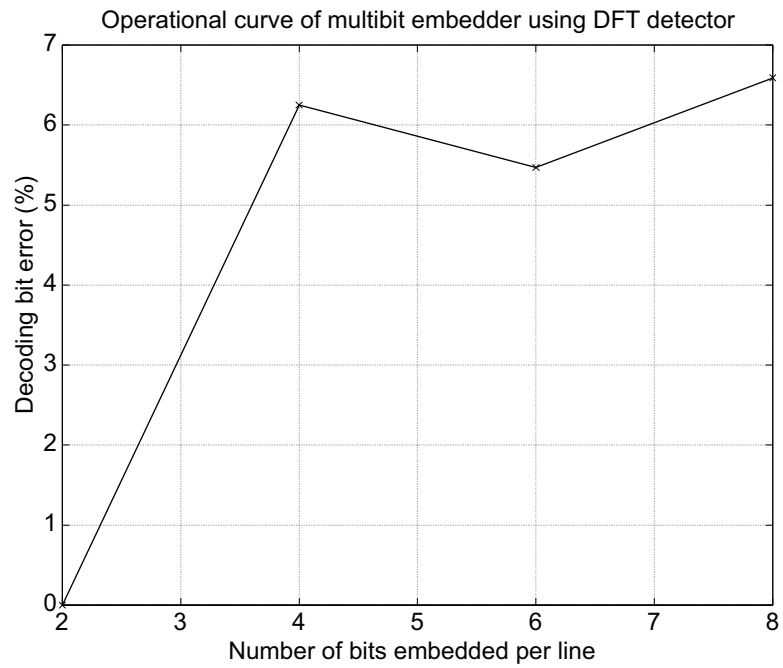


Figure 3.25. Bit level operational curve using DFT decoder.

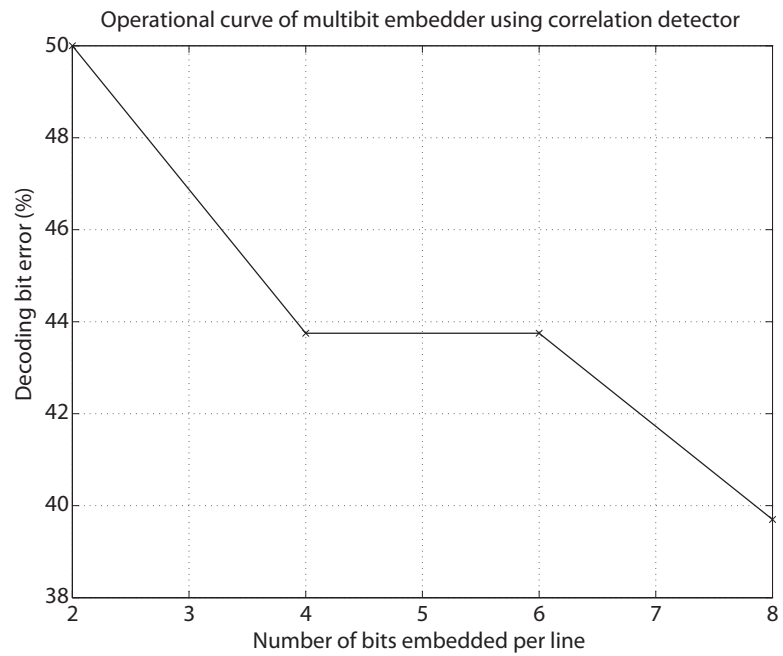


Figure 3.26. Bit level operational curve using correlation decoder.

can be used to inject an “artificial” banding signal into the document. In particular, we developed a system using laser intensity modulation which allows per-scan-line changes in laser intensity.

Figure 3.27 shows the effect of modulating laser intensity in different types of images. The first line is printed without any modulation. The second line is modulated with a high power 20 cycle/inch sinusoidal signal. The third line is modulated with a high power 40 cycle/inch sinusoidal signal. These signals can be easily seen in the halftone patches of Figure 3.27. This is because the frequency/amplitude combination of the signals are above the threshold developed in [78] below which human perceptibility is low.

However, this same signal is not perceptible in the saturated interior region of the text characters of Figure 3.27. This is not true for the edges of the text characters, specifically the left and right edges where the existence of the embedded signal is clearly seen in the enlarged character ‘I’ from the third line. This behavior allows the embedded signal to be estimated through extraction and analysis of the edges of individual text characters as described in [79].

A similar form of embedding can be performed by the driver before the document is sent to the printer, however in this case the resolution of the print process is the limiting factor. A straightforward approach would be to add or subtract pixels to the edges of characters. Our initial experiments suggest that the addition of one additional “layer” of pixels to the edge of a character using a 1200 DPI print process does not significantly impact the edge quality. A similar approach developed recently is described in [84].

3.4.1 Embedding and Detection

A subset of characters, $\Omega = \{ \text{b, B, D, E, F, h, H, k, K, L, m, M, n, N, p, P, r, R, u, U} \}$, containing a left vertical edge is chosen to embed signals into.

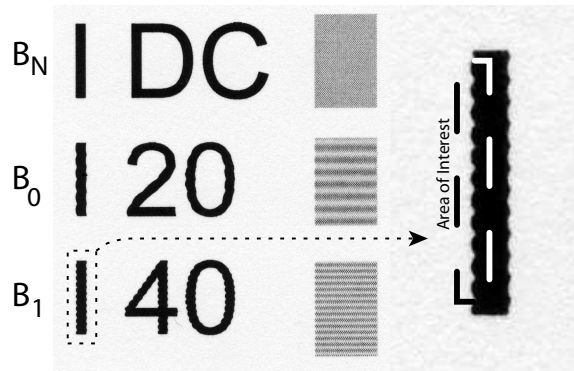


Figure 3.27. Large amplitude exposure modulation. First line has no modulation. Second line has 20 cycles/in modulation. Third line has 40 cycles/in modulation.

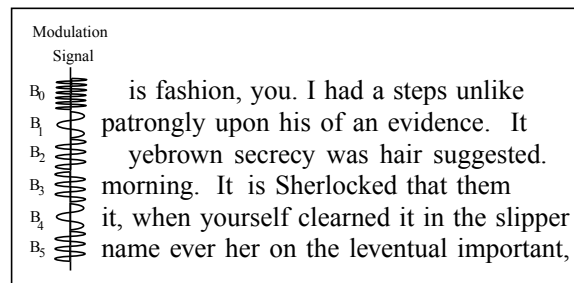


Figure 3.28. Modulation scheme for text documents.

The left vertical edge of these characters is modeled as a signaling period during which one channel symbol or group of bits is sent.

In [83] a signal set was considered where each symbol was a sum of sinusoids. Let $\mathbf{b} = \{b_0, b_1, \dots, b_n\}$ be a set of bits to be embedded into a line of text. The corresponding signal $B(y)$ can then be defined as

$$B[k] = \sum_{i=0}^n b_i A_i \sin\left(\frac{2\pi f_i k}{R_p}\right), \quad (3.27)$$

where

$$\mathbf{f} = f_0, f_1, \dots, f_n, \quad (3.28)$$

$$A_i = \frac{n-i}{n} A_{max} + \frac{i}{n} A_{min}, \quad (3.29)$$

where $f_i \in \{20, 30, 40, \dots, 100\}$, A_{max} is the amplitude to be used for frequency f_0 , and A_{min} is the amplitude to be used for frequency f_n . The amplitude varies linearly between frequencies f_0 and f_n . When viewed in the frequency domain, each frequency f_i corresponds to one bit in \mathbf{b} . For example, if $n = 8$, there would be 256 symbols, each corresponding to a different \mathbf{b} . Each signal $B[k]$ corresponds to one symbol which is embedded into a text line as shown in Figure 3.28.

To decode the embedded symbols, the document was scanned at a resolution $R_s = 2400$. Individual lines of text were then extracted and processed individually. All characters in the line of text were segmented from the scanned image. The edge profile $\hat{B}[k]$ was estimated for each extracted edge from the line. The power spectral density (PSD) of each profile was then obtained using a 240 point DFT as shown in Equation 3.30. 240 points are used to create 10 cycle/inch wide bins centered at the frequencies of interest.

$$S_{\hat{B}}[k] = \left(\sum_{n=0}^{239} \hat{B}[n] e^{j \frac{2\pi * k * n}{240}} \right)^2. \quad (3.30)$$

The original embedding frequencies were then chosen as those with PSD values greater than some threshold determined by empirical measurements or observations.

To embed information similarly into the text before it is sent to the printer, it is necessary to change the signal set since we can no longer make sub-pixel changes. Instead, we are limited by the printer resolution R_p , which in the case of many modern EP printers is 1200DPI. If we try to use the signal set in Equation 3.27 by simply thresholding the output to integer values (corresponding to adding or subtracting 1 or more pixels from the edge), this is equivalent to convolving the frequency response of the signal with a *sinc*, generating harmonics of each f_i which may make detection through use of the DFT difficult due to harmonics from the lower frequency signals affecting the higher frequencies.

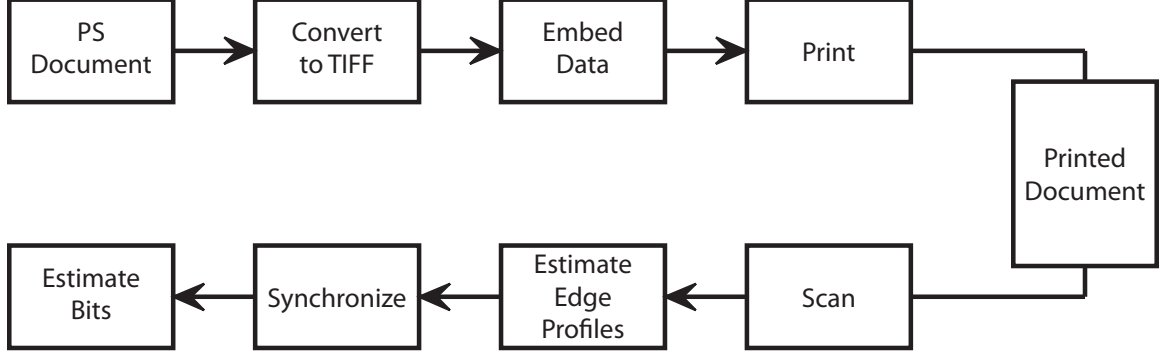


Figure 3.29. Block diagram of data hiding system.

Instead we choose to view the edges as a binary baseband channel. A simple antipodal signal set is selected with which to embed individual bits, specifically

$$s_p[k] = \begin{cases} -0.5 & : 0 \leq k < \frac{T_p}{2} \\ 0.5 & : \frac{T_p}{2} \leq k < T_p \\ 0 & : else \end{cases}, \quad (3.31)$$

where

$$s_{p,1}[k] = -s_{p,-1}[k] = s_p[k], \quad (3.32)$$

the signal length T_p is chosen to be 6 printer pixels and the subscript ‘ p ’ indicates that the sample rate of this signal is R_p . The non-modified edge is viewed to be at position 0.5 pixels. In this manner, $s_{p,i}[k] = 0.5$ corresponds to the addition of 0 pixels to the edge at position k , and $s_{p,i}[k] = -0.5$ corresponds to the addition of 1 pixel to the edge at position k . The received signal with mean subtracted will then lie between -0.5 and 0.5 . The choice of $T_p = 6$ was based on experiments in [82] so that the signals are 3-on 3-off to overcome issues related to printer dot overlap and modulation transfer function. This choice of T_p places the fundamental frequency of the signals at 100 cycles/inch which corresponds to alternating 1 and 0 bits.

Given that $R_p = 1200$ for the printer we use in our experiments, we find that the shortest edge length within the set Ω is approximately 100 printer pixels.

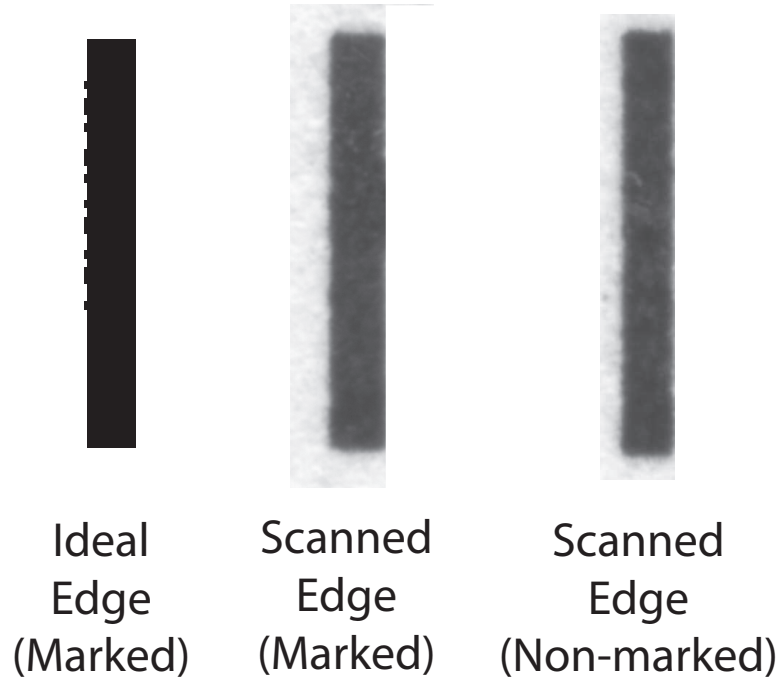


Figure 3.30. Example of embedded signal.

A block diagram of the system used to embed and detect these symbols is shown in Figure 3.29. We start with a document in PostScript (PS) format. This can be generated directly using \LaTeX or by word processors such as OpenOffice by printing directly to a file using a generic PS print driver. The PS document is converted to a TIFF bitmap image at R_p DPI using `ghostscript`. The optical character recognition (OCR) tool called `OCRAD` [85] is used to find the locations of characters in Ω . The embedder then embeds one symbol into the left vertical edge of each valid character, generating a new TIFF image containing the marked text. The new TIFF image is then printed at the native printer resolution to create the marked document.

An example of how this signal is embedded into a vertical edge is shown in Figure 3.30. The left image shows an edge as it appears in the marked TIFF image. The center image shows the same edge scanned from the marked document. For comparison, the right image shows an unmarked edge. It is difficult to see by inspection which edge is marked.

To detect the embedded bits, we first scan the document at a scan resolution of $R_s = 4800$ DPI. OCRAD is again used to identify the locations of characters in Ω . The edge profiles of valid characters are estimated using the R60 transition method [80]. The estimated edge profile, with sample rate R_s , can be written as

$$\hat{s}[k] = s_{p,i} \left\lfloor k \frac{R_p}{R_s} \right\rfloor + n_k = s_{s,i}[k] + n_k, \quad (3.33)$$

where the scaling of k by $\frac{R_p}{R_s}$ is to account for the change in sample rate from R_p to R_s , and n_k is the sample noise modeled as a white Gaussian noise process with zero mean and variance σ^2 .

The optimal detection scheme involves the use of a matched filter. Since antipodal signals are being used, the filter is defined as

$$h[k] = \begin{cases} s_s[T_s - k] & , \quad 0 \leq k < T_s \\ 0 & , \quad otherwise \end{cases}, \quad (3.34)$$

$$T_s = \left\lfloor T_p \frac{R_p}{R_s} \right\rfloor, \quad (3.35)$$

with the output defined as

$$r[k] = \hat{s}[k] \otimes h[k]. \quad (3.36)$$

This can be expanded as follows:

$$\begin{aligned} r[k] &= s_{s,i}[k] \otimes h[k] + n_k \otimes h[k] \\ &= s_{s,i}[k] \otimes s_s[T_s - k] + n_k \otimes h[k] \\ &= \sum_{j=0}^k s_{s,i}[j] s_s[j] + \sum_{j=0}^k n_k s_s[j] \\ &= r_s[k] + r_n[k] \end{aligned} \quad (3.37)$$

where r_s and r_n refer to the signal and noise portions of the filtered signal. Evaluating the above equation at a sampling time T_s and noting the fact that s_s is defined only on the interval $[0, T_s)$, the mean and variance of $r[k]$ can be obtained as

$$E \{r[t_s]\} = i ||s_s||^2, \quad (3.38)$$

and

$$Var \{r[t_s]\} = \sigma^2 ||s_s||^2. \quad (3.39)$$

What this shows is that the decision variable, $r[Ts]$, is Gaussian distributed at $\pm ||s_s||^2$ depending on whether the signal that was embedded was a 1 or -1 . Since the means are symmetric around zero, the optimal decision threshold can be chosen as $\gamma = 0$, or more formally

$$\gamma = \frac{(1)||s_s||^2 + (-1)||s_s||^2}{2} = 0. \quad (3.40)$$

If $r[Ts] > \gamma$, then $s_{p,1}$ was embedded, otherwise $s_{p,-1}$ was embedded.

The expected probability of bit error is then given by

$$P_e = \text{erfc} \left(\sqrt{\frac{||s_s||^2}{||s_s||^2 \sigma^2}} \right), \quad (3.41)$$

where the function erfc is the complementary Gaussian error function defined as

$$\text{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2} dt. \quad (3.42)$$

From the signal definition in Equation 3.31,

$$||s_s|| = T_p \frac{R_s}{R_o}, \quad (3.43)$$

which for $T_p = 6$, $R_p = 600$, and $R_s = 4800$, gives $||s_s|| = 48$. The noise variance was empirically obtained by printing a page of text with no embedded data, scanning it at 4800 DPI, and averaging the edge variance across all the edges in the page. This provided a noise estimate of $\sigma^2 = 1.3056$. The expected error probability from these parameters is

$$P_e = \text{erfc} \left(\sqrt{\frac{48^2}{48^2 \cdot 1.3056}} \right) = 0.2158. \quad (3.44)$$

It should be noted that this assumes perfect synchronization between the detector and the signal \hat{s} .

3.4.2 Results

A test page consisting of a half page of 12 point text with 454 embeddable characters was generated and was embedded with 6356 randomly generated 14-bit symbols. The parameters used were $A = 1$ and $T = 6$. The marked TIFF image was printed

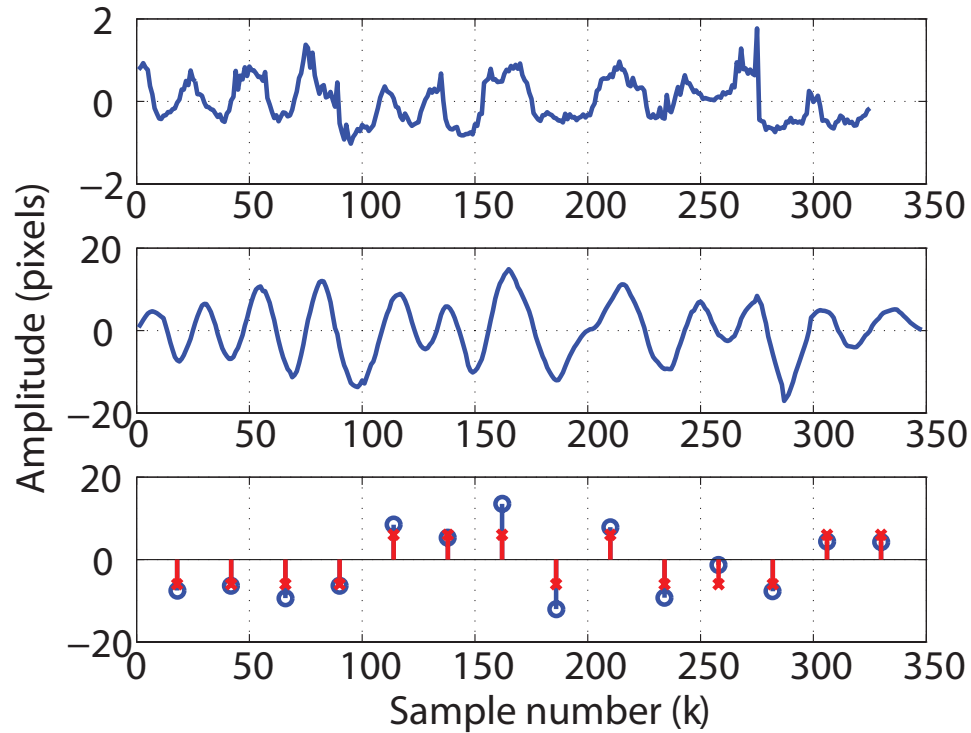


Figure 3.31. The top plot shows an instance of \hat{s} , a high pass filtered edge profile. The center plot shows the same signal after matched filtering, $d[k]$. The bottom plot illustrates sampling instances $d[4iT]$ (marked with blue 'O's') at which decisions are made as to which bit is present in each period T . The red 'X's mark the ground truth for each detected bit. In this case all bits were detected correctly.

on an HP Color LaserJet 3800dn, a 1200 DPI printer. The printed page was then scanned using an Epson Perfection 4490 flatbed scanner at 4800 DPI in 8 bit grayscale mode. With the parameters chosen in the previous section, we were able to embed approximately 13000 bits in a page of 12 point text with an experimental bit error rate of 31.34%. 104 symbols had greater than 7 bits in error, which may suggest that the detector experienced synchronization errors.

Figure 3.31 shows an example of the signals \hat{s} , $d[k]$, and $d[4iT]$. In the case of the signal shown here, all the bits were correctly detected. However from our results this clearly was not the case for all signals.

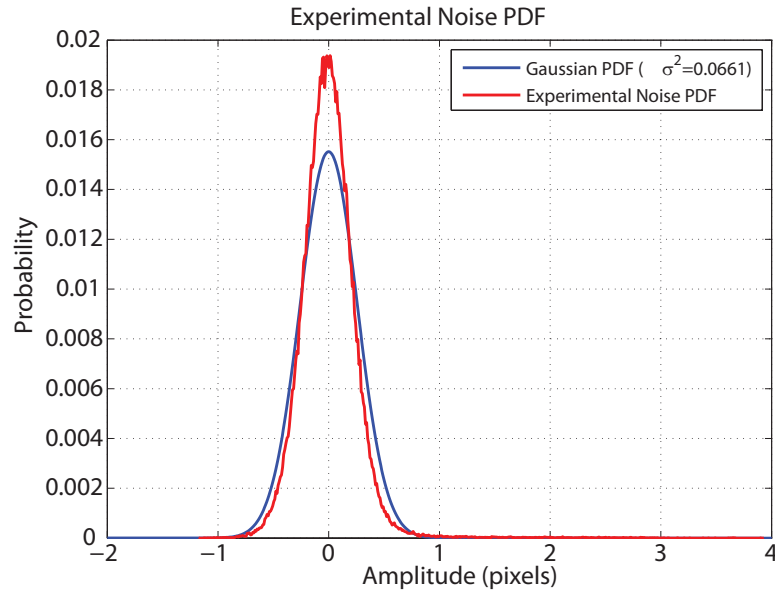


Figure 3.32. Experimental pdf of $n_h[k]$ overlaid on top of Gaussian pdf with estimated $\sigma_{n_h}^2 = 0.0661$.

Because detection using a matched filter is sensitive to phase, any offset in sampling times can cause an incorrect decision to be made, even if the phase is correct for detection of the first bit. There are several assumptions made earlier which may be contributing to synchronization error. The first is that we assumed the scan resolution to be exactly 4 times that of the print resolution. Even if this were true in a nominal case, the banding phenomenon in the printer would cause the spacing between scan lines to fluctuate.

In relation to the predicted error probability, the noise was assumed to be Gaussian in nature, which Figure 3.32 shows is nearly the case. In addition, the noise after high pass filtering was assumed to be wide-sense stationary which led to the simplified expression of the noise variance after matched filtering. However the autocorrelation of $n_h[k]$ in Figure 3.33 shows that this is not the case.

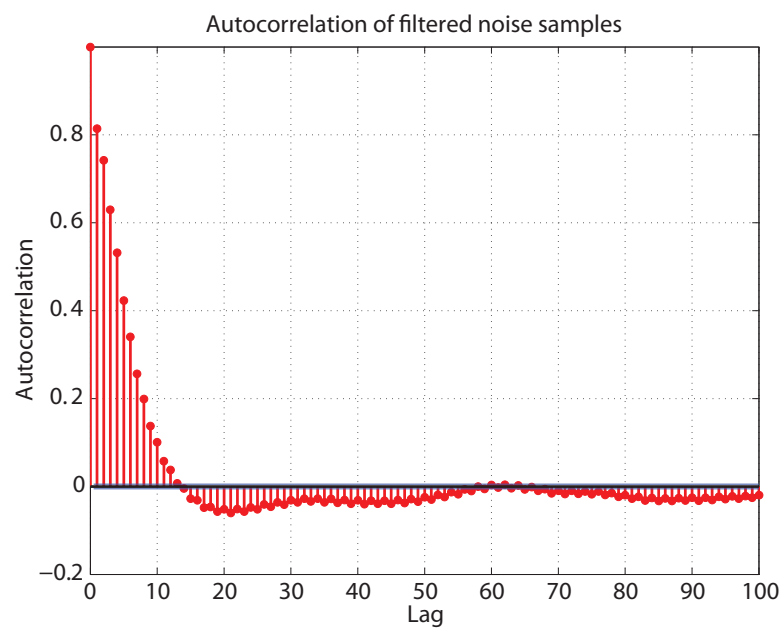


Figure 3.33. Autocorrelation of $n_h[k]$.

3.5 Counterfeit and Tamper Detection

Consider the problem of telling apart an original printed document from a version that is a product of one-or-more scan-and-print cycles where the document is not altered, and detecting and locating malicious change within printed documents.

We define a counterfeit, or non-authentic document, to be the product of a 2-step process (i) Acquiring a scanned digital version of the original document, and (ii) Reprinting the scanned original document either directly or after applying some transformation to obtain a second generation document.

We would ideally want the document to carry all necessary evidence of originality to both recognize a counterfeit (as described above) and locate malicious changes. The nontrivial technical challenge to overcome then is how to add security features to a text document without changing its logical text content such that the embedded information can be used to address the the two cases.

The following sections describe an approach to addressing these problems that was presented in [86]. We make use of the embedding techniques described in the previous section to overcome these challenges. In our implementation, for the first case, we maintain an electronic copy of the original document with the embedded data and use a simple threshold-scheme for detecting counterfeits. For the second case, our implementation can locate malicious changes in standardized contracts using only information present in the document. Using parameters that allow us to adjust the strength of the embedded signal, we have what is best called “calibrated fragility” that allows the embedded signal to withstand the noise from one printing operation but not from an additional scan-and-print cycle and also allows us to detect and locate malicious modifications to the document.

The various stages in the data embedding process are detailed in Figure 3.34. At a high level what we do is embed bits in a subset of characters in an electronic version of the original document. The bits encode a test of authenticity. Based on the

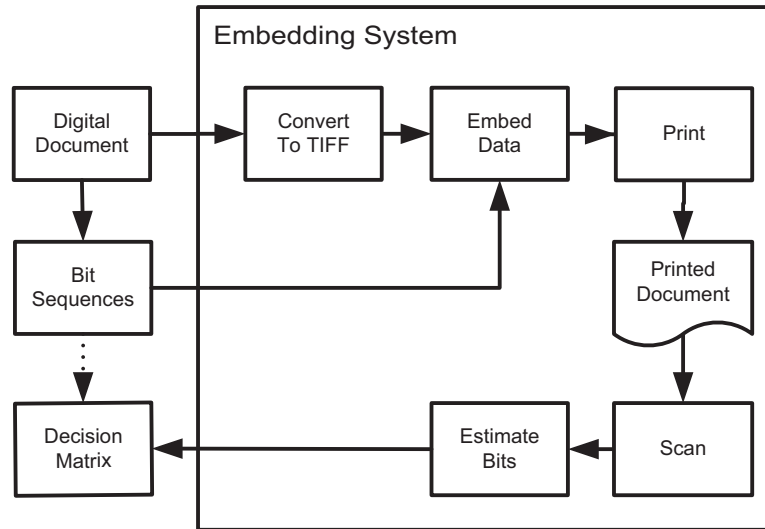


Figure 3.34. Block diagram of our modified embedding system.

method used to create these bit-sequences, we can detect products of scan-and-print cycles or locate malicious changes.

The data hiding system we use, described earlier, embeds signals which represent bits on the left edges of characters in the set

$$\Omega = \{b, B, D, E, F, h, H, k, K, L, m, M, n, N, p, P, r, R, u, U\}.$$

The characters in Ω contain a left vertical edge in which extrinsic data can be embedded. In this system the noise inherent in the printing process makes the signals difficult to duplicate without knowledge of the original data. Also the signals are not easily detectable by a human observer. The choices for the parameters of the signals are described in the following sections.

One of the tools that is used to achieve the goals of this section is a hash-based message authentication code (HMAC) [87]. A hash function $h = H(M)$ is a “one-way” function in the sense that it is trivial to compute the hash h given M , but computationally difficult or infeasible to find M given h [88]. Typically M is an arbitrary length message or bit sequence. The resulting hash h is a fixed length value. Two hash functions in common use today are SHA-1 and MD5, both of which are described in [87]. Additionally, it is difficult to find another message M' such

that $H(M') = H(M)$. Such functions are of critical importance in cryptography, especially as building blocks of authentication methods.

Suppose, for example, we wish to protect the content of a document. One way to do this would be to treat the contents of the document as the message M and generate the hash $H(M)$ of the document. Anyone who wanted to verify that the contents of the document, M' , could then simply generate the hash $H(M')$ and compare it to the original. If $H(M') = H(M)$, then with high probability $M' = M$ and the document's contents are authentic and unchanged.

For several reasons, applying a hash function in this manner to protect a message M is not secure. Although difficult, an attacker who has possession of M could find a message $M' \neq M$ such that $H(M') = H(M)$. An attacker could then pass M' as the original message. This is known as a collision attack.

One way to address this is to use a type of keyed hash function called a message authentication code (MAC). A MAC is a hash function that also takes a key k as an input. The key k is typically a fixed length pseudo-random bit sequence. The hash of the original message M in this case would be $h = MAC(M, k)$. Anyone with the key k can verify the hash of the message, and the hash and key are only shared with trusted parties. Let us assume an attacker in possession of M wants to generate a message $M' \neq M$ and pass it off as the original. Since the attacker does not know k , he can only assume some k' . If we assume k has a length of 128 bits, then there are $2^{128} \approx 3.4 \times 10^{38}$ possible keys for the attacker to choose from. Even if the attacker has the original message, with high probability he cannot determine the key that was used and therefore he cannot determine the original message hash to which he would match $MAC(M', k')$.

An HMAC is similar to a MAC except that it is also a function of the MAC of the message. The HMAC of a message M would be $h = HMAC(M, k, MAC(M, k))$. This “nested” MAC further addresses issues related to the collision attack described earlier. The specific HMAC implementation used is from the Python Cryptography Toolkit described in [89].

3.5.1 Counterfeit Detection

We can implement counterfeit-detection using the data-hiding system described above. The embedding system provides for every character c in the document: the column location C , the row location R , the character width w , the character height h . For each document we wish to protect, we use a unique identifier, i . Using key K , we compute a bit sequence of desired length using $HMAC(K, C, R, w, h, c, i)$ as the basis for the sequence.

For making a counterfeit, the scanning process produces a digital image. In this resulting image, the character positions will almost always differ from the original pixel positions. Hence, by using C and R in the HMAC, we make it very hard to reconstruct the original checksum. The document-specific identifier, i , ensures that the HMAC differs for characters whose instances in two different documents share values for C , R , w and h .

To detect if a document is genuine, we scan the document and recover the embedded bits using the procedure described in [90]. The recovered bits are then compared to the ground truth, the originally embedded bits, to obtain a percentage error of the number of incorrect bits. If the error is greater than some threshold then the document is considered fake (product of at least one scan-and-print cycle). Note this requires that a copy of the original data be stored.

3.5.2 Detecting Content-Tampering

We now approach the problem of locating malicious changes. The technique is described as it would apply to standard contracts. These documents follow a template and the “critical content” (data-items that are important) are filled in using this template (ex: transcripts where the legend for an institution’s transcript is consistent across all transcripts).

In general, we construct bit-sequences using HMACs that serve as integrity checks for one or more data item(s) and employ the same threshold scheme as above to detect the integrity of these items.

Detection of the item subjected to the tampering depends upon how the bit-sequences were embedded. In the following examples, data items that are to be protected are denoted d_1 through d_n . The checksums to be embedded to protect each of the n data items are denoted C_i . We modify the basis of the checksum to $HMAC(K, d_1, d_2, \dots, d_i)$ as opposed to the HMAC from the previous section, where d_k is the k th data-item to be protected. As a result, for a standard-contract, the document is more-or-less self-validating. We demonstrate two techniques below. The first uses n checksums to identify which of n items has been changed. We then explore a subset of this problem where we can locate tampering of only 1 out of n items. This relaxed requirement allows just $\log n + 1$ checksums to be used.

The checksums for each item need to be embedded in a reasonably sized region of text in the original document. By reasonably sized, we mean that the region of text must have a good density of characters from Ω . The region of text embedded with checksum C_i is denoted R_i .

Linear Number of Checksums

In this situation, there are n items we want to protect in the document and we construct n checksums. An example of such a situation would be a transcript where we are trying to protect $(n - 1)$ grades and the name of the student. During the embedding phase, each checksum is embedded in a predetermined region or subregion of the document. During the extraction phase, we first compute the checksum C_i for item d_i observed in the document. Then, we retrieve bits from the appropriate region and compare them with C_i . If the number of bits matching the recomputed checksum is less than some empirically derived threshold then we denote that data item as suspicious.

Logarithmic Number of Checksums

In this situation we aim to protect n data items in the document but are interested in locating which item when only one has been tampered. An example would be a transcript where we are interested in protecting $(n - 1)$ grades and the name of the student but want to detect a modification to any one of these items. We only need $\log n + 1$ checksums in this situation [91, 92].

- The first checksum embedded in region R_0 is a checksum used to verify all the data items.
- The next $\log n$ checksums, for $j = 1, \dots, \log n$, are for the concatenation of those d_i for which the integer i has a 1 in the j th least significant bit of its binary representation.
- To determine which d_i is corrupted, the binary representation of integer i is constructed one bit at a time, as follows: For $j = 0, \dots, \log n - 1$ in turn, if the number of bits in the j th computed checksum matching the stored checksum in region R_j is less than an empirically derived threshold then the j th bit of i is 1 and 0 otherwise. This i represents the data item d_i that has been tampered.

3.5.3 Data-Hiding Technique Parameters

The data hiding system in [90] works as follows. Let $s[k]$ be the horizontal offset of the pixels on the left edge of a symbol, s , from their ideal position. We assume that the non-modified edge is $s[k] = 0$. Signals are embedded by shifting edge pixels right or left by an integer number of pixels to create a detectable pattern or signal. In [90] a Manchester signaling scheme was used with an adjustable amplitude A and period T to control the strength and detectability of the embedded signal. The larger A is, the more easily the signal can survive the printing process.

Once the edges in a document are modified to contain the desired data, it is printed to create what we call the “genuine” or “authentic” document G_1 .

The embedded data is then recovered by first scanning the genuine document G_1 at high resolution to obtain a digital representation \hat{G}_1 . The embedded bits are then recovered from the character edges using a matched filter [90].

3.5.4 Constructing a Threshold for Counterfeit Detection

The threshold used for determining document genuinity is obtained empirically as follows. A non-genuine document is created by first scanning it and subsequently reprinting it without any document content being changed. The additional noise from the scan and reprint process implies that the bit error in G_1 will be lower than in any subsequent G_n for $n > 1$. We generate ten genuine documents G_1^i with differing content, and for each one we create two copies G_{2a}^i and G_{2b}^i . The first copy is simply a scanned and printed version of the original. The second copy includes a 50% thresholding step between the scan and print steps which is sometimes found on copymachines to increase text sharpness and contrast.

The embedded data is recovered from each of the scanned $\hat{G}_1^{(i)}$, $i = 1, \dots, 10$. Let T be the average percentage of correctly recovered bits across all 10 documents. Let s be the standard deviation of the percentage of correctly recovered bits across all 10 documents. If fewer than $(T - 2s)\%$ bits are retrieved correctly then the document is identified as counterfeit, otherwise the document is identified as genuine.

3.5.5 Results

The threshold value for genuinity determination was obtained using 10 documents with varying densities of characters from Ω . The parameters used for embedding the bit-sequences on the characters in each of these documents were $A = 3$, $r = 3$ synchronization bits, and $n = 4$ information bits per character (see [90] for details).

We observed that for genuine documents, on average, 92.26% of bits are retrieved. This is the T value we used for our printing process. The standard deviation in the percentage of bits retrieved was 3.21. This is the s value we used. For making a

Table 3.3. Preliminary results for determining document genuinity.
 $(T = 92.38, s = 3.08)$

Doc ID	$T_{\hat{G}_1}$	$T_{\hat{G}_{2a}}$	$T_{\hat{G}_{2b}}$	#chars $\in \Omega$	#chars
Doc01	93.57	36.05	46.88	168	661
Doc02	91.19	37.95	48.20	267	978
Doc03	95.47	53.54	50.86	257	901
Doc04	93.45	53.43	49.06	383	1361
Doc05	94.05	36.90	50.79	254	920
Doc06	86.03	24.26	29.78	119	449
Doc07	90.05	35.75	48.92	101	421
Doc08	92.93	52.88	53.80	73	362
Doc09	88.29	45.24	57.14	237	907
Doc10	95.35	45.35	54.94	184	640

counterfeit, two approaches were used: (i) scan the genuine document and reprint, and (ii) scan the genuine document, apply 50% thresholding and then print. From the counterfeits we constructed, we could retrieve only 42.14% percent of bits correctly for first case (without thresholding) and 49.04% bits for the second case (with thresholding). From the T and s values above, all the counterfeits were successfully identified as fake since the percentage of bits retrieved fell well below the $(T - 2s)\%$ threshold set by us. For protecting a document, we used a transcript consisting of grades from 7 courses. We aimed to protect the student name and all 7 grades resulting in 8 items overall. The transcript we used provided the grades on one side and the transcript legend on the other side. This text is consistent across all the transcripts issued by the institution. We used one paragraph as one region for embedding one checksum. So, for the N-Checksums technique, we used 8 paragraphs from this page. For the $\log n + 1$ checksums technique we used 4 paragraphs. We were successfully able to locate tampering of data using both techniques.

3.6 Printer Model

At the beginning of this chapter, a high level description of the EP process with laser power modulation was introduced. A model based on this description was developed in [93,94] to characterize dot placement and formation. A brief overview of the model from [94] is presented below.

The printed image is simulated by a two step process. First a high resolution probability map of toner distribution is generated corresponding to the original image. Toner particles are then placed according to the probability map on the high resolution grid to form a binary image of toner particles. The high resolution toner image is then filtered and downsampled to represent the scanned output of the printed document.

Let $s(x, y) \in [0, 1]$ be a $W \times H$ bitmap image of a document to be printed, rasterized at the printer's native resolution R_p with x denoting the column and y

denoting the row of each pixel in the image. Let U and V be the scaling factors used to generate the high resolution toner grid, that is each pixel in the original image is represented by $U \times V$ lattice cells for placement of toner particles. Let $\tilde{s}(x, y)$ be the binary image of size $WU \times HV$ corresponding to the toner grid at an effective resolution $R_t = R_p(U, V)$ DPI. Define the pdf of toner distribution

$$p = \Lambda e^{-\mu d^2}, \quad (3.45)$$

$$d = \sqrt{\left(\frac{u}{U} - (x - \Delta x)\right)^2 + \left(\frac{v}{V} - y\right)^2}, \quad (3.46)$$

where (u, v) are coordinates of pixels within the toner lattice and (x, y) are the coordinates of a black pixel in the $s(x, y)$. The quantity

$$\Delta x = k_1 e^{-k_2 V_r} + k_3, \quad (3.47)$$

represents the dot shift that is caused by the implementation of the embedding system and is a function of the laser control voltage $V_r \in [1, 2]$. In general, Λ and μ are also functions of V_r .

The toner probability map for a given image is generated as follows. For each (x, y) such that $s(x, y) = 1$, find the corresponding pixels (u, v) in the toner lattice that are within the neighborhood $d < D$ and add to them the corresponding probability p . After repeating this for every black pixel in s , the resulting \tilde{s} will contain a probability map of toner particle distribution. Using a uniform pseudo-random number generator each pixel in the toner lattice is then set to either zero or one according to the corresponding probability value. \tilde{s} is then filtered by the scanner psf h_s and downsampled to the scanner resolution R_s .

The original work in developing the model was focused solely on cluster-dot halftone printing. The parameters of the model were chosen to minimize the error between the simulated and actual halftone pattern generated by the printer. There are several effects that were noted in the previous chapters that the original model does not account for. Specifically the original model does not accurately represent the variance/raggedness of vertical edges or the difference in amplitude between the modulation present in the left and right edges of text.

The original values in [94] were obtained for a 600 DPI printing process by minimizing the error between simulated and experimental dot size, dot shift, and the resulting average graylevel of cluster-dot halftone patches with $U = V = 50$. The new values were similarly obtained by minimizing the error between the simulated and experimental edge profiles and edge sharpness for both the left and right edges where the edge sharpness metric used is the transition width from the R20 edge profile to the R80 edge profile. To obtain the new values, the parameters U and V were chosen to be $U = V = 10$, which for a 600 DPI print process gives a toner lattice cell size of approximately $4\mu\text{m}$ which more closely represents true toner particle size of around $10\mu\text{m}$ [95].

A page containing 360 vertical bars with dimensions 140×10 printer pixels was printed on a 600 DPI HP Color LaserJet 4500 in a 20 row by 18 column layout. Each row r was embedded with a square wave defined by the signal

$$V_r(i, r) = \begin{cases} 1.0 & 1 \leq i \leq 15 \\ 1.0 + r/20 & 16 \leq i \leq 30 \end{cases}, \quad (3.48)$$

where i is the vertical position in the vertical bar in units of printer pixels. As shown in the above equation, each row is embedded with a higher amplitude signal with row one having the lowest amplitude, and row 20 having the highest amplitude. After the test page was printed it is scanned in 8-bit grayscale using an Epson Perfection 4490 flatbed scanner at 4800 DPI. The left edges of each vertical bar are then estimated using the same methods as in Section 3.2, specifically the R60 edge profile. In addition, the edge transition width, defined by the difference between the R20 and R80 edge profiles is obtained. The Nelder-Mead simplex method [96] is used to obtain the best parameters Λ , μ , and Δx for each row of vertical bars independently by minimizing the squared difference between the simulated and measured edge profiles and edge width. To simplify the process, $\Delta\tilde{x}$ was defined to be 0 for $V_r = 2$. In the original work, $\Delta x = 0$ for $V_r = 1.5$.

Since the toner dots comprising a character edge overlap considerably, the parameter Δx estimated from the edge no longer represents the true dot shift, but instead

represents a composite value of dot shift plus edge shift due to overlap of the toner dots. The notation $\Delta\tilde{x}$ is used below to emphasize this difference with $\Delta\tilde{x}$ defined to be 0 for $V_r = 2$. This definition makes $\Delta\tilde{x}$ a relative measure of edge shift.

A table of estimated parameter values for multiple values of V_r are shown in Table 3.4. The relationship between each of the model parameters and V_r is modeled by the following power function

$$aV_r^b + c. \quad (3.49)$$

The estimated coefficients for each of the parameters are shown in Table 3.5. Plots of the estimated parameters and parametric fits are shown in Figures 3.35-3.37. Also shown in the plots are the original parameters as obtained in [94] and defined by

$$\mu = 0.3, \quad (3.50)$$

$$\Lambda = 0.7155V_r - 0.3741, \quad (3.51)$$

and

$$\Delta x = 27.42e^{-2.34*V_r} - 0.85. \quad (3.52)$$

The values for $\Delta\tilde{x}$ in Figure 3.37 have been shifted such that $\Delta\tilde{x} = 0$ at $V_r = 1.5$ for easier comparison with the original values since only the relative change between Δx values affects the simulated image with respect to edge profile and edge width.

Note that the parameter with the largest difference is Λ , followed by μ , and finally $\Delta\lambda x$ which is nearly unchanged. The decrease in Λ most likely accounts for the simplistic toner overlap model that was used. The old values for Λ was okay for halftone images where there was little to no toner dot interaction, however too much toner is placed by this model on an edge where the toner dots are right next to each other. The decrease in *Lambda* accounts for this model mismatch. The change in μ , the variance of the toner dot distribution, similarly decreases the effective amount of toner placed by the model on saturated edges.

Table 3.4. Parameters for dot model estimated using embedded edges.

V_r	μ	Λ	$\Delta\tilde{x}$
1.05	0.2207	0.1469	2.2476
1.10	0.2192	0.1587	1.9406
1.15	0.2175	0.1728	1.6388
1.20	0.2215	0.1761	1.5348
1.25	0.2166	0.1828	1.3478
1.30	0.2169	0.1808	1.1928
1.35	0.2194	0.1890	1.0917
1.40	0.2270	0.2151	0.8857
1.45	0.2283	0.2222	0.7670
1.50	0.2265	0.2304	0.6844
1.55	0.2246	0.2436	0.6601
1.60	0.2431	0.2776	0.4947
1.65	0.2503	0.3026	0.4416
1.70	0.2530	0.3284	0.3848
1.75	0.2867	0.4277	0.2797
1.80	0.2866	0.4279	0.2799
2.00	0.2840	0.5360	0.0000

Table 3.5. Model coefficients for each parameter.

Parameter	a	b	c
μ	0.1391	-2.5030	0.1853
Λ	0.5993	-2.5980	0.03777
$\Delta\tilde{x}$	0.1409	4.1410	-0.06652

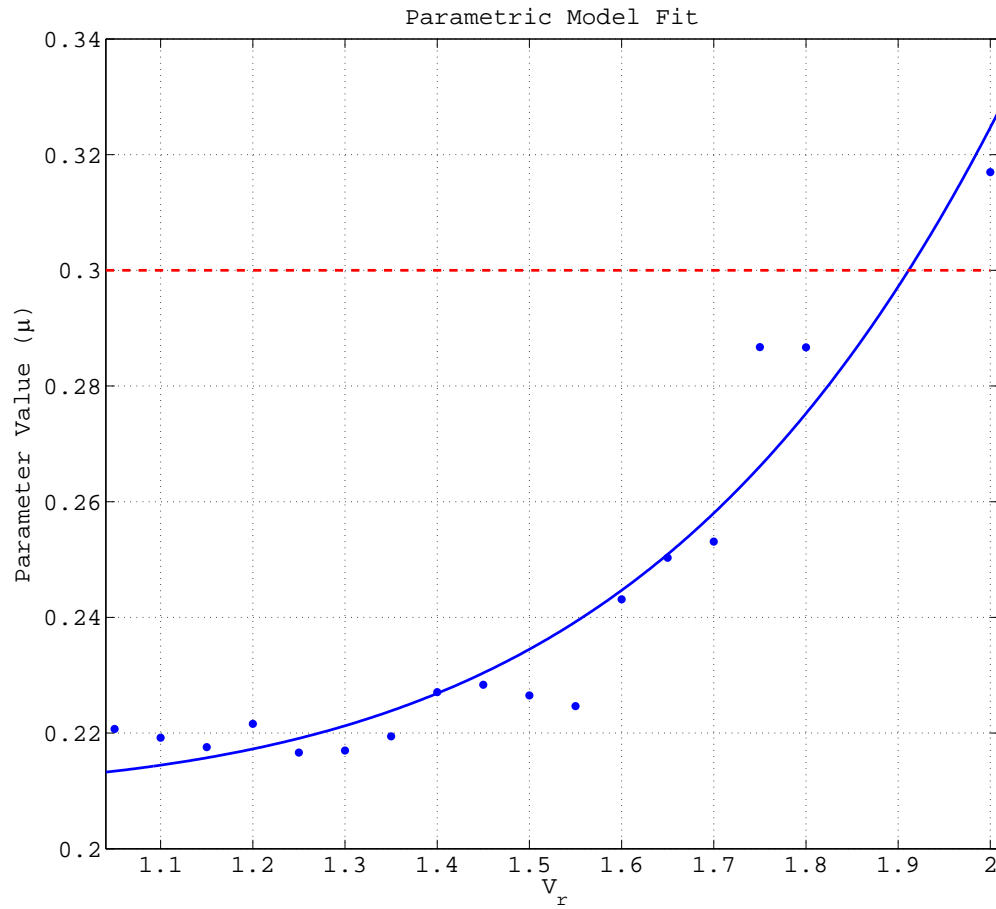


Figure 3.35. Estimated values of μ and parametric fit to the data. Dashed line represents original parameter values as defined in [94].

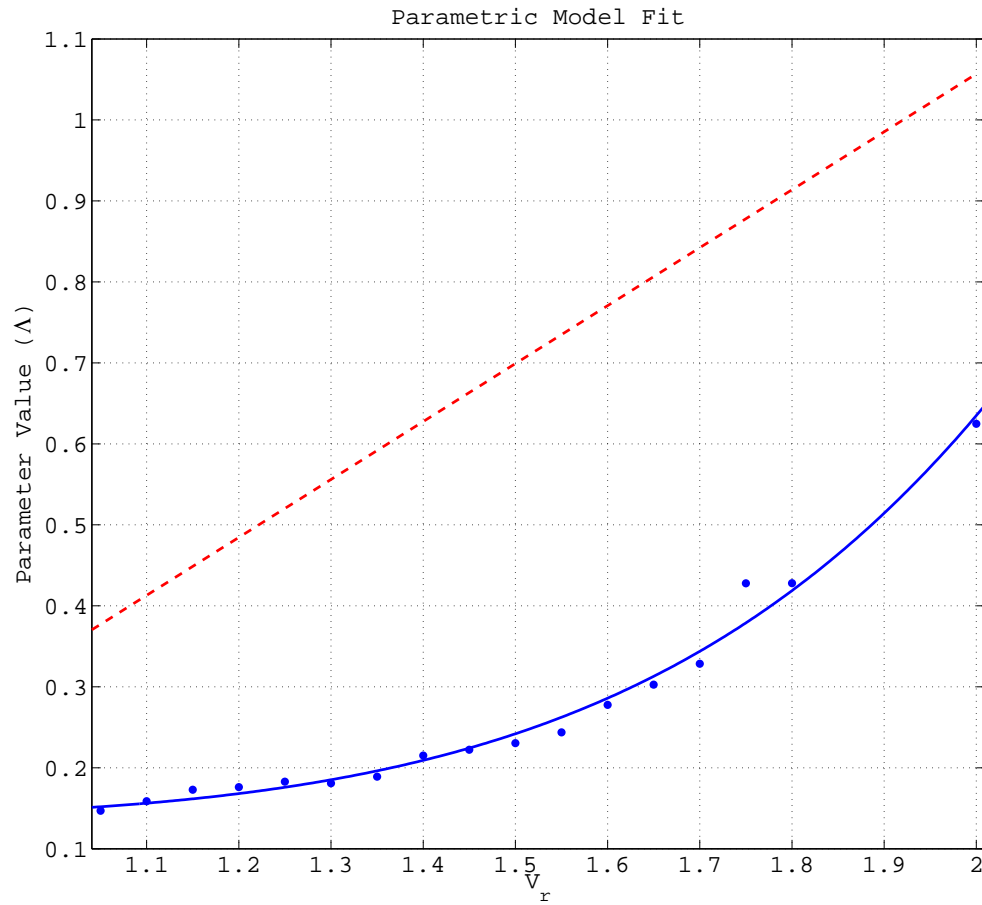


Figure 3.36. Estimated values of Λ and parametric fit to the data. Dashed line represents original parameter values as defined in [94].

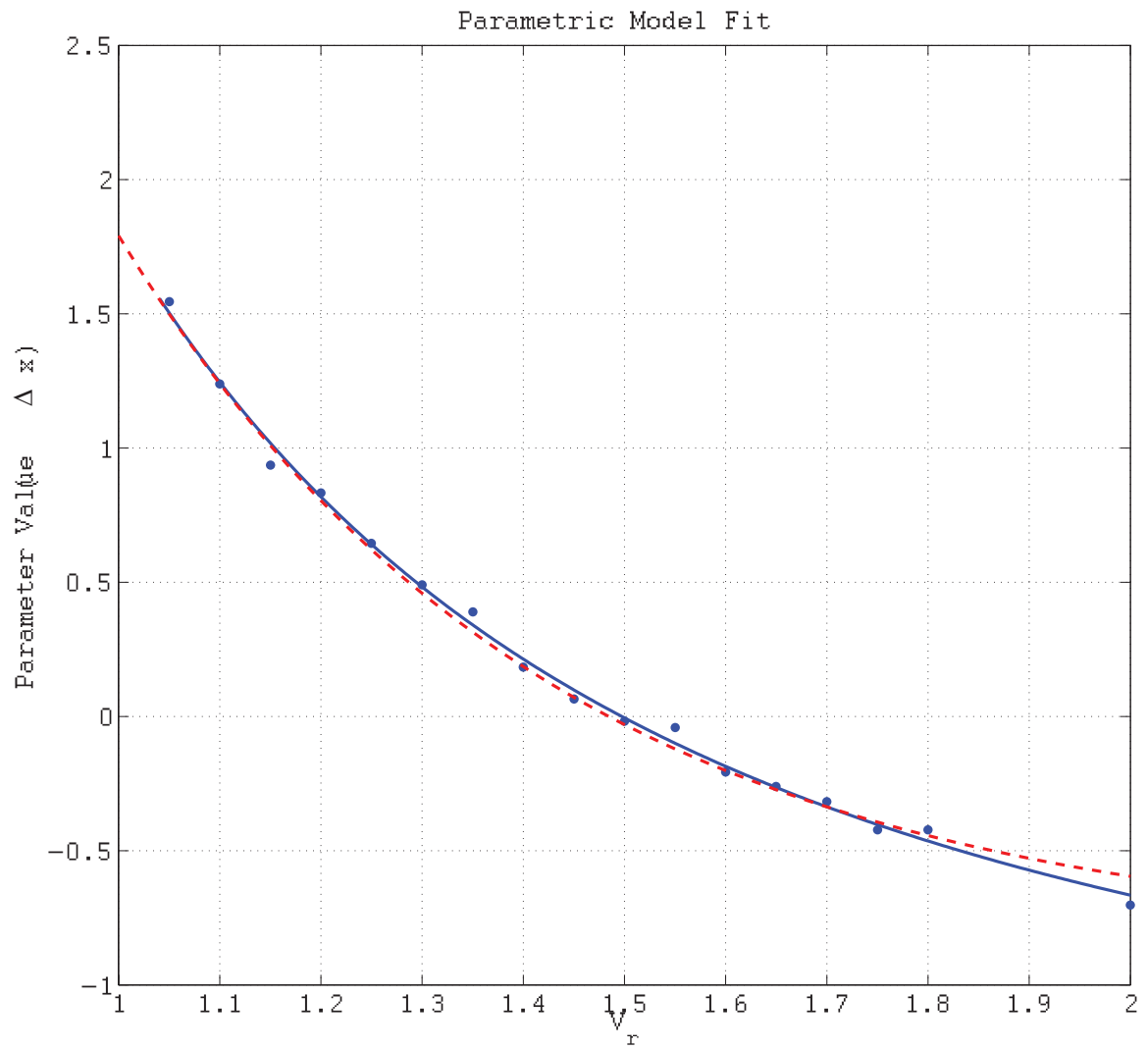


Figure 3.37. Estimated values of $\Delta\tilde{x}$ and parametric fit to the data. Dashed line represents original parameter values as defined in [94].

Figure 3.38 shows a vertical bar printed on an HP Color LaserJet 4500 and subsequently scanned using an Epson 4490 at 4800dpi. The modulation signal used is

$$V_r(i) = \begin{cases} 1.0 & 1 \leq i \leq 15 \\ 1.5 & 16 \leq i \leq 30 \end{cases}, \quad (3.53)$$

repeated for the length of the edge. Two simulated images are shown. In Figure 3.39 the image was simulated using the parameters from [94]. In Figure 3.40 the image was simulated using the modified parameters. The original parameters for a 600 DPI print process as chosen in [94] are

$$\mu = 0.3, \quad (3.54)$$

$$\Lambda = 0.7155V_r - 0.3741 = \begin{cases} 0.3414 & , \quad V_r = 2 \\ 0.6992 & , \quad V_r = 1.5 \end{cases}, \quad (3.55)$$

$$\Delta x = 27.42e^{-2.34*V_r} - 0.85 = \begin{cases} -0.5956 & , \quad V_r = 2 \\ -0.0302 & , \quad V_r = 1.5 \end{cases}. \quad (3.56)$$

The new parameters for the signal in Equation 3.53 are

$$\mu = \begin{cases} 0.1821 & , \quad V_r = 2 \\ 0.2841 & , \quad V_r = 1.5 \end{cases}, \quad (3.57)$$

$$\Lambda = \begin{cases} 0.2203 & , \quad V_r = 2 \\ 0.5361 & , \quad V_r = 1.5 \end{cases}, \quad (3.58)$$

$$\Delta \tilde{x} = \begin{cases} 0.0744 & , \quad V_r = 2 \\ 0.6888 & , \quad V_r = 1.5 \end{cases}. \quad (3.59)$$

Figures 3.41 and 3.42 show the estimated edge profiles and edge sharpness of the left and right edges for each of the three vertical bars. The red curve corresponds to Figure 3.39, the green curve to Figure 3.40, and the blue curve to Figure 3.38. It is seen that the original parameters create a much sharper edge and larger edge offsets than the empirical edge. The newly obtained parameters for the given V_r , however, match the empirical edge much more accurately.

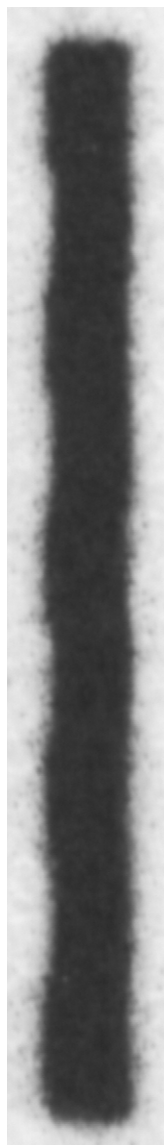


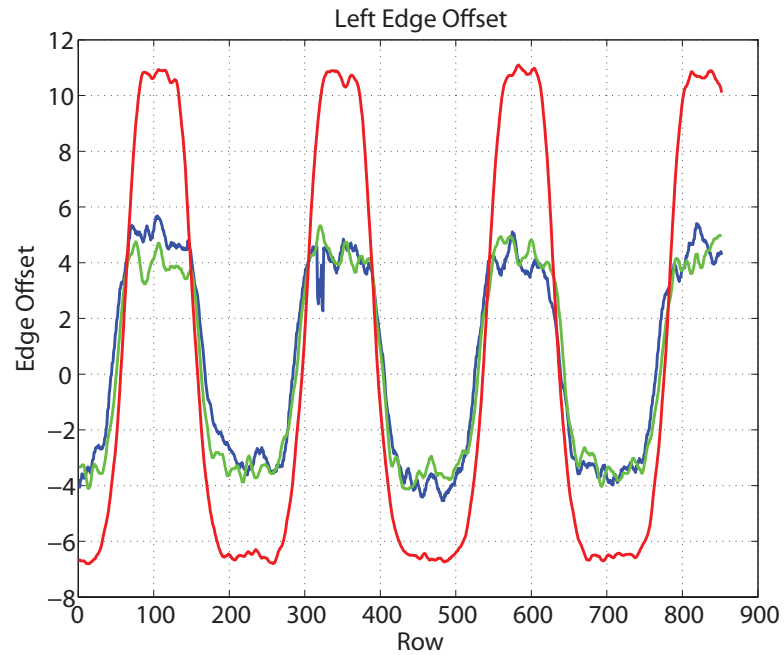
Figure 3.38. Vertical bar printed on an HP Color LaserJet 4500 at 600 DPI and scanned using an Epson 4490 at 4800 DPI. Dimensions of the bar as printed are $140 \times 10\text{px}$ at 600 DPI or approximately 0.233×0.0167 inches.



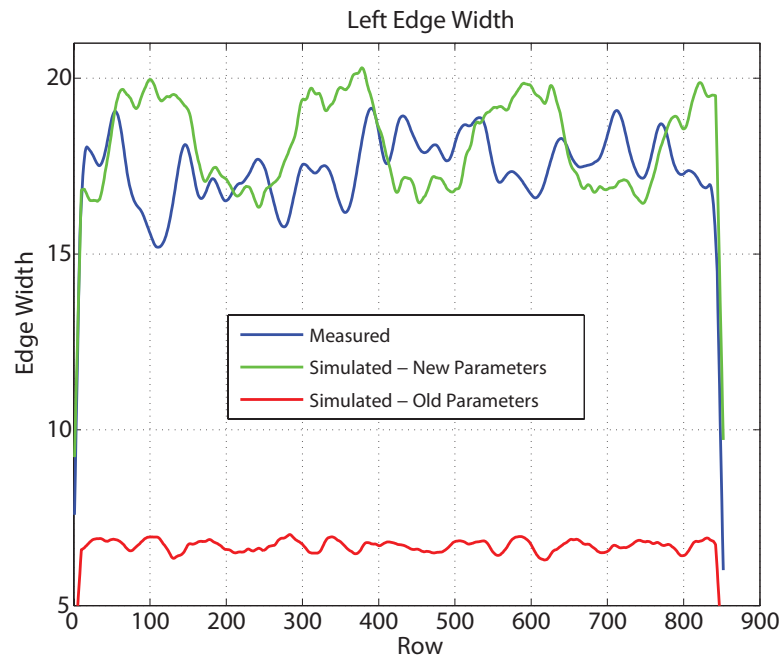
Figure 3.39. Vertical bar simulated using Chiang's method with original parameters.



Figure 3.40. Vertical bar simulated using Chiang's method with new parameters.

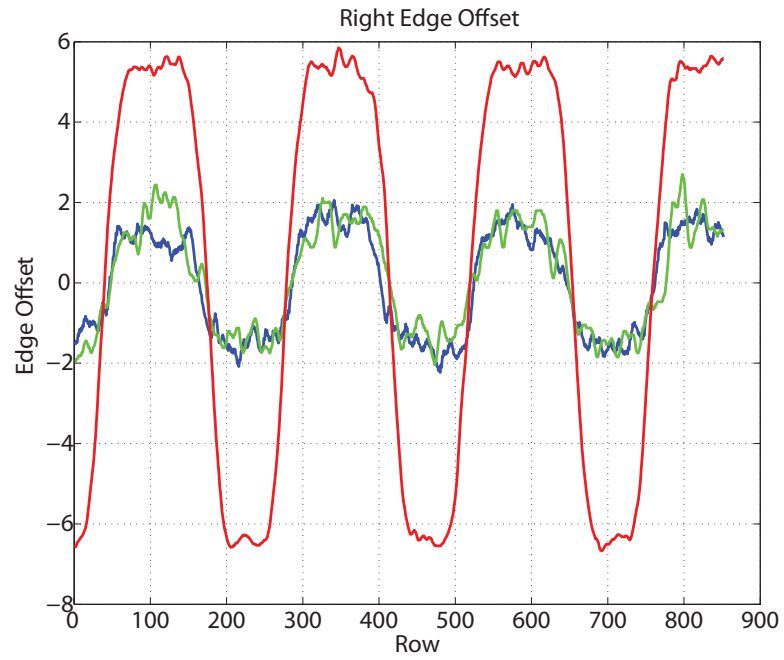


(a)

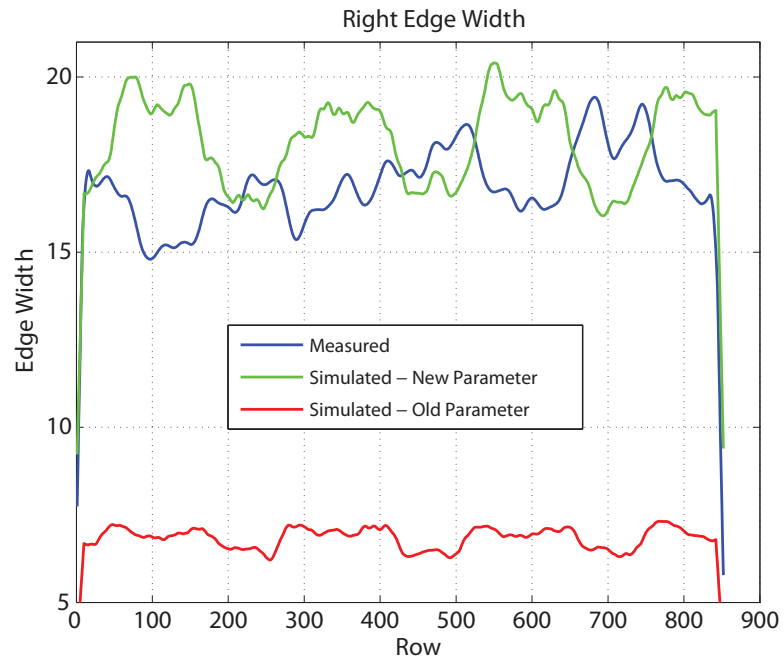


(b)

Figure 3.41. Results of simulation methods and experiments, embedding a square wave with amplitude 0.5. Subfigure (a) shows the relative edge offset of the left edge, while subfigure (b) shows the 20% to 80% transition width of the left edge. Red - Chiang's method with original parameters. Green - Chiang's method with new parameters. Blue - Measured data.



(a)



(b)

Figure 3.42. Results of simulation methods and experiments, embedding a square wave with amplitude 0.5. Subfigure (a) shows the relative edge offset of the right edge, while subfigure (b) shows the 20% to 80% transition width of the right edge. Red - Chiang's method with original parameters. Green - Chiang's method with new parameters. Blue - Measured data.

4. SUMMARY AND FUTURE DIRECTIONS

There currently exist techniques to secure documents such as bank notes using paper watermarks, security fibers, holograms, or special inks [8–11]. The problem is that the use of these security techniques can be cost prohibitive. Most of these techniques either require special equipment to embed the security features, or are too expensive for an average consumer. Additionally, there are a number of applications in which it is desirable to be able to identify the technology, manufacturer, model, or specific unit that was used to print a given document even if the printer in question does not make use of these existing security devices to explicitly identify itself. It would be useful to achieve the same or better level of protection without the use of any additional devices or technologies.

Two strategies are proposed for printer identification based upon examination of a printed document. The first strategy is passive. It involves characterization of the printer by finding features in the printed document that are intrinsic to that particular printer, model, or manufacturer’s products. This is referred to as the *intrinsic signature* [14].

The second strategy is active. It involves the embedding of an *extrinsic signature* into a printed page. This signature can be generated by modulating the process parameters of the printer mechanism to encode identifying information such as the printer serial number and date of printing. Detection of the extrinsic signature can be done using the tools developed for intrinsic signature detection.

4.1 Summary

The results presented in Section 2 indicate that good separation of materials produced by different printers is achievable when using graylevel co-occurrence based

texture features extracted from individual text characters in a printed document. Experiments using ten printers and a support vector machine classifier showed very low printer classification error even between printers with the same electromechanical structure. The technique was also shown to work for various font sizes, font types, paper types, and printer age when other variables are held constant. These results are very promising because these are the types of scenarios that would be encountered in a real forensic situation. Similarly, in Section 2.5.1 it was shown how the features migrate with consumable (toner cartridge) age indicating that it may be possible to estimate the age of the consumables at the time of printing.

The intrinsic nature of the features makes it difficult to obscure or remove them without physically modifying the printer itself. In Section 2.4.2 it was shown that by combining both texture features and banding features it is possible to identify a printer under several attack scenarios.

In Section 3 the development of a coding technique for the embedding of extrinsic signatures in text documents using the method developed in [46] was presented. Both time and frequency domain signaling and detection schemes were investigated. It was shown that performance can be improved by using a time domain signaling scheme with a correlation detector due to the limited length of text character edges. It has been shown that by treating the document as a communication channel, a coding technique allowing 3600 bits per page in 12 point text (assuming approximately 900 embeddable characters per page) with a 7.74% bit error rate is achievable.

A counterfeit and tamper detection method based on combinatorial group testing has been developed and investigated. The low error rate achievable by the data hiding system presented in Section 3.5 allowed reliable determination of document authenticity, and tampered data within a document.

In previous work reported in [94] a printer dot model to simulate the printing of cluster-dot halftone patterns was proposed. It has been shown that the original parameters chosen for that model do not adequately represent vertical edges in saturated regions such as those that occur in text. Estimating the parameters by minimizing

the error between the simulated and experimental edge profiles and edge sharpness for both the left and right edges provided values that more accurately represent the actual edge with and without embedded signals.

4.2 Contributions Resulting From This Research

In this thesis, techniques for intrinsic printer identification and extrinsic signature embedding are described. In this research, printer identification using texture features, and signature embedding in text documents have been investigated. The primary contributions in the area of intrinsic printer identification are as follows:

- The use of gray-level co-occurrence texture features as an intrinsic signature has been investigated. These features are shown to be robust to printer and consumable age as well as small changes in font size, font type, and paper type. In addition, the features are shown to exhibit behavior that may allow determination of consumable age.
- Combined use of gray-level co-occurrence features and banding features were investigated. These features when used together are shown to be robust against several attack modes that attempt to obscure or destroy the intrinsic signature of a printer.

The primary contributions in the area of extrinsic signature embedding and printer identification are as follows:

- A channel model for printed text documents was developed to aid in the embedding and detection of extrinsic signatures.
- Time and frequency domain embedding have been investigated with the use of both correlation, spectral analysis, and matched filter based detectors. It is shown that up to 3600 bits can be reliably embedded within a page of 12 point text.

- A method to detect counterfeit and content-tampering has been proposed that is based on a combinatorial group testing framework. It is shown that by using the proposed embedding system, document originality can be determined, and changes to key elements in a document can be identified.
- Parameters for a printer dot model that characterize printed edges in text have been obtained and have been shown to provide a good representation of printed output matching that of the physical embedding system.

4.3 Future Directions

The intrinsic signature based printer identification presented in this thesis provides a very useful tool for forensic investigators. One of the limitations at this point is that the features being used are dependent upon the font type and size. Making the features independent of the shape and size of the text would negate the need to retrain the classifier whenever an unknown document with a new typeface is encountered. One way of doing this may be to simply weight the graylevel co-occurrence matrix based on the shape of the region it is estimated from.

It was shown that several features related to the intrinsic signature of a printer may be useful in determining the age of the printer's consumables such as the toner cartridge or imaging drum. Further work quantifying the behavior of those changes over time may be useful to a forensic investigator in determining the time, or period of time, over which a document or set of documents were printed. The movement of these features over time may themselves be useful for printer identification.

The attacks considered for attempting to defeat the printer identification system were very simple, however other non-malicious attacks could also be considered. One example would be to print a text document in a composite black which would introduce signatures from multiple developer units into each character, as well as increase toner density in the printed region. The effect of a composite black on the printer

identification system presented in this thesis is unknown at this point and merits investigation.

Although only two specific embedding and detection techniques have been investigated, the set of all possible marking techniques can be abstracted and used to create a general capacity bound that can be used to judge both data hiding and signature embedding techniques. The types of marks that can be produced by a given electrophotographic printer is limited, and is further reduced by limiting the set of marks to those that do not create any perceptual difference in the document. Designing models to capture these perceptual differences can be useful in optimizing the marking technique itself.

The above two metrics could then be used to improve the counterfeit and tamper detection data hiding system by allowing both robust and fragile marks to be embedded into a document in a way that would allow copy, or copy generation detection (fragile marks) while still retaining tracking or tamper information (robust marks).

4.4 Publications Resulting From This Work

JOURNAL PUBLICATIONS

1. **A. K. Mikkilineni**, G. T.-C. Chiu, E. J. Delp, “Extrinsic Signatures for Printer Identification and Forensics,” 2012, to be submitted.
2. **A. K. Mikkilineni**, N. Khanna, G. T.-C. Chiu, E. J. Delp, “Intrinsic Signatures for Printer Identification and Forensics,” 2012, to be submitted.
3. N. Khanna, **A. K. Mikkilineni**, E. J. Delp, “Scanner identification using feature-based processing and analysis,” *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 1, pp. 123–139, March 2009.
4. P.-J. Chiang, N. Khanna, **A. K. Mikkilineni**, M. V. O. Segovia, S. Suh, J. P. Allebach, G. T.-C. Chiu, E. J. Delp, “Printer and scanner forensics,” *IEEE Signal Processing Magazine*, vol. 26, no. 2, pp. 72–83, March 2009.

BOOK CHAPTERS

1. P.-J. Chiang, N. Khanna, **A. K. Mikkilineni**, M. V. O. Segovia, J. P. Allebach, G. T.-C. Chiu, E. J. Delp, authors, “Printer and Scanner Forensics: Models and Methods,” *Intelligent Multimedia Analysis for Security Applications*, H. T. Sencar, S. Velastin, N. Nikolaidis, S. Lian, Eds., Berlin, Springer-Verlag, 2010, pp. 145-187.

CONFERENCE PUBLICATIONS

1. S. Palakodety, **A. K. Mikkilineni**, M. Atallah, E. J. Delp, “Is this hardcopy an original?,” 2012, to be submitted.
2. **A. K. Mikkilineni**, N. Khanna, E. J. Delp, “Forensic printer detection using intrinsic signatures,” *Proceedings of the SPIE International Conference on Media Watermarking, Security, and Forensics III*, vol. 7880, San Francisco, CA, USA, January 2011.
3. **A. K. Mikkilineni**, N. Khanna, E. J. Delp, “Texture based attacks on intrinsic signature based printer identification,” *Proceedings of the SPIE International Conference on Media Forensics and Security II*, vol. 7541, San Jose, CA, USA, January 2010.
4. **A. K. Mikkilineni**, G. T.-C. Chiu, J. P. Allebach, E. J. Delp, “High-capacity data hiding in text documents,” *Proceedings of the SPIE International Conference on Media Forensics and Security*, vol. 7254, San Jose, CA, USA, 2009, p. 72540X.
5. N. Khanna, **A. K. Mikkilineni**, G. T.-C. Chiu, J. P. Allebach, E. J. Delp, “Survey of scanner and printer forensics at Purdue University,” *Proceedings of the Second International Workshop on Computational Forensics (IWCF 2008)*, Washington, DC, USA, August 2008, pp. 22–34.

6. **A. K. Mikkilineni**, P.-J. Chiang, G. T.-C. Chiu, J. P. Allebach, E. J. Delp, "Channel model and operational capacity analysis of printed text documents," *Proceedings of the SPIE International Conference Security, Steganography, and Watermarking of Multimedia Contents IX*, vol. 6505, San Jose, CA, USA, February 2007.
7. N. Khanna, **A. K. Mikkilineni**, P.-J. Chiang, M. V. Ortiz, S. Suh, G. T.-C. Chiu, J. P. Allebach, E. J. Delp, "Sensor forensics: Printers, cameras and scanners, they never lie," *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, Beijing, China, July 2007, pp. 20–23.
8. P.-J. Chiang, **A. K. Mikkilineni**, E. J. Delp, J. P. Allebach, G. T.-C. Chiu, "Development of an electrophotographic laser intensity modulation model for extrinsic signature embedding," *Proceedings of the IS&T's NIP23: International Conference on Digital Printing Technologies and Digital Fabrication*, vol. 23, Anchorage, AK, USA, September 2007, pp. 561–564.
9. **A. K. Mikkilineni**, P.-J. Chiang, S. Suh, G. T.-C. Chiu, J. P. Allebach, E. J. Delp, "Information embedding and extraction for electrophotographic printing processes," *Proceedings of the SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents VIII*, vol. 6072, San Jose, CA, USA, January 2006, pp. 385–396.
10. O. Arslan, R. M. Kumontoy, P.-J. Chiang, **A. K. Mikkilineni**, J. P. Allebach, G. T.-C. Chiu, E. J. Delp, "Identification of inkjet printers for forensic applications," *Proceedings of the IS&T's NIP21: International Conference on Digital Printing Technologies*, vol. 21, Baltimore, MD, USA, September 2005, pp. 235–238.
11. P.-J. Chiang, **A. K. Mikkilineni**, O. Arslan, R. M. Kumontoy, G. T.-C. Chiu, E. J. Delp, J. P. Allebach, "Extrinsic signature embedding in text document using exposure modulation for information hiding and secure printing in elec-

- trophotography,” *Proceedings of the IS&T’s NIP21: International Conference on Digital Printing Technologies*, vol. 21, Baltimore, MD, USA, September 2005, pp. 231–234.
12. **A. K. Mikkilineni**, O. Arslan, P.-J. Chiang, R. M. Kumontoy, J. P. Allebach, G. T.-C. Chiu, E. J. Delp, “Printer forensics using SVM techniques,” *Proceedings of the IS&T’s NIP21: International Conference on Digital Printing Technologies*, vol. 21, Baltimore, MD, USA, September 2005, pp. 223–226.
 13. **A. K. Mikkilineni**, P.-J. Chiang, G. N. Ali, G. T.-C. Chiu, J. P. Allebach, E. J. Delp, “Printer identification based on graylevel co-occurrence features for security and forensic applications,” *Proceedings of the SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents VII*, vol. 5681, San Jose, CA, USA, March 2005, pp. 430–440.
 14. G. N. Ali, P.-J. Chiang, **A. K. Mikkilineni**, G. T.-C. Chiu, E. J. Delp, J. P. Allebach, “Application of principal components analysis and gaussian mixture models to printer identification,” *Proceedings of the IS&T’s NIP20: International Conference on Digital Printing Technologies*, vol. 20, Salt Lake City, UT, USA, October 2004, pp. 301–305.
 15. P.-J. Chiang, G. N. Ali, **A. K. Mikkilineni**, G. T.-C. Chiu, J. P. Allebach, E. J. Delp, “Extrinsic signatures embedding using exposure modulation for information hiding and secure printing in electrophotographic devices,” *Proceedings of the IS&T’s NIP20: International Conference on Digital Printing Technologies*, vol. 20, Salt Lake City, UT, USA, October 2004, pp. 295–300.
 16. **A. K. Mikkilineni**, P.-J. Chiang, G. N. Ali, G. T.-C. Chiu, J. P. Allebach, E. J. Delp, “Printer Identification based on textural features,” *Proceedings of the IS&T’s NIP20: International Conference on Digital Printing Technologies*, vol. 20, Salt Lake City, UT, October 2004, pp. 306–311.

17. **A. K. Mikkilineni**, G. N. Ali, P.-J. Chiang, G. T.-C. Chiu, J. P. Allebach, E. J. Delp, "Signature-embedding in printed documents for security and forensic applications," *Proceedings of the SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents VI*, vol. 5306, San Jose, CA, USA, January 2004, pp. 455–466.
18. G. N. Ali, P.-J. Chiang, **A. K. Mikkilineni**, J. P. Allebach, G. T.-C. Chiu, E. J. Delp, "Intrinsic and extrinsic signatures for information hiding and secure printing with electrophotographic devices," *Proceedings of the IS&T's NIP19: International Conference on Digital Printing Technologies*, vol. 19, New Orleans, LA, USA, September 2003, pp. 511–515.

LIST OF REFERENCES

LIST OF REFERENCES

- [1] S. Suh, J. P. Allebach, G. T.-C. Chiu, and E. J. Delp, "Printer mechanism-level data hiding for halftone documents," *Proceedings of the IS&T's NIP22: International Conference on Digital Printing Technologies*, vol. 22, Denver, CO, USA, September 2006, pp. 436–440.
- [2] A. M. Eskicioglu and E. J. Delp, "An overview of multimedia content protection in consumer electronics devices," *Signal Processing : Image Communication*, vol. 16, no. 7, pp. 681–699, April 2001.
- [3] A. M. Eskicioglu, J. Town, and E. J. Delp, "Security of digital entertainment content from creation to consumption," *Signal Processing : Image Communication*, vol. 18, no. 4, pp. 237–262, April 2003.
- [4] C. I. Podilchuk and E. J. Delp, "Digital watermarking: Algorithms and applications," *IEEE Signal Processing Magazine*, vol. 18, no. 4, pp. 33–46, July 2001.
- [5] M. Barni, C. I. Podilchuk, F. Bartolini, and E. J. Delp, "Watermark embedding: hiding a signal within a cover image," *IEEE Communications Magazine*, vol. 39, no. 8, pp. 102–108, August 2001.
- [6] B. Macq, J. Dittmann, and E. J. Delp, "Benchmarking of image watermarking algorithms for digital rights management," *Proceedings of the IEEE*, vol. 92, no. 6, pp. 971–984, June 2004.
- [7] R. W. Wolfgang, C. I. Podilchuk, and E. J. Delp, "Perceptual watermarks for digital images and video," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1108–1126, July 1999.
- [8] R. L. van Renesse, "Paper based document security - a review," *European Conference on Security and Detection, ECOS97*, London, UK, April 1997, pp. 75–80.
- [9] R. L. Renesse, *Optical Document Security*. Boston, MA, USA: Artech House, 1998.
- [10] D. Tyagi, M. Zaretsky, T. Tombs, and P. Lambert, "Use of clear toner in electrophotography for security applications," *Proceedings of the IS&T's NIP24: International Conference on Digital Printing Technologies and Digital Fabrication*, vol. 24, Pittsburgh, PA, USA, 2008, pp. 773–776.
- [11] M. Schmid, "Optical security in ink: an industry standard that continues to evolve," *Proceedings of the SPIE International Conference on Optical Security and Counterfeit Deterrence Techniques VI*, vol. 6075, San Jose, CA, USA, February 2006, pp. 265–270.

- [12] S. Simske, J. Aronoff, M. Sturgill, and G. Golodetz, "Security printing deterrents: a comparison of tij, dep and lep printing," *Proceedings of the IS&T's NIP23: International Conference on Digital Printing Technologies and Digital Fabrication*, vol. 23, Anchorage, AK, USA, September 2007, pp. 543–548.
- [13] M. Gaubatz and S. Simske, "Printer-scanner identification via analysis of structured security deterrents," *Proceedings of the First IEEE International Workshop on Information Forensics and Security (WIFS 2009)*, London, UK, December 2009, pp. 151–155.
- [14] A. K. Mikkilineni, G. N. Ali, P.-J. Chiang, G. T.-C. Chiu, J. P. Allebach, and E. J. Delp, "Signature-embedding in printed documents for security and forensic applications," *Proceedings of the SPIE/IS&T Conference on Security, Steganography and Watermarking of Multimedia Contents VI*, E. J. D. III and P. W. Wong, Eds., vol. 5306, San Jose, CA, USA, January 2004, pp. 455–466.
- [15] G. N. Ali, P.-J. Chiang, A. K. Mikkilineni, J. P. Allebach, G. T. Chiu, and E. J. Delp, "Intrinsic and extrinsic signatures for information hiding and secure printing with electrophotographic devices," *Proceedings of the IS&T's NIP19: International Conference on Digital Printing Technologies*, vol. 19, New Orleans, LA, USA, September 2003, pp. 511–515.
- [16] G.-Y. Lin, J. M. Grice, J. P. Allebach, G. T.-C. Chiu, W. Bradburn, and J. Weaver, "Banding artifact reduction in electrophotographic printers by using pulse width modulation," *Journal of Imaging Science and Technology*, vol. 46, no. 4, pp. 326–337, July/August 2002.
- [17] C.-L. Chen and G. T.-C. Chiu, "Banding reduction in electrophotographic process," *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, vol. 1, Como, Italy, July 2001, pp. 81–86.
- [18] M. Chen, E. K. Wong, N. Memon, and S. Adams, "Recent developments in document image watermarking and data hiding," *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 4518, Denver, CO, USA, August 2001, pp. 166–176.
- [19] J. Oliver and J. Chen, "Use of signature analysis to discriminate digital printing technologies," *Proceedings of the IS&T's NIP18: International Conference on Digital Printing Technologies*, vol. 18, San Diego, CA, USA, September 2002, pp. 218–222.
- [20] T. Tanaka, "Metrology of small scale features in electrophotographic non-image areas as forensic evidence," *Proceedings of the IS&T's NIP17: International Conference on Digital Printing Technologies*, vol. 17, Fort Lauderdale, FL, USA, September 2001, pp. 590–593.
- [21] J. Tchan, "Classifying digital prints according to their production process using image analysis and artificial neural networks," *Proceedings of the SPIE International Conference on Optical Security and Counterfeit Deterrence Techniques III*, vol. 3973, San Jose, CA, USA, 2000, pp. 105–116.
- [22] J. T. Brassil, S. Low, and N. F. Maxemchuk, "Copyright protection for the electronic distribution of text documents," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1181–1196, July 1999.

- [23] J. T. Brassil, S. Low, N. F. Maxemchuk, and L. O’Gorman, “Electronic marking and identification techniques to discourage document copying,” *IEEE Journal on Selected Areas in Communication*, vol. 13, no. 8, pp. 1495–1504, October 1995.
- [24] S. H. Low and N. F. Maxemchuk, “Capacity of text marking channel,” *IEEE Signal Processing Letters*, vol. 7, no. 12, pp. 345–347, December 2000.
- [25] K. Kaneda, K. Hirano, K. Iwamura, and S. Hangai, “Information hiding method utilizing low visible natural fiber pattern for printed documents,” *Proceedings of the IEEE International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP ’08)*, Harbin, China, August 2008, pp. 319–322.
- [26] R. Villan, S. Voloshynovskiy, O. Koval, J. Vila, E. Topak, F. Deguillaume, Y. Rytsar, and T. Pun, “Text data-hiding for digital and printed documents: Theoretical and practical considerations,” *Proceedings of SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents VIII*, vol. 6072, San Jose, CA, USA, January 2006, pp. 406–416.
- [27] R. Villan, S. Voloshynovskiy, O. Koval, F. Deguillaume, and T. Pun, “Tamper-proofing of electronic and printed text documents via robust hashing and data-hiding,” *Proceedings of SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents IX*, vol. 6505, San Jose, CA, USA, February 2007, p. 65051T.
- [28] P. Borges, J. Mayer, and E. Izquierdo, “Robust and transparent color modulation for text data hiding,” *IEEE Transactions on Multimedia*, vol. 10, no. 8, pp. 1479–1489, December 2008.
- [29] A. K. Bhattacharjya and H. Ancin, “Data embedding in text for a copier system,” *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, vol. 2, Kobe, Japan, 1999, pp. 245–249.
- [30] S. Das, S. Rane, and A. Vetro, “Hiding information inside structured shapes,” *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Dallas, TX, USA, March 2010, pp. 1782–1785.
- [31] Q. Mei, E. Wong, and N. Memon, “Data hiding in binary text documents,” *Proceedings of SPIE International Conference on Security and Watermarking of Multimedia Contents III*, vol. 4314, San Jose, CA, United states, 2001, pp. 369–375.
- [32] T. N. Pappas, J. P. Allebach, and D. L. Neuhoff, “Model-based digital halftoning,” *IEEE Signal Processing Magazine*, vol. 20, no. 4, pp. 14–27, July 2003.
- [33] M. S. Fu and O. C. Au, “Data hiding watermarking for halftone images,” *IEEE Transactions on Image Processing*, vol. 11, no. 4, pp. 477–484, April 2002.
- [34] O. Bulan, V. Monga, G. Sharma, and B. Oztan, “Data embedding in hard-copy images via halftone-dot orientation modulation,” *Proceedings of the SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents X*, vol. 6819, San Jose, CA, USA, January 2008, p. 68190C.

- [35] G. Sharma and S. Wang, "Show-through watermarking of duplex printed documents," *Proceedings of the SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents VI*, vol. 5306, San Jose, CA, USA, June 2004, pp. 670–684.
- [36] Z. Fan, "Information embedding using two-layer conjugate screening," *Proceedings of the SPIE International Conference on Optical Security and Counterfeit Deterrence Techniques V*, vol. 5310, San Jose, CA, USA, June 2004, pp. 170–175.
- [37] S. V. Voloshynovskiy, F. Deguillaume, and T. Pun, "Multibit digital watermarking robust against local nonlinear geometrical distortions," *Proceedings of the IEEE International Conference on Image Processing*, vol. 3, Thessaloniki, Greece, October 2001, pp. 999–1002.
- [38] —, "Content adaptive watermarking based on a stochastic multiresolution image modeling," *Proceedings of the Tenth European Signal Processing Conference (EUSIPCO 2000)*, vol. 4, Tampere, Finland, September 2000, pp. 1953–1956.
- [39] S. V. Voloshynovskiy, F. Deguillaume, S. Pereira, and T. Pun, "Optimal adaptive diversity watermarking with channel state estimation," *Proceedings of the SPIE International Conference on Security and Watermarking of Multimedia Contents III*, vol. 4314, San Jose, CA, USA, 2001, pp. 673–685.
- [40] F. Deguillaume, S. V. Voloshynovskiy, and T. Pun, "Method for the estimation and recovering from general affine transforms in digital watermarking applications," *Proceedings of the SPIE International Conference on Security and Watermarking of Multimedia Contents IV*, vol. 4675, San Jose, CA, USA, April 2002, pp. 313–322.
- [41] D. Kacker and J. P. Allebach, "Joint halftoning and watermarking," *IEEE Transactions on Signal Processing*, vol. 51, no. 4, pp. 1054–1068, April 2003.
- [42] D. J. Lieberman and J. P. Allebach, "A dual interpretation for direct binary search and its implications for tone reproduction and texture quality," *IEEE Transactions on Image Processing*, vol. 9, no. 11, pp. 1950–1963, November 2000.
- [43] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Transactions on Image Processing*, vol. 6, no. 12, pp. 1673–1687, December 1997.
- [44] S. Suh, J. P. Allebach, G. T.-C. Chiu, and E. J. Delp, "Printer mechanism-level data hiding for halftone documents," *Proceedings of the IS&T's NIP22: International Conference on Digital Printing Technologies*, vol. 22, Denver, CO, USA, September 2006, pp. 436–440.
- [45] O. Arslan, R. M. Kumontoy, P. ju Chiang, A. K. Mikkilineni, J. P. Allebach, G. T. C. Chiu, and E. J. Delp, "Identification of inkjet printers for forensic applications," *Proceedings of the IS&T's NIP21: International Conference on Digital Printing Technologies*, vol. 21, Baltimore, MD, USA, September 2005, pp. 235–238.
- [46] P.-J. Chiang, G. N. Ali, A. K. Mikkilineni, G. T.-C. Chiu, J. P. Allebach, and E. J. Delp, "Extrinsic signatures embedding using exposure modulation for information hiding and secure printing in electrophotographic devices," *Proceedings of the IS&T's NIP20: International Conference on Digital Printing Technologies*, vol. 20, Salt Lake City, UT, USA, October 2004, pp. 295–300.

- [47] G. N. Ali, P.-J. Chiang, A. K. Mikkilineni, J. P. Allebach, G. T. Chiu, and E. J. Delp, "Intrinsic and extrinsic signatures for information hiding and secure printing with electrophotographic devices," *Proceedings of the IS&T's NIP19: International Conference on Digital Printing Technologies*, vol. 19, New Orleans, LA, USA, September 2003, pp. 511–515.
- [48] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-3, no. 6, pp. 610–621, November 1973.
- [49] R. W. Connors, M. M. Trivedi, and C. A. Harlow, "Segmentation of a high-resolution urban scene using texture operators," *Computer Vision, Graphics, and Image Processing*, vol. 25, no. 3, pp. 273–310, March 1984.
- [50] A. K. Mikkilineni, P.-J. Chiang, G. N. Ali, G. T.-C. Chiu, J. P. Allebach, and E. J. Delp, "Printer identification based on textural features," *Proceedings of the IS&T's NIP20: International Conference on Digital Printing Technologies*, vol. 20, Salt Lake City, UT, October 2004, pp. 306–311.
- [51] A. K. Mikkilineni, P.-J. Chiang, G. N. Ali, G. T. C. Chiu, J. P. Allebach, and E. J. Delp, "Printer identification based on graylevel co-occurrence features for security and forensic applications," *Proceedings of the SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents VII*, vol. 5681, San Jose, CA, USA, March 2005, pp. 430–440.
- [52] C. R. Giardina and E. R. Dougherty, *Morphological Methods in Image and Signal Processing*. Englewood Cliffs, NJ, USA: Prentice Hall, 1988.
- [53] Y. Sun, "Normal mammogram analysis," Ph.D. dissertation, Purdue University, West Lagayette, IN, August 2004.
- [54] R. W. Connors, "Towards a set of statistical features which measure visually perceivable qualities of textures," *Proceedings of the IEEE Computer Society Conference on Pattern Recognition Image Processing*, August 1979, pp. 382–390.
- [55] R. W. Connors and C. A. Harlow, "Towards a structural textural analyzer based on statistical methods," *Computer Graphics and Image Processing*, vol. 12, no. 3, pp. 224–256, March 1980.
- [56] M. Wertheimer, "Principles of perceptual organization," *Readings in Perception*, D. Beardslee and M. Wertheimer, Eds. New York, NY, USA: Van Nostrand, 1958, pp. 115–134.
- [57] B. Julesz, "Visual pattern discrimination," *IRE Transactions on Information Theory*, vol. 8, no. 2, pp. 84–92, February 1962.
- [58] T. McConnell. (1999) Monkey: a markov chain random text generating program. [Online]. Available: <http://barnyard.syr.edu/~tmc/>
- [59] Project gutenber. [Online]. Available: <http://www.gutenberg.net/>
- [60] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. San Diego, CA, USA: Academic Press, 1990.

- [61] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, "An introduction to kernel-based learning algorithms," *IEEE Transactions on Neural Networks*, vol. 12, no. 2, pp. 181–201, March 2001.
- [62] N. Christianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge, UK: Cambridge University Press, 2000.
- [63] B. Schölkopf, C. J. Burges, and A. J. Smola, *Advances in Kernel Methods: Support Vector Machines*. MIT Press, Cambridge, MA, 1999.
- [64] S. Knerr, L. Personnaz, and G. Dreyfus, "Single-layer learning revisited: A step-wise procedure for building and training a neural network," *Neurocomputing: Algorithms, Architectures and Applications*, ser. NATO ASI Series, F. Fogelman Soulié and J. Hérault, Eds. Springer-Verlag, 1990, vol. F68, pp. 41–50.
- [65] A. K. Mikkilineni, O. Arslan, P.-J. Chiang, R. M. Kumontoy, J. P. Allebach, G. T.-C. Chiu, and E. J. Delp, "Printer forensics using SVM techniques," *Proceedings of the IS&T's NIP21: International Conference on Digital Printing Technologies*, vol. 21, Baltimore, MD, USA, September 2005, pp. 223–226.
- [66] T. Joachims, "Making large-scale support vector machine learning practical," *Advances in Kernel Methods: Support Vector Machines*, B. Schölkopf, C. Burges, and A. Smola, Eds. MIT Press, Cambridge, MA, 1998.
- [67] A. K. Mikkilineni, N. Khanna, and E. J. Delp, "Texture based attacks on intrinsic signature based printer identification," *Proceedings of the SPIE International Conference on Media Forensics and Security II*, vol. 7541, San Jose, CA, USA, January 2010.
- [68] —, "Forensic printer detection using intrinsic signatures," *Proceedings of the SPIE International Conference on Media Watermarking, Security, and Forensics III*, vol. 7880, San Francisco, CA, USA, January 2011.
- [69] P. Pudil, F. Ferri, J. Novovicova, and J. Kittler, "Floating search methods for feature selection with nonmonotonic criterion functions," *Proceedings of the 12th IAPR International Conference on Pattern Recognition*, vol. 2, Jerusalem, Israel, October 1994, pp. 279–283.
- [70] P. Somol, P. Pudil, J. Novovicova, and P. Paclik, "Adaptive floating search methods in feature selection," *Pattern Recognition Letters*, vol. 20, no. 11-13, pp. 1157–1163, November 1999.
- [71] A. Jain and D. Zongker, "Feature selection: Evaluation, application, and small sample performance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 153–158, February 1997.
- [72] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*, 2nd ed. New York, USA: Wiley-Interscience, 2001.
- [73] G.-Y. Lin, J. M. Grice, J. P. Allebach, G. T.-C. Chiu, W. Bradburn, and J. Weaver, "Banding artifact reduction in electrophotographic printers by using pulse width modulation," *The Journal of Imaging Science and Technology*, vol. 46, no. 4, pp. 326–337, July/August 2002.

- [74] C.-L. Chen, G. T.-C. Chiu, and J. P. Allebach, "Banding reduction in electrophotographic processes using human contrast sensitivity function shaped photoconductor velocity control," *Journal of Imaging Science and Technology*, vol. 47, no. 3, pp. 209–223, May/June 2003.
- [75] M. T. S. Ewe, G. T.-C. Chiu, J. M. Grice, J. P. Allebach, C. Chan, and W. Foote, "Banding reduction in electrophotographic processes using a piezoelectric actuated laser beam deflection device," *The Journal of Imaging Science and Technology*, vol. 46, no. 5, pp. 433–442, September/October 2002.
- [76] E. M. Williams, *The Physics and Technology of Xerographic Processes*. New York, NY: Wiley, 1984.
- [77] D. Kacker, T. Camis, and J. P. Allebach, "Electrophotographic processes embedded in direct binary search," *IEEE Transactions on Image Processing*, vol. 11, no. 3, pp. 243–257, March 2002.
- [78] P.-J. Chiang, A. K. Mikkilineni, O. Arslan, R. M. Kumontoy, G. T.-C. Chiu, E. J. Delp, and J. P. Allebach, "Extrinsic signature embedding in text document using exposure modulation for information hiding and secure printing in electrophotography," *Proceedings of the IS&T's NIP21: International Conference on Digital Printing Technologies*, vol. 21, Baltimore, MD, USA, September 2005, pp. 231–234.
- [79] A. K. Mikkilineni, P.-J. Chiang, S. Suh, G. T. C. Chiu, J. P. Allebach, and E. J. Delp, "Information embedding and extraction for electrophotographic printing processes," *Proceedings of the SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents VIII*, vol. 6072, San Jose, CA, USA, January 2006, pp. 385–396.
- [80] ISO, *Information technology – Office equipment – Measurement of image quality attributes for hardcopy output – Binary monochrome text and graphic images*, ISO Std. 13 660, 2001.
- [81] A. K. Mikkilineni, P.-J. Chiang, G. T.-C. Chiu, J. P. Allebach, and E. J. Delp, "Channel model and operational capacity analysis of printed text documents," *Proceedings of the SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents IX*, vol. 6505, San Jose, CA, USA, February 2007.
- [82] P.-J. Chiang, A. K. Mikkilineni, E. J. Delp, J. P. Allebach, and G. T.-C. Chiu, "Extrinsic signatures embedding and detection in electrophotographic halftone images through laser intensity modulation," *Proceedings of the IS&T's NIP22: International Conference on Digital Printing Technologies*, vol. 22, Denver, CO, USA, September 2006, pp. 432–435.
- [83] A. K. Mikkilineni, P.-J. Chiang, G. T.-C. Chiu, J. P. Allebach, and E. J. Delp, "Data hiding capacity and embedding techniques for printed text documents," *Proceedings of the IS&T's NIP22: International Conference on Digital Printing Technologies*, vol. 22, Denver, CO, USA, September 2006, pp. 444–447.
- [84] A. Varna, S. Rane, and A. Vetro, "Data hiding in hard-copy text documents robust to print, scan and photocopy operations," *Proceedings on the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Taipei, Taiwan, April 2009, pp. 1397–1400.

- [85] I. Free Software Foundation. (2006) Ocrad - the gnu ocr. [Online]. Available: <http://www.gnu.org/software/ocrad/>
- [86] S. Palakodety, A. K. Mikkilineni, M. Atallah, and E. J. Delp, "Is this hardcopy an original?" 2012, to be submitted.
- [87] D. R. Stinson, *Cryptography: Theory and Practice*, 2nd ed. Chapman & Hall/CRC, 2002.
- [88] B. Schneier, *Applied Cryptography*, 2nd ed. Wiley, October 1995.
- [89] D. C. Litzenger. (2012) Pycrypto - the python cryptography toolkit. [Online]. Available: <https://www.dlitz.net/software/pycrypto/>
- [90] A. K. Mikkilineni, G. T.-C. Chiu, J. P. Allebach, and E. J. Delp, "High-capacity data hiding in text documents," *Proceedings of the SPIE International Conference on Media Forensics and Security*, vol. 7254, San Jose, CA, USA, 2009, p. 72540X.
- [91] R. Dorfman, "The detection of defective members of large populations," *The Annals of Mathematical Statistics*, vol. 14, pp. 436–440, 1943.
- [92] D.-Z. Du and F. K. Hwang, *Combinatorial Group Testing and Its Applications*. World Scientific Publishing Co., 1999.
- [93] P.-J. Chiang, A. K. Mikkilineni, E. J. Delp, J. P. Allebach, and G. T.-C. Chiu, "Development of an electrophotographic laser intensity modulation model for extrinsic signature embedding," *Proceedings of the IS&T's NIP 23: International Conference on Digital Printing Technologies and Digital Fabrication*, vol. 23, Anchorage, AK, USA, September 2007, pp. 561–564.
- [94] P.-J. Chiang, "Extrinsic signatures embedding and detection in electrophotographic halftoned images through exposure modulation," Ph.D. dissertation, Purdue University, West Lagayette, IN, May 2009.
- [95] P. C. Julien and R. J. Gruber, "Dry toner technology," *Handbook of Imaging Materials*, A. S. Diamond and D. S. Weiss, Eds. New York, NY, USA: Marcel Dekker, Inc., 2002, pp. 173–208.
- [96] E. K. Chong and S. H. Zak, *An Introduction to Optimization*. Hoboken, NJ, USA: Wiley-Interscience, 2008.

VITA

VITA

Aravind K. Mikkilineni was born in Columbus, Ohio. He received the B.S. Electrical and Computer Engineering degree in 2002 from The Ohio State University, and the M.S. Electrical and Computer Engineering degree in 2004 from Purdue University.

He is a member of Eta Kappa Nu, IEEE, and a student member of both the ASME and IS&T.