Document Degradation Models and a Methodology for Degradation Model Validation

by

Tapas Kanungo

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

University of Washington

1996

Approved by____
(Chairperson of Supervisory Committee)

Program Authorized
to Offer Degree____________________________________

Date____

Signature______________________________

Date__________________________________

University of Washington

Abstract

Document Degradation Models and a Methodology for Degradation Model Validation

by Tapas Kanungo

Chairperson of Supervisory Committee: *Professor Robert M. Haralick*
Department of Electrical Engineering

Printing, photocopying and scanning processes degrade the image quality of a document. Although research in document understanding started in the sixties, only two document degradation models have been proposed thus far. Furthermore, no attempts have been made to rigorously validate them. In document understanding research, models for image degradations are crucial in many ways. Models allow us to (i) conduct controlled experiments to study the break-down points of the systems, (ii) create large data sets with groundtruth for training classifiers, (iii) design optimal noise removal algorithms, (iv) choose values for the free parameters of the algorithms, etc.

In this thesis two document degradation models are described. The first model accounts for local pixel-level degradations that occur while printing, photocopying and scanning a document. The second model accounts for the perspective and illumination distortions that occur while photocopying or scanning a thick, bound document. The local distortion model allows us the create large data sets of synthetically generated documents, *in any language,* along with the associated groundtruth information quite easily. Unlike isolated character databases, our data sets are a much better representation of the real world since they account for the real-world character and word occurence probabilities, and character and word bi-gram probabilities naturally. Moreover, since our methodology puts the text, layout, formatting, resolution, and font details of the document image under the experimenter's control, a large variety of controlled experiments that were not possible earlier are now possible.

Next, an automatic document registration and character groundtruthing procedure is described. This procedure produces very accurate character groundtruth for scanned documents in any language, which had not been possible until now. The method essentially registers the ideal image to a scanned version and then transforms the groundtruth associated with the ideal image through the registration transformation. This method can be used to generate groundtruth for documents in any language, and even FAXed documents. A data set having 33 English scanned document images with character groundtruth for 62000 symbols was created using this procedure.

A non-parametric statistical procedure for estimating the parameters of the local degradation model from a sample of real degraded documents is then discussed. The estimation procedure allows researchers to generate large data sets from small samples of real data. Such procedures for estimating parameters do not exist for other document degradation models. In fact, our approach can be easily adapted to estimate the parameters of other models as well.

Finally, a statistical methodology that can be used to validate the local degradation models is described. This method is based on a non-parametric, two-sample permutation test. A variant of the method allows approximate validation tests instead. Another standard statistical device – the power function – is then used to choose between algorithm variables such as distance functions. Since the validation and power function procedures are independent of the model, they can be used to validate any other degradation model. A method for comparing any two models is also described. It uses p-values associated with the estimated models to select the model that is closer to the real world.

TABLE OF CONTENTS

List of Figures v

List of Tables viii

Chapter 1: Introduction 1

1.1 Contributions 3

1.2 Overview 4

Chapter 2: Related Literature 6

2.1 Degradation Models 6

2.2 Document Registration 7

2.3 Statistical Model Validation 7

2.4 Document Degradation Model Validation 7

Chapter 3: Document Degradation Models 9

3.1 A Local Document Degradation Model 9

3.1.1 Implementation 10

3.1.2 Degrading Complete Pages 10

3.2 A Global Page Degradation Model 12

3.2.1 The Optical Setup 12

3.2.2 Deformation Model for the Physical Page Bending Process . . 12

3.2.3 Perspective Distortion Model 16

3.2.4 Nonlinear Illumination Model 18

3.2.5 Nonlinear Optical Point Spread Function 20

3.2.6 Simulation of the Perspective Distortion Model 20

3.3 Summary 22

Chapter 4: Synthetic Groundtruth 27
4.1 Groundtruth for Synthetic Data 27
4.2 Other Languages 31
4.3 Music Symbols 36
4.4 Line Drawings 36
4.5 Summary 36

Chapter 5: Document Registration 41
5.1 The groundtruth generation methodology: a closed loop approach . . 41
5.2 Geometric Transformations 43
5.3 Estimation of geometric transformation parameters 44
5.3.1 Similarity transformation 45
5.3.2 Affine transformation 46
5.3.3 Projective transformation 47
5.4 Validating geometric transformations 48
5.5 Dealing with nonlinearities 49
5.5.1 Piecewise bilinear interpolation functions 49
5.6 Finding corresponding points 51
5.6.1 Correspondence in documents with text 51
5.6.2 Correspondence in calibration documents 52
5.7 Summary 52

Chapter 6: Model Validation and Parameter Estimation 54
6.1 Statistical Problem Definition 54
6.2 Model Validation 55
6.3 Power Functions 60
6.4 Distance Functions, Outliers, and Robust Statistics 62
6.5 Estimation of the Degradation Model Parameters 64
6.6 Comparing Two models 65
6.7 Discussion 66

Chapter 7: Experimental Protocol and Results 67
7.1 Protocol for Synthetic vs. Synthetic 68

7.2 Protocol for Real vs. Real Experiment 69
7.3 Outliers and Distance Function Comparisons 69
7.4 Protocol for Calibration 75
7.5 Data Collection 78
7.6 Protocol for Generating Real Ground Truth 78
7.7 Protocol for Estimating Real Model Parameters 81
7.8 Protocol for Validating Real vs. Synthetic Degradations 81
7.9 Discussion 87
7.9.1 Comparing Two models 88
7.9.2 Approximate Tests 89

Chapter 8: Discussion **91**
8.1 The Scientific Method 91
8.1.1 Electrical Engineering and the Scientific Method 92
8.1.2 Computer Science and Scientific Method 93
8.1.3 Performance Evaluation and Characterization 94
8.2 Automatic Groundtruth: Making the Scientific Method Practical . . . 94
8.3 Degradation Models 95
8.4 Validating What? 96

Chapter 9: Conclusions **98**
9.1 Summary of Contributions 100
9.2 Future Research 102

Bibliography **104**

Appendix A: Null Distribution for Gaussian Populations **109**
A.1 Inter Cluster Mean Distance 109
A.2 Likelihood distance 110

Appendix B: Power Functions for Gaussian Populations **117**

Appendix C: Multivariate Hypothesis Testing for Gaussian Data **119**
C.1 Computer vision and multivariate testing 119

C.2 The hypotheses 120
C.3 Definitions 122
C.4 Test statistics, null distributions and power 123
C.4.1 Test 1: $\mu = \mu_0$ with known $\Sigma = \Sigma_1$ 123
C.4.2 Test 2: $\mu = \mu_0$ with unknown Σ 124
C.4.3 Test 3: $\Sigma = \Sigma_0$ with known $\mu = \mu_1$ 124
C.4.4 Test 4: $\Sigma = \Sigma_0$ with unknown μ 125
C.4.5 Test 5: $\Sigma = \Sigma_0$ and $\mu = \mu_0$ 125
C.5 Validating theory and software 126
C.5.1 Empirical null distributions 126
C.5.2 Kolmogorov-Smirnov tests 127

LIST OF FIGURES

3.1 Local document degradation model 11

3.2 Ideal document page typeset using LaTeX and IEEE Transaction's typesetting style. 13

3.3 Synthetically degraded version of the document in figure 3.2. 14

3.4 The setup while photocopying a thick, bound document. 15

3.5 The bending deformation of the document pages. 17

3.6 Perspective distortion. 19

3.7 If the image plane is not at focus then a point P maps to a disk of radius d. 21

3.8 This is the original binary image before undergoing perspective distortion. 23

3.9 This image is produced after a document undergoes perspective distortion. 24

3.10 This image is produced after undergoing a perspective distortion that is sharper than the perspective distortion shown in 3.9. 25

4.1 Degradation of a sample English document. 29

4.2 Groundtruth corresponding to the document shown in Figure 4.1. . . 30

4.3 Overlaying groundtruth on degraded documents 31

4.4 Degradation of Devanagari documents. 32

4.5 Groundtruth corresponding to the Devanagari document shown in Figure 4.4. 33

4.6 Degradation of Arabic documents. 34

4.7 Groundtruth corresponding to the Arabic document shown in Figure 4.6. 35

4.8 Degradation of music documents. 37

4.9 Groundtruth corresponding to the music document shown in Figure 4.8. 38

4.10 Degradation of linedrawings. 39

6.1 Permutation test for Gaussian distributions. 57
6.2 Nonparametric test when the two samples are from arbitrary distributions. 58
6.3 This figure shows the permutation procedure for computing the null distributions. 59
6.4 Power function. 61
6.5 Distance functions. 63

7.1 Local document degradation model 70
7.2 Power plots for the local document degradation model 71
7.3 Power functions for newarest neighbor distance. 72
7.4 Power functions for median nearest neighbor distance. 73
7.5 Power functions for 10% trimmed mean nearest neighbor distance. . . 74
7.6 Calibration images 76
7.7 A scatter plot of the error vectors computed between real calibration points and projected ideal calibration points. 77
7.8 Ground truth for real documents. 79
7.9 A submimage of a FAXed document with the ground truth overlaid. . 80
7.10 A submimage of a Hindi document in Devanagri script with the ground truth overlaid. 80
7.11 Estimation of degradation model. 82
7.12 The objective value as a function of the structuring element size, k. Rest of the parameters are fixed at the optimal solution. 83
7.13 The objective value as a function of the structuring element size. . . . 84
7.14 The objective value as a function of α_0. 85
7.15 The objective function as a function of the decay constant α. 86

A.1 Two sample hypothesis testing for Gaussian samples. 116

C.1 Test 1 distributions. 128
C.2 Test 2 distributions. 129
C.3 Test 3 distributions. 130
C.4 Test 4 distributions. 131

C.5 Test 5 distributions. 132

LIST OF TABLES

C.1 Kolmogorov-Smirnov test results for empirical null distributions of the five tests. 133

ACKNOWLEDGMENTS

I would like to thank Professor Robert M. Haralick, my doctoral advisor, for guiding me on my Ph.D. problem and for funding me throughout my stay. Very few researchers in the vision community have the strong statistical approach that he has been pushing forth and it has been a great privilege working with him. I would also like to thank Dr. Henry Baird (Bell Labs), Professor David Madigan (Statistics), Professor Ihsin Phillips (U. Seattle, CS), Professor Linda Shapiro (CS) and Professor Les Altas (EE) for agreeing to serve on my Ph.D. committee.

Professors David Madigan and Werner Stuetzle from the Statistics department introduced me to non-parametric statistics and permutation tests, which forms a central part of the degradation model validation and estimation methodology. Dr. Henry Baird invited me to work on the validation problem at Bell Labs, Murray Hill, during summer of 1994, and the collaboration and inciteful comments resulted in many interesting research directions.

I would like to thank all the colleagues in the document understanding group for stimulating discussions, and specially to Jaekyu Ha and Su Chen for sharing code. Professor Ihsin Phillips and Dr. Bhabutosh Chanda for comments on my work and for taking care of our project details and ensuring that our project deadlines were met.

I would also like to thank Dr. Byron Dom, Wayne Niblack, Dave Steele, from IBM Almaden, Dr. John Palmer from Psychology Dept., for many discussions and collaborations on various papers. The list won't be complete without thanking colleagues in the lab, and in particular with whom I shared many seminar and Cafe Latte sessions: Ken Thornton, Desika Nadadur, Xufei Liu, Anand Bedekar, Changkyu Lee, Bharath Modayur and Mauro&Wendy Costa.

Thanks to friends Sinan Karasu and Linda Knapp for the *Hindu Soma*; Octavia Camps for the Cafe Au Lait, technical collaboration and proof reading my dissertation; and brothers Manas and Rajesh and their families for the support. Also thanks to Dad for the constant encouragement, and for letting me play with his Ph.D. students, white mice, and ultra centrifuges when I was a kid; and Mom for the ever loving care and support.

DEDICATION

To my parents, Professor Madhu Sudan Kanungo and Mrs. Sarat Kanungo.

Chapter 1

INTRODUCTION

A few years ago the Defense Advanced Research Projects Agency (DARPA) convened a workshop [DAR92] to identify the problems that were stalling progress in document understanding research. The workshop had equal participation from the academia, the industry, and the government. At the conclusion of the meeting, the consensus was that a major impediment to progress in all subareas of document understanding research is the lack of performance evaluation methods. The absence of benchmarks had resulted in algorithm developers reporting accuracy of their Optical Character Recognition (OCR) systems at near-perfect levels ($> 99\%$). However, since the OCR systems are evaluated on different data sets, a 98% accuracy level of one OCR system is different from a 98% accuracy level of another OCR system. Thus it is difficult for the users to compare different OCR systems on a common quantitative basis. In addition, since the data sets used for evaluation are not representative samples of the population of degraded documents we encounter in real life, the reported accuracies of the OCR system again lose their meaning.

The current method of evaluating OCR algorithms is via *confusion matrices.* Row coordinates of confusion matrices represent the true identities of the characters while the column coordinates represent the identity reported by the OCR algorithm. The method is straight forward. Isolated, degraded characters are presented to the OCR system one at a time, and the system reports the hypothesized identity of the characters. The matrix entry representing the true identity and the corresponding reported identity is incremented. Once the data set is exhausted, the entries of the matrix are divided by the total number of characters used, to get the estimated error probabilities. This method of evaluation has many drawbacks. First, it does not take into account the context in which the characters appear. Thus degradations due to joining of characters, which are prevalent in real documents, are not represented by the data set. Second, a data-entry person has to isolate the characters and hand-enter the correct identities. This process is expensive, laborious and prone to human errors. Because of these drawbacks, we do not find databases with a large number of isolated, real characters.

The issue of context can be resolved by scanning real documents, hand-entering the corresponding groundtruth ASCII strings representing the document, and comparing the groundtruth string against the string reported by the OCR algorithm. This method is again very expensive, laborious, and prone to errors. Despite these problems, we have created various document databases with groundtruth. The errors in the groundtruth were reduced by a process of cross-checking.

A second methodology for evaluation of OCR algorithms is by synthetically degrading ideal documents. Here one first uses a wordprocessor to create an ideal document, in any language and formatted in any style or font. A bitmap version of these documents is created and then degraded using a computer model of the real degradation process. This method has many advantages. First, since the ideal document is created using a wordprocessor, the groundtruth information associated with each character – location, identity, font type, etc. – is known without error. Second, the word processor can be used to reformat the documents (two columns, one column, various font types, sizes, etc.) to study the sensitivity of the OCR algorithm to these variables. Third, since the degradation model is under our control, we can create documents with varying levels of degradations and study how and where the OCR algorithm breaks down. Fourth, sample size is not a problem at all – any number of degraded samples can be created since all that needs to be done is to simulate another set of characters. Fifth, given the original formatted documents, the groundtruth information is available free, in contrast to the groundtruth generation method mentioned earlier. In addition, there is no dearth of formatted documents – we create such documents daily, and so do academic journal publishers. Sixth, the model itself can be used for creating noise removal algorithms.

The main drawbacks with the above methodology are that (i) it relies heavily on the simulation model being correct, that is, it assumes that the simulation model mimics reality closely, and that (ii) the current document understanding community is skeptical of using synthetic data. It is thus imperative that we validate the degradation model against real data. Only then the simulations can be used *in place of* real data. If the degradation model is not validated, results on the synthetically degraded documents should be used with caution. They are still useful since they give *some* indication about the performance of the OCR algorithm.

A variant of this methodology allows us to gather real groundtruth at no expense. Instead of degrading the document using a simulation model, we can print the document and then scan it back. Now we have a real document but the original groundtruth position information is incorrect since the document has undergone a spatial transformation. However, if the spatial transformation can be estimated, then the groundtruth associated with the real data can be computed by transforming the groundtruth associated with the ideal document through the estimated spatial mapping function. The key problem here is to estimate the spatial transform that registers the ideal document image to the real one. In fact, as we shall see later in this dissertation, in order to validate the degradation model, we require both real and ideal groundtruth.

1.1 Contributions

The main contributions that are presented in this thesis are:

1. A model for the local degradations that are introduced while printing, photocopying and scanning a document. The model lends itself easily for generating synthetically degraded documents. It is parameterized and thus allows us to create documents with varying levels of degradations.

2. A physical model that accounts for the perspective and illumination distortions that occur while photocopying or scanning a thick, bound book. This model is also parametrized and allows us to synthetically generate distorted document images.

3. A methodology for degradation model parameter estimation. Given a sample of real images, the nonparametric estimation procedure finds the parameter values that make the simulated samples closest to the real samples. Thus, from the user point of view, a person having a small sample of real images can create a large sample by first estimating the parameters of the model and then synthetically generating a large data set.

4. A methodology for degradation model validation. Given a sample of real documents, and a sample of synthetic documents, this nonparametric hypothesis testing procedure tests the null hypothesis whether or not the two populations come from the same underlying distribution.

5. A methodology based on the power function that allows to optimize the validation procedure. The validation procedure has variables such as certain distance functions. This power function procedure allows us to select the distance function that makes the validation procedure more powerful (in a statistical sense).

6. An automatic groundtruth generation procedure for real documents. Given an ideal document and the corresponding groundtruth, the algorithm can generate the groundtruth for any printed, photocopied and scanned version of the ideal document image. The procedure was used to groundtruth over thirty real documents having 62000 characters. This method also works for documents in other languages. The availability of such data set allows the evaluation of OCR systems at a symbol level. This was was unthinkable until now.

7. All the software and the groundtruth data sets will be made available to researchers on a CD-ROM.

1.2 Overview

In chapter 2 we survey the related literature in the areas of degradation models, document registration, model validation, and statistical hypothesis testing and discuss the shortcomings of the current literature.

In chapter 3 we describe first describe a document degradation model for the local distortions that occur while printing, photocopying and scanning documents. Then we describe a physical model for the perspective and illumination distortions that occur while photocopying or scanning a thick bound book. The methodology described in chapter 3 is independent of the language in which the document is written.

In chapter 4 we describe a methodology for automatically generating groundtruth for synthetically generated documents. This methodology is not restricted to English documents. In fact, the methodology is used for generating groundtruth for Arabic, Devanagari, and Music documents. We show that the same method can also be used to generate groundtruth for engineering linedrawings document images.

To generate groundtruth for real data, we must register the ideal document to a real one generated by printing and subsequently scanning the ideal document. This registration algorithm is described in chapter 5. We show that the registration does not amount to simple translation and rotation of the ideal document. Nonlinearities in the imaging systems have to be overcome before one can achieve a registration of documents to high accuracy levels.

The validation methodology we have adopted compares degraded characters obtained from the real world by printing and scanning documents, to the synthetically degraded characters that result from the use of a degradation model. In chapter 6 we use synthetic and real data sets to validate our degradation model. In the same chapter we also provide methods for estimating the parameters of the model for a given real data set, and methods of selecting various algorithm parameters such as distance functions.

In chapter 7 we give experimental results for the validation and estimation experiments, and in chapter 8 we give our conclusions. .

Chapter 2

RELATED LITERATURE

The main areas that our work touches upon are document degradation models, document registration, groundtruthing, statistical hypothesis testing, and model validation. Here we review the relevant literature in these areas.

2.1 Degradation Models

The earliest work on document degradation models is that of Baird [Bai90, Bai93, Bai92]. What follows is a summary of his model. The input to the model is an ideal bilevel image, derived from artwork purchased from typeface manufacturers, and described at a spatial sampling rate much higher than the typical scanner (output) sampling rate. When the model is simulated, the parameters take effect in this order: the input image is rotated, scaled, and translated; then the output resolution and per–pixel jitter (random distribution of sensor centers) determine the locations of the centers of the output pixel sensors; for each pixel sensor a blurring kernel is applied, giving an analog intensity value; per–pixel sensitivity noise is added; finally, each pixel's intensity is thresholded. The output image is bilevel, at the output spatial sampling rate.

Unfortunately, the degradation model is not validated. Furthermore, the paper advocates the use of isolated, synthetically degraded characters. Thus the degradation due to merging of neighboring characters is not reflected in their model. Furthermore, the occurrence probabilities of individual characters in real-world text are not reflected in when isolated character experiments are conducted.

In contrast, our document degradation model, which is described in Chapter 3, advocates the use of complete documents for generating synthetically degraded characters. It thus takes into account the degradations arising due to merging of characters, the occurrence probabilities of individual characters, and the variability in the layout structure of the documents. The pixel degradations themselves are based on a local morphological model, which models the final spatial characteristics of the degradation process rather than the underlying physical process.

2.2 Document Registration

Extensive work on document registration has been reported in the literature. However, most of this literature pertains to the problem where a fixed ideal form has to be registered to a scanned, hand-filled form. The general idea is to extract the information filled by a human in the various fields of the form. One procedure is to introduce special registration marks on the document and then match the ideal registration marks to the ones detected on the scanned document image. Others extract features from the scanned forms and match them to the features in the ideal form [DR93, CF90]. Unfortunately we cannot use this body of work since there are no universal landmarks that appear in each type of document.

2.3 Statistical Model Validation

In the statistics literature, model validation is called 'hypothesis testing.' Many text books that have a good discussion on hypothesis testing procedures, for example [CB90] and [Arn90] give a good treatment of parametric statistics. In Appendix C we give an overview of multivariate hypothesis testing for Gaussian data. Although parametric statistics can answer many questions, in some situations modeling a population via parametric functions is not possible. In such cases one has to use nonparametric tests. Two textbooks that discuss nonparametric techniques in great detail are [Goo94] and [ET93]. However, most hypothesis testing methods reported in the statistical literature assumes that the data is finite dimensional, continuous, and have a known distribution such as Gaussian. In our validation problem, the data consists of scanned characters, which are binary matrices of varying dimensions. Thus, all the standard hypothesis testing techniques cannot be directly used for validating any degradation model.

2.4 Document Degradation Model Validation

To the best of our knowledge, the only other work on validation of degradation models is that of Nagy and Lopresti [Nag94, LLT94, LLT96]. They are of the opinion that a degradation model is valid if the OCR confusion matrices resulting from synthetically degraded documents are similar to the OCR confusion matrices produced from real documents. Unfortunately, this methodology validates the model-OCR combination and not the model itself. Thus, if the OCR system automatically scales documents, their validation process will not detect any difference between the real documents and the synthetically degraded documents even if the degradation process scaled the document. Furthermore, although they treat the OCR as a black box, the OCR algorithm itself has many parameters that can greatly influence the decision of the validation procedure. Another drawback of their approach is that their method does not lend itself naturally to comparison with other validation methods.

Our validation method on the other hand reduces the problem of model validation to a nonparametric statistical hypothesis testing problem, which is a well studied and accepted method in statistics. In addition, we do not use big OCR systems for the validation procedure, but simple distance functions between characters. Although the validation process now becomes a function of these distance functions, it is much simpler than OCR black boxes. Finally, we provide a technique for comparing our validation method with other validation methods. This comparison procedure is based on 'power functions,' which again are standard statistical devices for comparing hypothesis testing procedures.

Chapter 3

DOCUMENT DEGRADATION MODELS

Printing, photocopying and scanning processes degrade the image quality of a document. In this chapter we describe two document degradation models. First we describe a degradation model for local distortions that are introduced during the printing, photocopying and scanning processes. Then we describe a model for the perspective and illumimination distortions that get introduced when we photocopy or scan thick bound books.

3.1 A Local Document Degradation Model

In this section we present a model that accounts for (i) pixel inversion (from foreground to background and vice-versa) that occurs independently at each pixel due to light intensity fluctuations, sensitivity of the sensors, and the thresholding level, and (ii) blurring that occurs due to the point-spread function of the scanner optical system.

The degradation model has six parameters: $\Theta = (\eta, \alpha_0, \alpha, \beta_0, \beta, k)^t$. These parameters are used to degrade an ideal binary image as follows.

1. Compute the distance d of each pixel from the character boundary.

2. Flip each foreground pixel with a probability

$$p(0|1, d, \alpha_0, \alpha) = \alpha_0 e^{-\alpha d^2} + \eta.$$

3. Flip each background pixel with a probability

$$p(1|0, d, \beta_0, \beta) = \beta_0 e^{-\beta d^2} + \eta.$$

4. Perform a morphological closing operation with a disk structuring element of diameter k.

Thus, we model the pixel-flipping probability of a pixel as a function of its distance from the nearest boundary pixel. The foreground and background 4-neighbor distance can be computed using any distance transform algorithm (see [Bor86]). The parameters α_0 and β_0 are the initial values for the exponentials. The decay speed of the exponentials is controlled by the parameters α and β. The parameter η is the constant probability of flipping for all pixels. Finally, the last parameter k, which is the size of the disk used in the morphological closing operation, accounts for the correlation introduced by the point-spread function of the optical system.

Software for simulating noisy documents using the above degradation model is available from University of Washington English Document Database I and the model itself has appeared in the literature [KHP94, KHP93]. The application of the various steps of our model is shown in figure 3.1.

3.1.1 Implementation

The noise-free documents are typeset using the LaTeX formatting system [Lam86, Knu88]. The ASCII files containing the text and the LaTeX typesetting information are then converted into a device independent format (DVI) using LaTeX. A software program called DVI2TIFF – which is a modified version of a DVI file previewer called XDVI [V+90] – is run to produce one bit/pixel binary images in TIFF format from the DVI files. Besides producing the binary images of the documents, DVI2TIFF also produces the groundtruth information regarding each character on the document image. Examples of the groundtruth information are given in the next chapter.

The local document degradation model itself is another software program called DDM. This program takes as input an ideal binary document image in TIFF format, and a file containing the degradation model parameter values, and produces the binary degraded images in TIFF format.

Both programs – DVI2TIFF and DDM – are implemented in the C language and have been tested on SUN and IBM machines running the UNIX operating system. The software is available on the UW CD-ROM-1 [HP+].

3.1.2 Degrading Complete Pages

Since the input to the degradation software can be any LaTeX formatted ASCII file, the same text can be formatted in various styles (single column, multiple column,

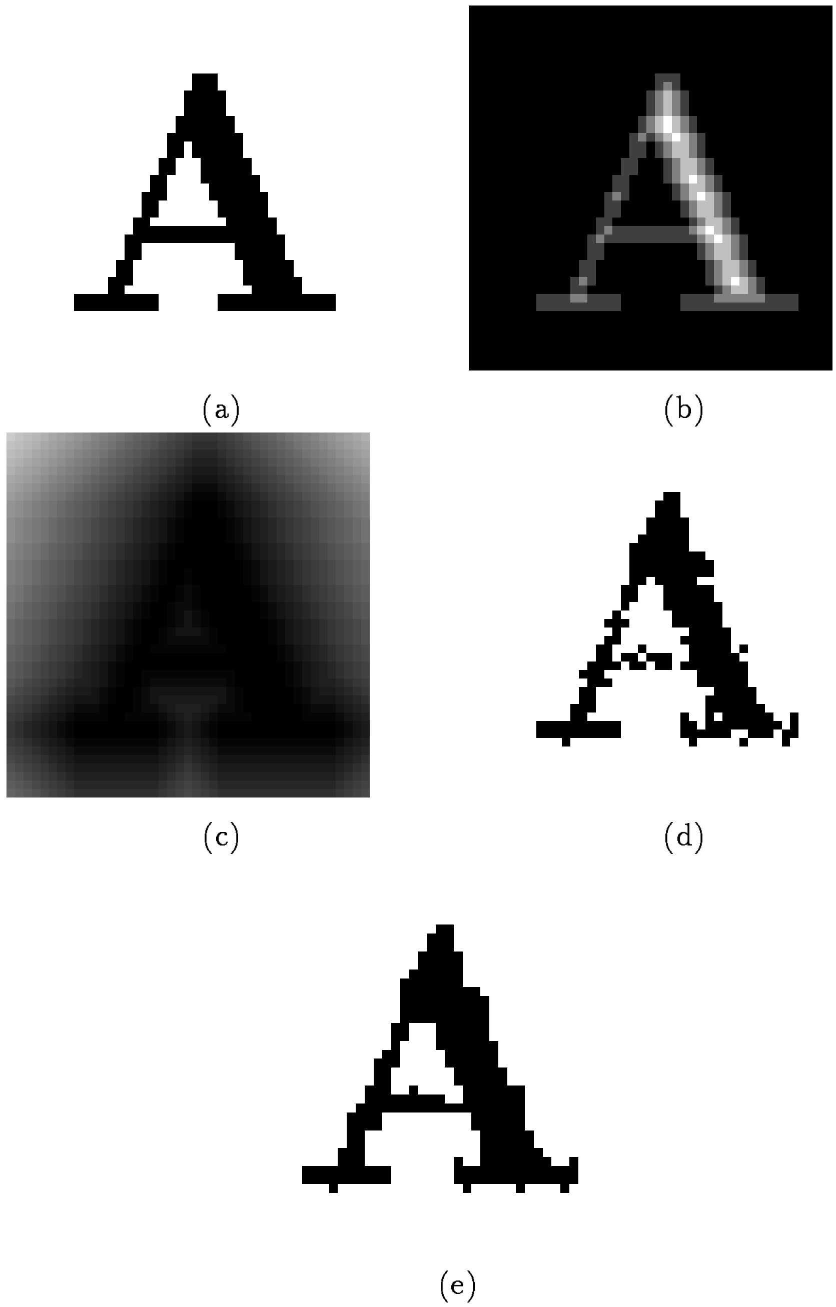


Figure 3.1: Local document degradation model: (a) Ideal noise-free character; (b) Distance transform of the foreground; (c) Distance transform of the background; (d) Result of the random pixel-flipping process. The probability of a pixel flipping is: $P(0|d,\beta,f) = P(1|d,\alpha,b) = \alpha_0 e^{-\alpha d^2}$ here $\alpha = \beta = 2$, $\alpha_0 = \beta_0 = 1$; (e) morphological closing of result in (d) by a 2×2 binary structuring element.

report, book, etc.), font types (Roman, Helvetica, etc), and font sizes (9pt, 10pt, 12pt, etc.). Thus the performance of any character recognition system can be studied by providing as input the same (or different) text formatted in various styles with varied but controlled degradation.

We now show examples where we degrade complete document pages using our degradation model. In figure 3.2, we show an ideal document formatted in LaTeX using the IEEE Transaction journal's typesetting style. In figure 3.3 we show a degraded version of the document in figure 3.2.

In the next section we describe a model for the distortions that occur while photocopying a think, bound book. The model accounts for the physical deformation of the document page, the perspective distortion that occurs because of the bending, the nonlinear intensity variations due to change in the surface-normal direction, and the nonlinear optical point-spread function.

3.2 A Global Page Degradation Model

In this section we model the perspective distortion that occurs while photocopying or scanning thick, bound documents. Perspective distortion is modeled by studying the underlying perspective geometry of the optical system of photocopiers and scanners. An illumination model is proposed to account for the nonlinear intensity change occurring across a page in a perspective-distorted document.

3.2.1 The Optical Setup

A typical setup for scanners and photocopiers is shown in the figure 3.4. In the figure we have shown a book that is to be photocopied. The page to be photocopied is not flat on the document glass since the book is tightly bound and the 'spine' of the book is thick. We model four sources of degradation in the following sections.

3.2.2 Deformation Model for the Physical Page Bending Process

First the page itself undergoes a physical deformation where the document page goes through a 'bending' process near the 'spine' of a thick, bound document. The page is no longer a flat surface on the document glass but a curved surface bending away from the glass near the spine of the book. We model this curved portion of the document

This Is A Sample File Using The 'IEEEtran.sty', To Help You Estimate Your Page Count And Facilitate Input-Processing Of Your Compuscript

ERB, Woody, Pheff, Bont, Tranman, IP, Dalton, Christine and OOZ

Abstract— **The theoretical analysis and derivation of artificial neural systems consist essentially of manipulating symbolic mathematical objects according to certain mathematical and biological knowledge. A simple observation has been made that this work can be done more efficiently with computer assistance by using and extending methods and systems of symbolic computation. In this paper, after presenting the mathematical characteristics of neural systems and a brief review on Liapunov stability theory, we present some features and capabilities of existing systems and our extension for manipulating objects occurring in the analysis of neural systems. Then, some strategies and a toolkit developed in MACSYMA for computer aided analysis and derivation are described. A concrete example is given to demonstrate the derivation of a hybrid neural system, i.e. a system which in its learning rule combines elements of supervised and unsupervised learning. The future work and directions on this topic are indicated.**

Keywords— **CA system, computer aided analysis and derivation, Liapunov function, neural system, symbolic computation.**

I. Introduction

SINCE the early 1940s a large number of artificial neural systems have been proposed by neural scientists. The dynamical behavior of these systems may be mathematically described by sets of coupled equations like differential equations for formal neurons with graded response. The investigation of essential features of neural systems such as stability and adaptation depends strongly upon the state of the mathematical theory to be applied and on a concrete and efficient analysis of dynamical equations. Unlike abstract theoretical research in which the mathematical objects adopted are frequently assumed to be of certain canonical form, the neurodynamics is usually complicated due to various biological facts which should be taken account of to a degree as large as possible. Consequently, this makes the analysis and derivation very complex, sometimes to an extent which is beyond human capacity, and the traditional methods and tools of mathematics are not always sufficient. It is therefore proposed in [19] to use and extend the methods and software systems of symbolic computation for handling, analyzing and constructing neurodynamics and its related objects. The present paper is the continuation of our work in this direction. The attempt is to demonstrate how symbolic computation can be applied to aid the analysis and derivation of neural systems.

This would be where the author affiliation would be, together with the IEEE log Number, like so.
The research of D. Wang is supported by a grant from SIEMENS AG Munich. He is with the Research Institute for Symbolic Computation, Johannes Kepler University, A-4040 Linz, Austria.
B. Schürmann is with the Corporate Research and Development, ZFE IS INF 2, Siemens AG, 8000 München 83, Germany.

In contrast to the approximative character of numerical calculations, symbolic computation treats objects with semantics like functions, formulae and programs. A variety of software systems for performing symbolic computation have been developed for research and applications in natural and technical sciences. However, the existing systems cannot be directly used for the analysis and derivation of neural systems as the operations on the occurring objects, particularly those involving an unspecified number of arguments like indefinite summations, have not yet been taken into account. To achieve our goal, some rules for differentiating and integrating indefinite summations with respect to indexed variables were proposed [20]. A toolkit has been designed and implemented in MACSYMA for manipulating these objects occurring in the analysis and derivation of neural systems [21].

In the next section, we introduce the general method and techniques for the stability analysis of artificial neural systems. The role of symbolic computation for representing and manipulating the objects concerning neural systems is discussed in Section III. In Section IV we present some strategies for using computer algebra (CA) systems and their extension to analyse the stability of neural systems and to derive novel stable systems. A brief description of a toolkit developed in MACSYMA is also provided. A concrete example is given in Section V to illustrate the derivation of a hybrid model by our toolkit. Section VI contains a discussion on future developments. The paper is closed with a brief summary.

II. Stability Analysis of Neural Systems

Consider artificial neural systems which are described by coupled systems of differential equations of the form

$$\dot{x} = F(x, w, K) \quad (1)$$

and

$$\dot{w} = G(x, w, K) \quad (2)$$

where $x = (x_1(t), ..., x_n(t))$ is the activation state vector, $w = (w_{ij}(t))$ is the weight matrix of dimension $n \times n$, n is the number of nodes and K is an external time-independent pattern vector. Such systems of differential equations which describe the neural model will occasionally be named *neurodynamics*.

Once a neural model is proposed, its main features are represented by its dynamic behavior. The adaptability of

1

Figure 3.2: Ideal document page typeset using $\LaTeX$ and IEEE Transaction's typesetting style.

This Is A Sample File Using The 'IEEEtran.sty', To Help You Estimate Your Page Count And Facilitate Input-Processing Of Your Compuscript

ERB, Woody, Pheff, Bont, Tranman, IP, Dalton, Christine and OOZ

Abstract—The theoretical analysis and derivation of artificial neural systems consist essentially of manipulating symbolic mathematical objects according to certain mathematical and biological knowledge. A simple observation has been made that this work can be done more efficiently with computer assistance by using and extending methods and systems of symbolic computation. In this paper, after presenting the mathematical characteristics of neural systems and a brief review on Liapunov stability theory, we present some features and capabilities of existing systems and our extension for manipulating objects occuring in the analysis of neural systems. Then, some strategies and a toolkit developed in MACSYMA for computer aided analysis and derivation are described. A concrete example is given to demonstrate the derivation of a hybrid neural system, i.e. a system which in its learning rule combines elements of supervised and unsupervised learning. The future work and directions on this topic are indicated.

Keywords—CA system, computer aided analysis and derivation, Liapunov function, neural system, symbolic computation.

I. Introduction

SINCE the early 1940s a large number of artificial neural systems have been proposed by neural scientists. The dynamical behavior of these systems may be mathematically described by sets of coupled equations like differential equations for formal neurons with graded response. The investigation of essential features of neural systems such as stability and adaptation depends strongly upon the state of the mathematical theory to be applied and on a concrete and efficient analysis of dynamical equations. Unlike abstract theoretical research in which the mathematical objects adopted are frequently assumed to be of certain canonical form, the neurodynamics is usually complicated due to various biological facts which should be taken account of to a degree as large as possible. Consequently, this makes the analysis and derivation very complex, sometimes to an extent which is beyond human capacity, and the traditional methods and tools of mathematics are not always sufficient. It is therefore proposed in [19] to use and extend the methods and software systems of symbolic computation for handling, analyzing and constructing neurodynamics and its related objects. The present paper is the continuation of our work in this direction. The attempt is to demonstrate how symbolic computation can be applied to aid the analysis and derivation of neural systems.

This would be where the author affiliation would be, together with the IEEE log Number, like so.
The research of D. Wang is supported by a grant from SIEMENS AG Munich. He is with the Research Institute for Symbolic Computation, Johannes Kepler University, A-4040 Linz, Austria.
B. Schürmann is with the Corporate Research and Development, ZFE IS INF 2, Siemens AG, 8000 München 83, Germany.

In contrast to the approximative character of numerical calculations, symbolic computation treats objects with semantics like functions, formulae and programs. A variety of software systems for performing symbolic computation have been developed for research and applications in natural and technical sciences. However, the existing systems cannot be directly used for the analysis and derivation of neural systems as the operations on the occurring objects, particularly those involving an unspecified number of arguments like indefinite summations, have not yet been taken into account. To achieve our goal, some rules for differentiating and integrating indefinite summations with respect to indexed variables were proposed [20]. A toolkit has been designed and implemented in MACSYMA for manipulating these objects occurring in the analysis and derivation of neural systems [21].

In the next section, we introduce the general method and techniques for the stability analysis of artificial neural systems. The role of symbolic computation for representing and manipulating the objects concerning neural systems is discussed in Section III. In Section IV we present some strategies for using computer algebra (CA) systems and their extension to analyse the stability of neural systems and to derive novel stable systems. A brief description of a toolkit developed in MACSYMA is also provided. A concrete example is given in Section V to illustrate the derivation of a hybrid model by our toolkit. Section VI contains a discussion on future developments. The paper is closed with a brief summary.

II. Stability Analysis of Neural Systems

Consider artificial neural systems which are described by coupled systems of differential equations of the form

$$\dot{x} = F(x, w, K) \tag{1}$$

and

$$\dot{w} = G(x, w, K) \tag{2}$$

where $x = (x_1(t), ..., x_n(t))$ is the activation state vector, $w = (w_{ij}(t))$ is the weight matrix of dimension $n \times n$, n is the number of nodes and K is an external time-independent pattern vector. Such systems of differential equations which describe the neural model will occasionally be named *neurodynamics*.

Once a neural model is proposed, its main features are represented by its dynamic behavior. The adaptability of

1

Figure 3.3: Synthetically degraded version of the document in figure 3.2.

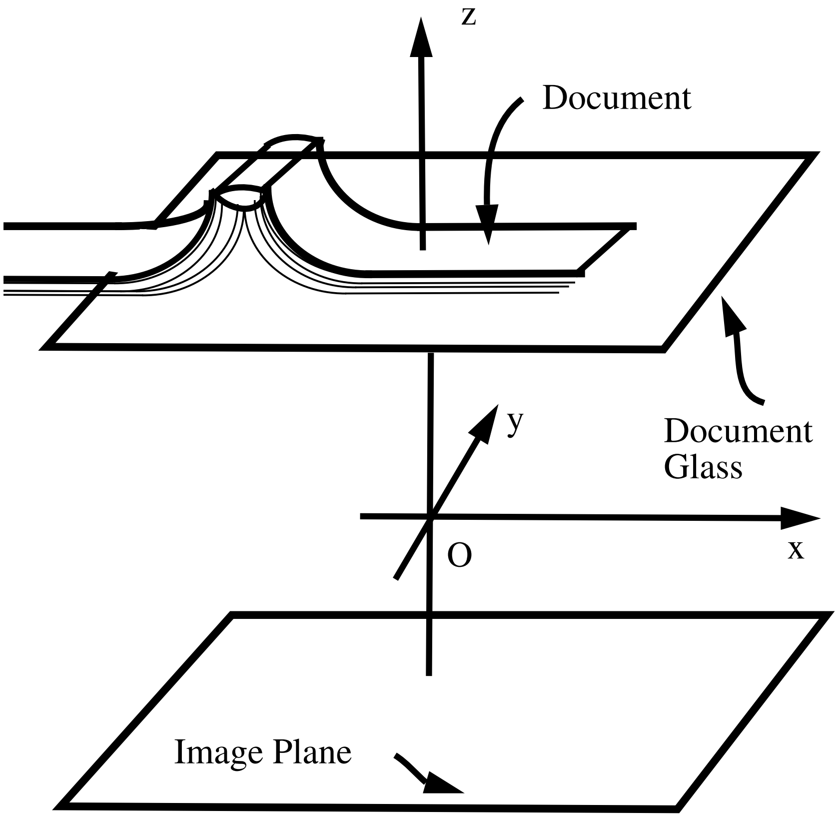


Figure 3.4: The setup while photocopying a thick, bound document. The center of perspectivity is at O, which is also the origin of the coordinate frame.

page as a circular arc segment along the x axis and assume that there is no such deformation along the y axis. The global rotation and translation can be modeled in another stage. Figure 3.5 illustrates this deformation phenomenon.

Let $A = (x_a, y_a, f)'$, $B = (x_b, y_b, f)'$. Furthermore, let ρ be the radius of the deformation circle and let the bent segment subtend an angle θ at the center of the circle D. Let the point A map to the point $A' = (x_{a'}, y_{a'}, z_{a'})'$ after deformation. Then the coordinates of A' are given by

$$x_{a'} = x_a + \rho(\theta - \sin\theta) \tag{3.1}$$

$$y_{a'} = y_a \tag{3.2}$$

$$z_{a'} = f + \rho(1 - \cos\theta) \tag{3.3}$$

Let the point $P = (x_p, y_p, f)'$, be such that $x_a \leq x_p \leq x_b$. and let P map to the point $P' = (x_{p'}, y_{p'}, z_{p'})'$ after deformation. Let the angle subtended by the arc $P'C$ at the center D be ϕ where

$$\phi = \frac{x_a + \rho\theta - x_p}{\rho} = \theta - \left(\frac{x_p - x_a}{\rho}\right). \tag{3.4}$$

Now the coordinates of P' can be calculated as given below

$$x_{p'} = x_p + \rho(\phi - \sin\phi) \tag{3.5}$$

$$y_{p'} = y_p \tag{3.6}$$

$$z_{p'} = f + \rho(1 - \cos\phi) \tag{3.7}$$

Note that for points P in the original document with $x_p > x_b$, we have no deformation and hence $P' = P$.

3.2.3 Perspective Distortion Model

The bending deformation is followed by a perspective distortion where the point P' on the document maps to the point P'' on the image. See figure 3.6. Let the focal length of the optical system be f and let the center of perspectivity, O, be at the origin. Assume that the image plane is at the focal plane at $-f$. Let $P'' = (x_{p''}, y_{p''}, z_{p''})'$ be the perspective projection of the point P' on the document page. The coordinates of

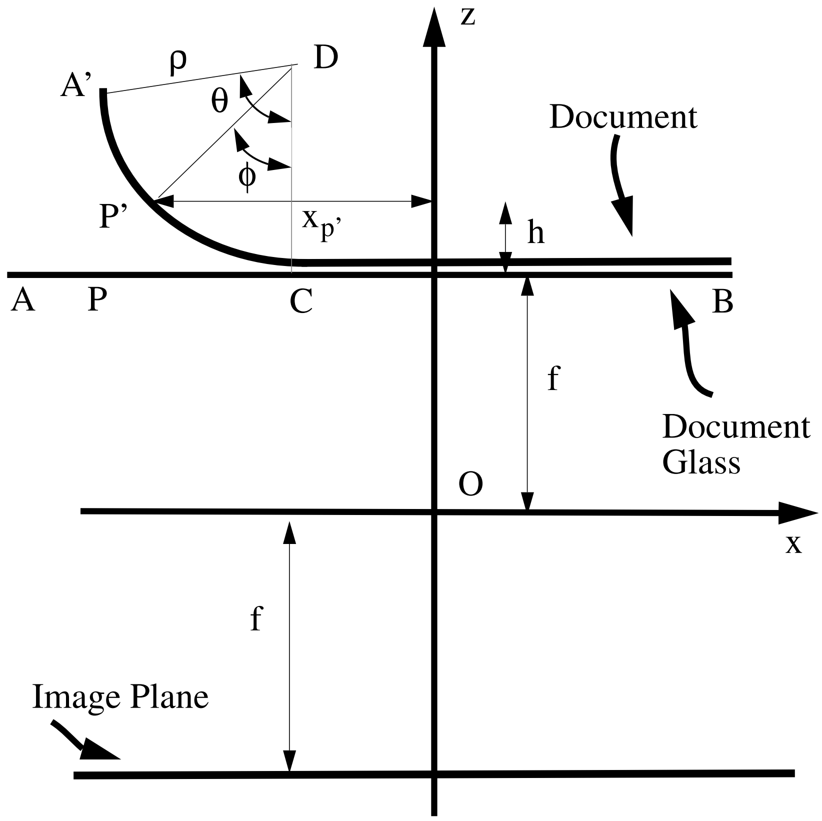


Figure 3.5: The bending deformation of the document pages. The side view of figure 1 while looking in the positive y direction is shown. The points A',B', and C' on the document page would have been at the points A, B, and C on the document glass if the page had not been curved. The curve $A'P'C$ is modeled as a circular arc segment that subtends an angle θ at the center, D, of the circle, which has a radius ρ. Here $h = \rho(1 - \cos\theta)$ and $x_{p'} = x_p + \rho(\phi - \sin\phi)$ where $\phi = \theta - (x_p - x_a)/\rho$. Rest of the page from C to B along the x axis is not deformed. It is assumed that the page does not undergo any deformation in the y direction either.

P'' are given by the following equations [HS92, Hor86]

$$\begin{aligned}
x_{p''} &= -f\left(\frac{x_{p'}}{f+\rho(1-\cos\phi)}\right) && (3.8)\\
&= -f\left(\frac{x_p+\rho(\phi-\sin\phi)}{f+\rho(1-\cos\phi)}\right) && (3.9)\\
y_{p''} &= -f\left(\frac{y_{p'}}{f+\rho(1-\cos\phi)}\right) && (3.10)\\
&= -f\left(\frac{y_p}{f+\rho(1-\cos\phi)}\right) && (3.11)\\
z_{p''} &= -f && (3.12)
\end{aligned}$$

Note that for points P in the original document with $x_p > x_b$, we have no bending or perspective deformation and hence $-P'' = P' = P$.

3.2.4 Nonlinear Illumination Model

Since the document page is no longer flat but a curved surface, the illumination on the document is not constant. The illumination at a point P' on the document pages is inversely proportional to the distance of point P' from the light source L. The light source L moves below the document glass from one end to the other. Let the distance between the document glass and the light source L be l_0. See figure 3.6. At the places where the page is curved the distance between the light source and the document pages is $l = l_0 + \rho(1 - \cos\phi)$ where ϕ is the angle arc $P'B$ subtends at B. Note ϕ is also the angle between the normal at P' and the negative z direction. We model the illumination as a diffuse lighting model. Thus the intensity of light is proportional to the cosine of the angle ϕ. Furthermore, after reflection, the diffuse model assumes the intensity of light is the same in all directions [HS92, Hor86]. Let I_0 be the intensity at a point where the document is not curved, i.e., the distance between the light and the point under consideration is l_0. Thus

$$I_0 \propto \frac{1}{l_0^2} \qquad (3.13)$$

Next, the intensity at $I_{p'}$ a point on the curved part is proportional to $\cos\phi$ and inversely proportional to $(l_0 + \rho(1 - \cos\phi))^2$. Thus

$$I_{p'} \propto \frac{\cos\phi}{(l_0+\rho(1-\cos\phi))^2} \qquad (3.14)$$

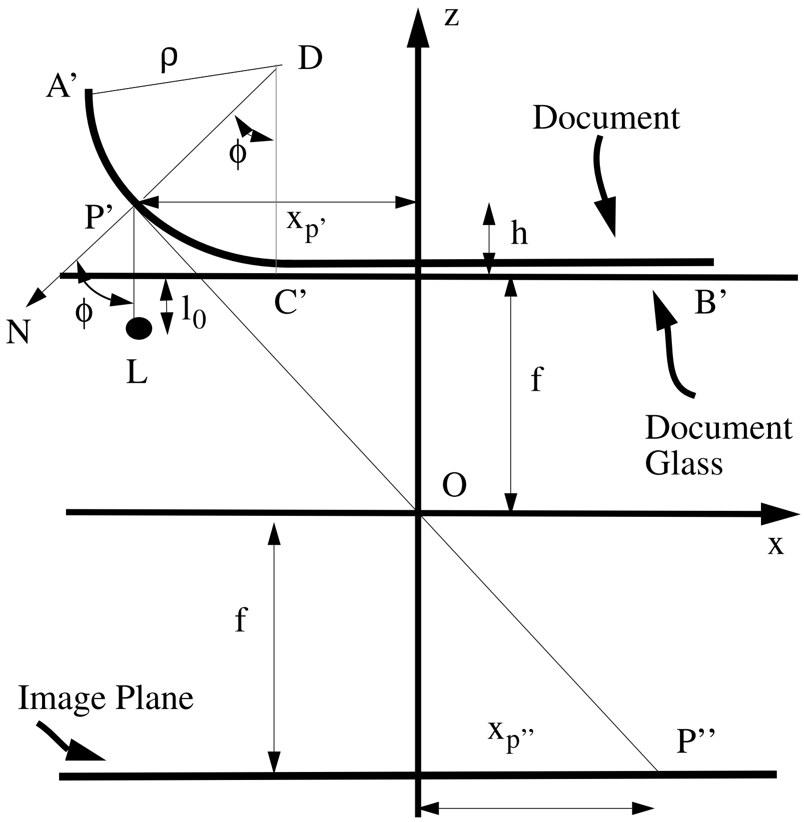


Figure 3.6: Perspective distortion. The point P' on the bent document page projects to the point P'' on the image plane. The coordinates of P'' are given as $x_{p''} = -f \cdot x_{p'}/(f+h)$, $y_{p''} = -f \cdot y_{p'}/(f+h)$, and $z_{p''} = -f$, where $h = \rho(1-\cos\phi)$ and $\phi = \theta - (x_p - x_a)/\rho$.

Thus taking a ratio of the above two equations we have

$$I_{p'} = I_0 \left(\frac{l_0}{l_0 + \rho(1 - \cos\phi)} \right)^2 \tag{3.15}$$

Under the assumption of diffuse lighting we have $I_{p'} = I_{p''}$

3.2.5 Nonlinear Optical Point Spread Function

Let us assume that a point P is on the focal plane, Then, if the image plane is not at the focus, the image P' of the point P will be blurred. In fact, the image of a point geometrically is a disk if the image plane is not in focus [SG88, Pen88, Hor86, HS92]. See figure 3.7. If Δ is the diameter of the lens, and h is the distance of the image plane from the focal plane, then the diameter of the disk is given by

$$d = \Delta \left(\frac{h}{f} \right) \ . \tag{3.16}$$

But due to optical irregularities, in reality we do not get a disk as the image but blurred version of a disk. In fact this blurred disk can be modeled as a Gaussian with a standard deviation $\sigma = k \cdot d$, where k is a camera constant.

Notice that in our case, the distance of a point on the document page is in focus only if the document page is in on the document glass (the focal plane). The curved region, in particular is not in focus since the points in that region are different distances from the focal plane. Thus the amount of blurring that a point goes through is different for the points on the curved segment.

Algorithmically, after performing the bending transformation, perspective distortion, and nonlinear illumination, another stage is necessary where the image is convolved with a space-varying Gaussian kernel. The kernel has a standard deviation σ given by

$$\sigma = k \cdot \rho(1 - \cos\phi) \ . \tag{3.17}$$

in the curved regions and constant σ_0. else where.

3.2.6 Simulation of the Perspective Distortion Model

In this section we show some simulation results of the model discussed thus far. The original nondistorted image is shown in figure 3.8. The dimensions of the image are

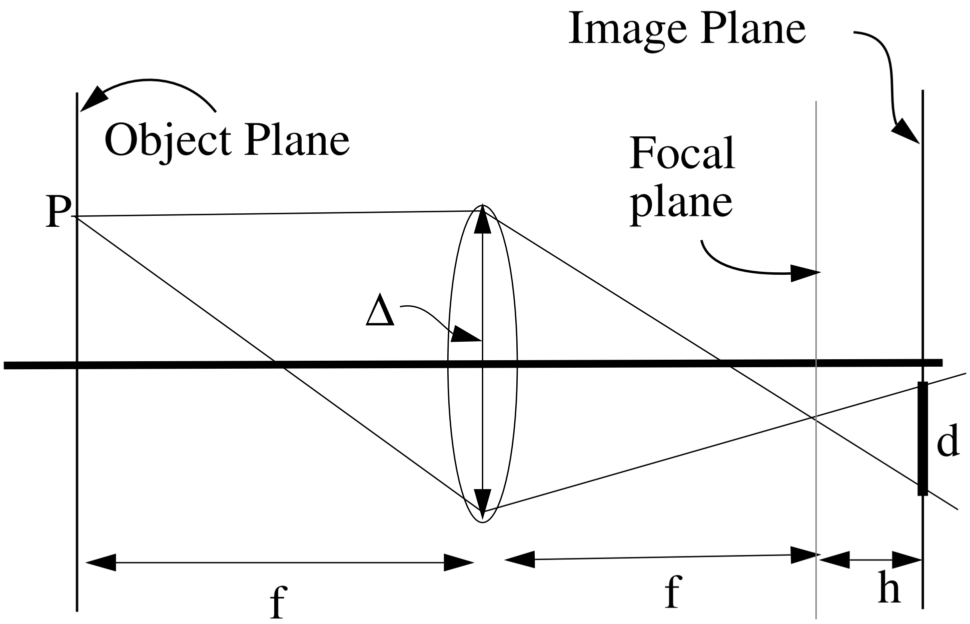


Figure 3.7: This figure illustrates the fact that if the image plane is not at focus then a point P maps to a disk of radius d. If the diameter of the lens is Δ and the focal length is f, the disk has a diameter $d = \Delta \cdot (h/f)$. Note in the real world the disk becomes blurred and can be approximated by a Gaussian. See text for more details.

201×201. The convolution kernel size used was 5×5. Two perspective deformations are shown in figures 3.9 and 3.10. The parameters, in units of pixels, used for generating figure 3.9 are:

$$\rho = 152.87 \tag{3.18}$$
$$\theta = 30^o \tag{3.19}$$
$$f = 80 \tag{3.20}$$
$$\Delta = 20 \tag{3.21}$$
$$k = 8 \tag{3.22}$$
$$l_0 = 10 \tag{3.23}$$

The parameters used for generating figure 3.10 are:

$$\rho = 95.54 \tag{3.24}$$
$$\theta = 30^o \tag{3.25}$$
$$f = 50 \tag{3.26}$$
$$\Delta = 50 \tag{3.27}$$
$$k = 1 \tag{3.28}$$
$$l_0 = 20 \tag{3.29}$$

3.3 Summary

We described a model for local distortions that occur during the printing, photocopying and scanning processes. The distortions are modeled in terms of distance transforms and the morphological closing operation. Under the model, the probability of a pixel flipping decreases exponentially as its distance from the boundary of a character increases. Furthermore, the flipping probability of a pixel is dependent on the pixels in a local neighborhood, which is modeled using via the morphological closing operation.

The implementation of the model allows us to degrade entire document pages and not just isolated characters. Since text in any language can be typeset using formatters like LaTeX, exisiting journals articles, book chapters, memos and letters can

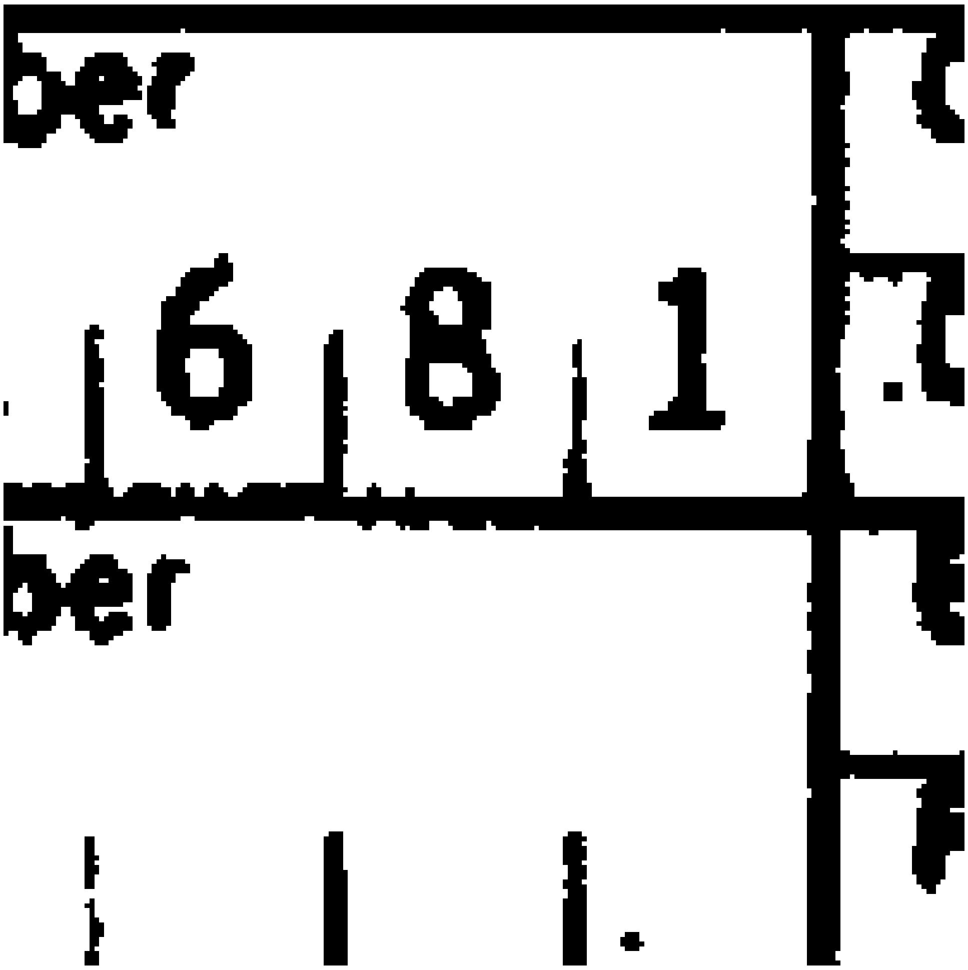


Figure 3.8: This is the original binary image before undergoing perspective distortion.

Figure 3.9: This image is produced after a document undergoes perspective distortion. Notice that the bend is very gradual and the intensity of light decreases as you go along the curved region. Furthermore, the text is no longer horizontal but curved inward. In addition, the blurring gets progressively worse toward the left edge of the image.

Figure 3.10: This image is produced after undergoing a perspective distortion that is sharper than the perspective distortion shown in 3.9. Furthermore, the intensity variation is not as pronounced as in the previous example.

be degraded synthetically and used for evaluating OCR algorithms – new documents need not be manually printed, photocopied and scanned specifically for evaluating algorithms. This reduces lot of manual overhead. In addition, since the model is parametric, degradations of varying levels and types can be introduced by simply changing the parameter values. Furthermore, the probability of occurrence of each characters is automatically set correctly if the same types of documents are used for training and testing. In addition, the use of entire document pages allows us to introduce degradations due to touching characters; and the joint probabilites of characters occuring in a particular sequence are represented correctly.

We also described a model for the perspective distortion occurring during the photocopying and scanning process. This model accounts for the physical deformation of the document page, perspective distortion, nonlinear intensity variations, and nonlinear optical point-spread function. We gave simulation results that showed that perspective model can account for the degradations.

Two related problems – parameter estimation and model validation – are addressed in chapter 6.

Chapter 4

SYNTHETIC GROUNDTRUTH

In the previous chapter we described a document degradation model that allowed us to generate synthetically degraded documents in any quantity and at various degradation levels. The system, in fact, gives us more than just degraded documents. It allows us to generate groundtruth corresponding to these degraded documents. In this chapter we describe the groundtruth information provided by our methodology for synthetically degraded documents. In the next chapter we show how this can be generated for real scanned document images as well.

4.1 Groundtruth for Synthetic Data

Groundtruth information is essential for evaluating any system that senses a real environment, not just document understanding systems. By *groundtruth* we mean the correct information about the scene that the vision system is trying to sense and interpret. For instance, in 3D-CAD vision, the groundtruth is the actual identity, position, and orientation of the CAD objects in the scene. In the case of document understanding, groundtruth means the correct location, size, font, and bounding box of the individual symbols on the document image. This information, of course, needs to be 100 percent accurate, otherwise the systems being evaluated will be penalized incorrectly.

Such groundtruth information is invaluable for performance evaluation of OCR algorithms:

Layout: We can keep the actual text of the document the same but change the layout. For example, we can switch from a single column format to a two column format. That will allow us to test whether or not the accuracy depends on the layout. Similarly, we can have tables and figures either inserted in the text, or have them on separate pages. By making such changes we can study if the OCR system can identify the figure and caption regions properly.

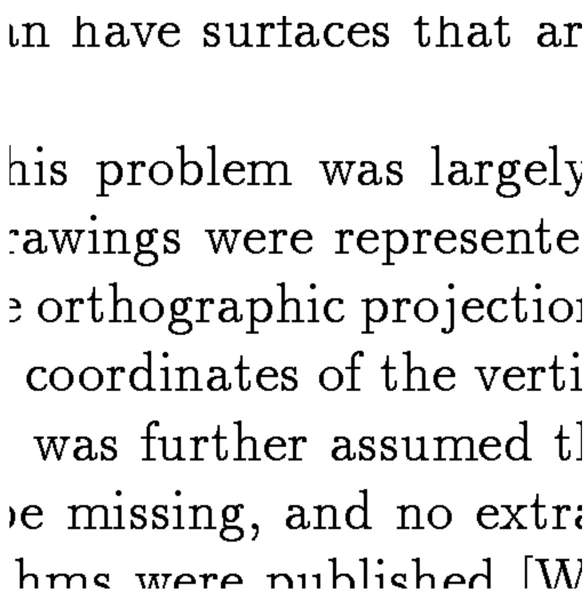
Style: Page numbers can be printed on top or bottom; the document may or may not have a runninghead; various indentation lengths can be varied; the columns can be justified or ragged. Thus by changing these variables, we can study how robust the OCR system is with respect to these style parameters.

Font: OCR systems can be very sensitive to the fonts used. Thus we can study the performance of the OCR algorithm by changing the various fonts (Helvetica, Times Roman, etc.) used in the documents, while keeping the text unchanged. Furthermore, OCR algorithms usually have a subsystem that identifies the font in a particular zone. Performance of such systems can be done if the groundtruth information about the fonts is available.

Size: Just as some OCR systems have subsystems that identify the font types, other OCR systems have subsystems that identify character size, which then is used by the recognition engine. Having the bounding box, location and identity, information associated with each symbol on the page will allow us to evaluate the performance of these subsystems.

Location and Identity: Finally, since the groundtruth contains the location and identity (e.g., which character, or math symbol) of each symbol on the page, we can use this information to evaluate the performance of the OCR system.

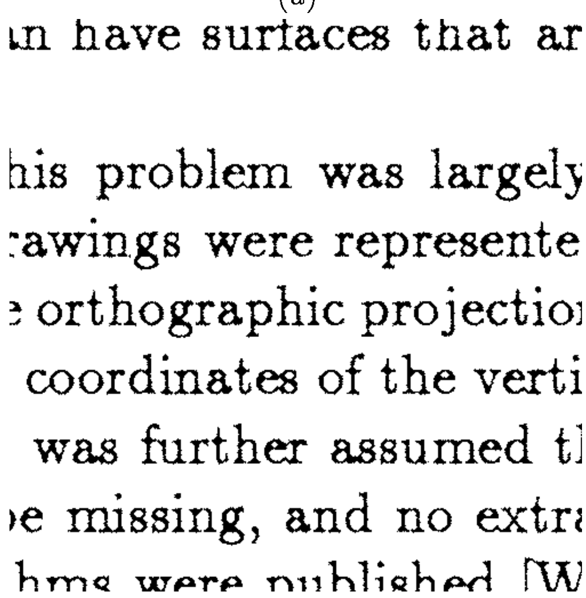
In Figure 4.1(a) we have a document formatted in a single column format. Part of the groundtruth corresponding to the document shown in Figure 4.1 is shown in Figure 4.2. Each row contains information regarding one symbol on a page. For instance, the first line provides us the information that a symbol is present at the location (469,570) in the document image, which has height of 46 pixels, a width of 28 pixels, is of 10 point Computer Modern Roman Bold font, and is the numeral '1.' In Figure 4.3 we have used the groundtruth file to find the location of all the letter 'e's of 12 point size in the document and have overlaid the information on the degraded document image.



ın have surfaces that ar

his problem was largely
:awings were represente
ɜ orthographic projectio
coordinates of the verti
was further assumed t]
›e missing, and no extrɛ
hms were published [W

(a)



ın have surfaces that ar

his problem was largely
:awings were represente
ɜ orthographic projectio
coordinates of the verti
was further assumed t]
›e missing, and no extrɛ
hms were published [W

(b)

Figure 4.1: (a) An ideal document image typeset using LaTeX. The document is in one column format, the font used is Computer Modern Roman and the font size is 12 point. A synthetically degraded version of the document is shown in (a). The degradation model parameter values used are $\alpha_0 = \beta_0 = 1.0$, $\alpha = \beta = 2.5$, and $k = 5$.

```
#
# Bounding box information for:
# Dvi file: survey.dvi ; page 3 of total 25
#
# Formats for Fonts, Rectangles, Lines and Points:
#
# Line: x1 y1 x2 y2
# Point: x y
# Rect: x y width height
# Font: x y width height font-name decimal-code ascii
#
# The origin is at top-left of the image
# The x axis corresponds to the columns and increases from left to right
# The y axis corresponds to the rows and increases from top to bottom.
#
#
#                       0,0
#                        +-----------> +x
#                        |
#                        |
#                        |
#                       \ /
#                        +y
#
Font: 469 570 28 46 cmbx10 49 1
Font: 589 567 26 49 cmbx10 73 I
Font: 621 584 41 32 cmbx10 110 n
Font: 664 570 24 46 cmbx10 116 t
Font: 696 584 29 32 cmbx10 114 r
```

Figure 4.2: Groundtruth corresponding to the document shown in Figure 4.1.

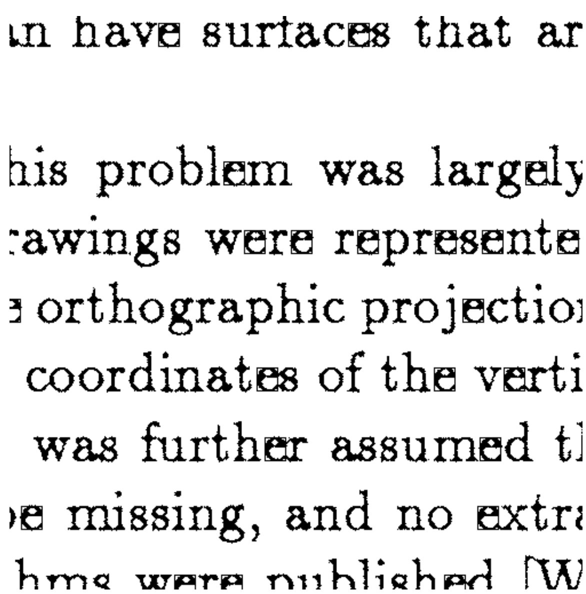


Figure 4.3: The groundtruth file corresponding to Figure 4.1 was used to find the location of all the letter 'e's of 12 point size in the document, and then the information was overlaid on the degraded image shown in Figure 4.1(a).

4.2 Other Languages

The degradation model and the groundtruth generation methodology are not restricted to the Roman script. Since the model can be applied to any binary image, text written in any language can be synthetically degraded. Furthermore, since most languages can be typeset using LaTeX, the corresponding groundtruth can be automatically created using our groundtruth generation software. We now show some examples where we synthetically degrade Devanagari and Arabic scripts, and produce the corresponding groundtruth. In Figure 4.4(a) we show an ideal image of a Hindi document written in Devanagari script. The corresponding degraded image is shown in Figure 4.4(b). Finally, in Figure 4.5 we show the groundtruth corresponding to the Hindi document. Similarly, in Figure 4.6(a) we show and ideal Arabic document. We degrade this ideal bitmap using our model and the resulting image is shown in Figure 4.6(b). The corresponding groundtruth is shown in Figure 4.7.

मोहन राकेश: मिस पाल

वह दूर से दिखायी देती आकृति मिस पाल ही हो सकती थी। फिर भी विश्वास करने के लिए मैंने अपना चश्मा ठीक किया। निःसंदेह, वह मिस पाल ही थी। यह तो खैर मुझे पता था कि वह उन दिनों कुल्लू में ही कहीं रहती है, पर इस तरह अचानक उससे भेंट हो जायेगी, यह नहीं सोचा था। और उसे सामने देखकर भी मुझे विश्वास नहीं हुआ कि वह स्थायी रूप से कुल्लू और मनाली के बीच उस छोटे-से गाँव में रहती होगी। जब वह दिल्ली से नौकरी छोड़कर आयी थी, तो लोगों ने उसके बारे में क्या-क्या नहीं सोचा था!

बस रायसन के डाकखाने के पास पहुंचकर रुक गयी। मिस पाल डाकखाने के बाहर खड़ी पोस्टमास्टर से कुछ बात कर रही थी। हाथ में वह एक थैला लिये थी। बस के रुकने पर न जाने किस बात के लिए पोस्टमास्टर को धन्यवाद देती हुई वह बस की तरफ मुड़ी। तभी मैं उतरकर उसके सामने पहुँच गया। एक आदमी के अचानक सामने आ जाने से मिस पाल थोड़ा अचकचा गयी, मगर मुझे पहचानते ही उसका चेहरा खुशी और उत्साह से खिल गया।

(a)

मोहन राकेश: मिस पाल

वह दूर से दिखायी देती आकृति मिस पाल ही हो सकती थी। फिर भी विश्वास करने के लिए मैंने अपना चश्मा ठीक किया। निःसंदेह, वह मिस पाल ही थी। यह तो खैर मुझे पता था कि वह उन दिनों कुल्लू में ही कहीं रहती है, पर इस तरह अचानक उससे भेंट हो जायेगी, यह नहीं सोचा था। और उसे सामने देखकर भी मुझे विश्वास नहीं हुआ कि वह स्थायी रूप से कुल्लू और मनाली के बीच उस छोटे-से गाँव में रहती होगी। जब वह दिल्ली से नौकरी छोड़कर आयी थी, तो लोगों ने उसके बारे में क्या-क्या नहीं सोचा था!

बस रायसन के डाकखाने के पास पहुंचकर रुक गयी। मिस पाल डाकखाने के बाहर खड़ी पोस्टमास्टर से कुछ बात कर रही थी। हाथ में वह एक थैला लिये थी। बस के रुकने पर न जाने किस बात के लिए पोस्टमास्टर को धन्यवाद देती हुई वह बस की तरफ मुड़ी। तभी मैं उतरकर उसके सामने पहुँच गया। एक आदमी के अचानक सामने आ जाने से मिस पाल थोड़ा अचकचा गयी, मगर मुझे पहचानते ही उसका चेहरा खुशी और उत्साह से खिल गया।

(b)

Figure 4.4: Degradation of Devanagari documents. (a) An ideal Devanagari document; (b) A degraded version of the Devanagari document shown in (a). The degradation model parameter values used were $\alpha_0 = \beta_0 = 1.0$, $\alpha = \beta = 2.5$, and $k = 5$.

```
# Font: x y width height font-name decimal-code ascii
#
# The origin is at top-left of the image
#                      0,0
#                       +-----------> +x
#                       |
#                       |
#                       |
#                      \ /
#                       +y
#
Font: 941 313 39 32 dvng10 109 m
Font: 961 299 35 46 dvng10 111 o
Font: 992 313 34 40 dvng10 104 h
Font: 1023 313 37 32 dvng10 110 n
Font: 1079 313 33 32 dvng10 114 r
Font: 1108 313 21 32 dvng10 65 A
Font: 1125 313 41 32 dvng10 107 k
Font: 1131 299 22 14 dvng10 3 Λ
Font: 1162 313 43 32 dvng10 102 f
Font: 1215 323 5 20 cmr10 58 :
Font: 1257 299 28 46 dvng10 69 E
Font: 1274 313 39 32 dvng10 109 m
Font: 1308 313 41 32 dvng10 115 s
Font: 1369 313 35 32 dvng10 112 p
Font: 1399 313 21 32 dvng10 65 A
Font: 1416 313 43 32 dvng10 108 l
```

Figure 4.5: Groundtruth corresponding to the Devanagari document shown in Figure 4.4.

نوادر

جحا وحميره العشرة

اشترى جحا عشرة حمير . فرح بها وساقها أمامه ، ثمّ ركب واحدا منها . وفي الطّريق عدّ حميره وهو راكب ، فوجدها تسعة . ثمّ نزل وعدّها فرآها عشرة فقال : أمشي وأكسب حمارا ، أفضل من أن أركب وأخسر حمارا .

الولد والطّبل

طلب ولد من أبيه أن يشتري له طبلا صغيرا . فرفض الوالد ، وقال له : يا بنيّ ، لو اشتريت لك طبلا فسوف تزعجنا بصوته .

قال الولد : لا تغضب يا أبي . لا أطبّل به ، إلّا وأنت نائم .

(a)

نوادر

جحا وحميره العشرة

اشترى جحا عشرة حمير . فرح بها وساقها أمامه ، ثمّ ركب واحدا منها . وفي الطّريق عدّ حميره وهو راكب ، فوجدها تسعة . ثمّ نزل وعدّها فرآها عشرة فقال : أمشي وأكسب حمارا ، أفضل من أن أركب وأخسر حمارا .

الولد والطّبل

طلب ولد من أبيه أن يشتري له طبلا صغيرا . فرفض الوالد ، وقال له : يا بنيّ ، لو اشتريت لك طبلا فسوف تزعجنا بصوته .

قال الولد : لا تغضب يا أبي . لا أطبّل به ، إلّا وأنت نائم .

(b)

Figure 4.6: Degradation of Arabic documents. (a) An ideal Arabic document; (b) A degraded version of the Arabic document shown in (a). The degradation model parameter values used were $\alpha_0 = \beta_0 = 1.0$, $\alpha = \beta = 2.5$, and $k = 5$.

```
# Font: x y width height font-name decimal-code ascii
#
# The origin is at top-left of the image
#                      0,0
#                       +-----------> +x
#                       |
#                       |
#                       |
#                      \ /
#                       +y
#
Font: 1220 601 24 27 nash14 80
Font: 1250 589 18 22 nash14 88
Font: 1275 570 6 41 nash14 64
Font: 1278 596 29 32 nash14 241
Font: 1309 583 7 6 nash14 9
Font: 1304 595 12 16 nash14 75
Font: 1090 649 13 6 nash14 16
Font: 1089 661 15 22 nash14 232
Font: 1097 677 26 23 nash14 81
Font: 1119 672 20 11 nash14 229
Font: 1134 650 20 33 nash14 134
Font: 1153 669 28 14 nash14 170
Font: 1178 645 10 38 nash14 203
Font: 1197 642 6 41 nash14 64
Font: 1231 661 15 22 nash14 232
Font: 1239 666 26 23 nash14 81
Font: 1263 661 17 22 nash14 30
Font: 1265 689 13 6 nash14 10
```

Figure 4.7: Groundtruth corresponding to the Arabic document shown in Figure 4.6.

4.3 Music Symbols

Music scores can also be formatted using LaTeX, and many packages are available for typesetting music. Thus, the same groundtruth generation methodology can be used for automatically creating groundtruth for synthetically degraded music documents. In Figure 4.8(a) we have a subimage of an ideal music document. In Figure 4.8(b) we have a synthetically degraded version of the document in Figure 4.8(a). The corresponding groundtruth is shown in Figure 4.9.

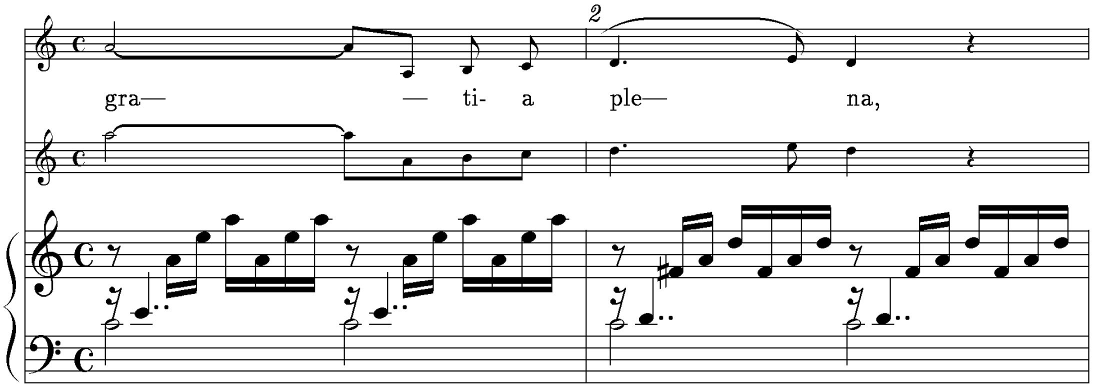
4.4 Line Drawings

Another subarea of document understanding research is that of linedrawing understanding. Here semantic information has to be extracted from scanned documents containing graphical symbols and other geometric entities that appear on the document page according to some accepted 2D syntax such as the ISO or ANSI standard [Ame82]. The goal of these linedrawing understanding systems is to convert all the paper drawings into CAD format so that the storage, manipulation, and extraction of information is easier and inexpensive. Various types of linedrawings are of interest: mechanical CAD drawings, wiring diagrams, electronic circuit diagrams, architectural drawings, etc.

Such drawings cannot be created using LaTeX. But many CAD modelers such as AutoCAD allow us to create various types of linedrawings. These drawings can then be stored symbolically using the IGES file format [SW86], which is an accepted industry-standard, or as a binary image. Thus, once again we can apply our methodology for generating degraded linedrawings and the corresponding groundtruth: degrade the linedrawing image using the degradation model and then use the information in the corresponding IGES file to create the groundtruth. Figure 4.10(a) shows a subimage of an ideal linedrawing document. Figure 4.10(b) shows a synthetically degraded version of the document in Figure 4.10(a).

4.5 Summary

In this chapter we saw that for various types of synthetically degraded documents we can automatically create the groundtruth information – information regarding



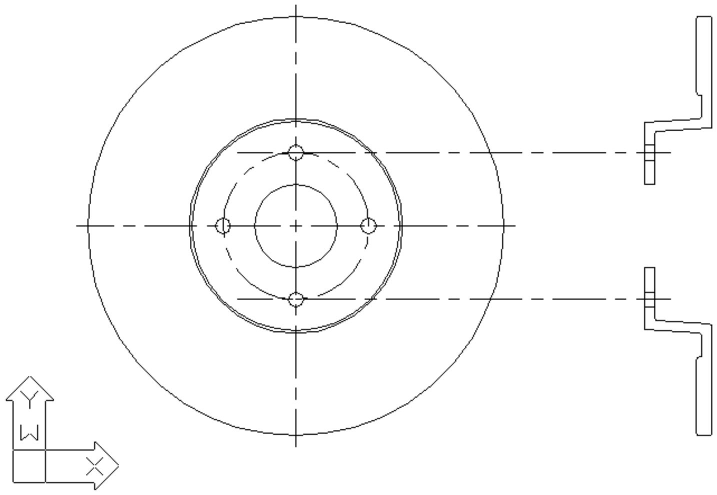
(a)

(b)

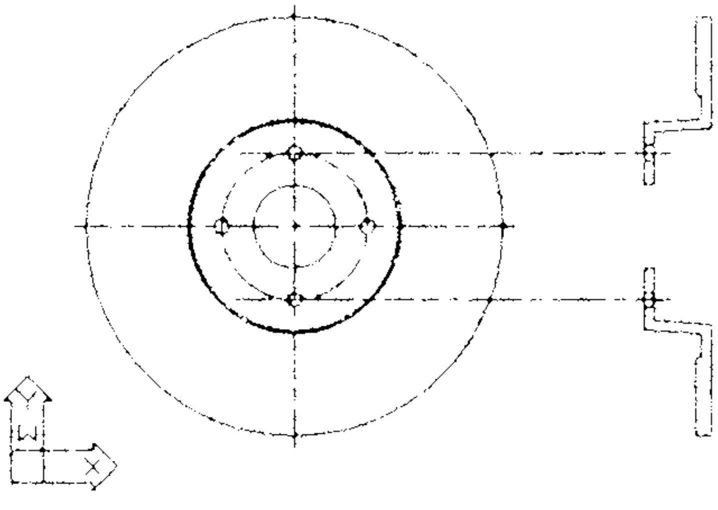
Figure 4.8: Degradation of music documents.(a) An ideal music document; (b) A degraded version of the music document shown in (a). The degradation model parameter values used were $\alpha_0 = \beta_0 = 1.0$, $\alpha = \beta = 2.5$, and $k = 5$.

```
#
# Rect: x y width height
# Font: x y width height font-name decimal-code ascii
#
#                       0,0
#                        +-----------> +x
#                        |
#                        |
#                        |
#                       \ /
#                        +y
#
Font: 256 816 27 266 musicbra 12
Rect: 293 507 1894 2
         ...
Rect: 293 454 2 631
Font: 301 999 56 69 musikn20 73
Font: 306 777 55 154 musikn20 71
Font: 313 430 36 100 musikn13 71
Font: 382 1022 33 43 musikn20 83
Font: 379 671 21 28 musikn13 83
         ...
Font: 437 570 20 31 cmr10 103 g
Font: 460 571 15 20 cmr10 114 r
Font: 479 571 20 20 cmr10 97 a
Font: 500 578 44 2 cmr10 124 |
Rect: 452 441 2 47
Font: 436 481 17 14 musikn13 34
         ...
```

Figure 4.9: Groundtruth corresponding to the music document shown in Figure 4.8.



(a)



(b)

Figure 4.10: Degradation of linedrawings. (a)An ideal linedrawing produced by AutoCAD; (b) A degraded version of linedrawing shown in (a).

the location, identity, size and font type – that is 100% accurate. This is possible since the typesetting languages and stystems store such information in order to create these documents. Thus, for synthetically degraded documents, one know the correct result that any OCR system should produce. No manual groundtruthing is required – the groundtruth inforation is produced at no cost and instantaneously. Thus the methodology described in this chapter and the last allow us to create large databases of synthetically degraded documents with 100% accurate groundtruth. This methodology thus paves way for conducting large controlled experiments on OCR systems that were earlier not possible.

Another important advantage of using our methodology is that we can generate synthetically degraded documents and the corresponding groundtruth for *any* language. We showed examples where we generated degraded Devanagri and Arabic texts with groundtruh. And for doing so, one did not have to know the language. In contrast, manual groundtruth collection would have required the knowledge of the language in which the text being groundtruh is written. The same method can be also used to generate synthetically degraded music documents and engineering drawings and the corresponding groudtruth in IGES format.

Chapter 5

DOCUMENT REGISTRATION

Collecting accurate groundtruth for characters in a real document is a difficult task that is not possible manually because (i) accuracy in delineating groundtruth character bounding boxes is not high enough, (ii) it is extremely laborious and time consuming and (iii) the manual labor required for this task is prohibitively expensive. Furthermore, in many cases of badly degraded documents, such as FAX document, it is not even possible to read the words, let alone groundtruth them.

In the previous chapter we described a method for generating 100% accurate groundtruth for synthetic documents. Unfortunately, all the information becomes useless once the document is printed and scanned, which makes the scanned document undergo a geometric transformation (translation, rotation, scale, etc.). However, if there is a way of estimating the transformation that the ideal document undergoes when it is printed and scanned, groundtruth for the real data can be easily computed by transforming the ideal groundtruth using the same geometric transformation.

In this chapter we show how we generate the character groundtruth for real documents. First we to generate the groundtruth for the ideal documents and then find a 2D-2D transformation that registers the ideal documents to the real documents. The estimated mapping that registers the ideal document image to the real document image is then used to transform the ideal groundtruth to get the groundtruth for the real documents. The groundtruth generated by this method, besides being directly useful for evaluating the performance of OCR systems, is crucial for validating document degradation models.

5.1 The groundtruth generation methodology: a closed loop approach

First, the documents are typeset using LaTeX. Next these documents are converted into binary bitmap images, which are our ideal noise-free documents. When the ideal bitmap is generated from the DVI files, the corresponding groundtruth (location, bounding box, font type and size, and identity of each character) is also generated.

The ideal document image is then printed and scanned. At this point, although the groundtruth for the ideal image is known, the groundtruth for the real scanned image is unknown. However, if the exact transformation that registers the ideal and degraded images were known, we could compute the groundtruth for the real image simply by transforming the bounding box coordinates of the ideal groundtruth by the same transformation.

Thus the groundtruth creation problem now reduces to finding an appropriate transformation that models the geometric distortions the document image undergoes when it is printed and then scanned. Whatever the functional form of the transformation, to estimate the parameters of the transformation we require corresponding feature points from the ideal and real images. Thus, a rough outline of the groundtruth generation method is:

1. Generate ideal document images and the associated character groundtruth.

2. Print the ideal documents and scan it back.

3. Find feature points $p_1, \ldots, p_n$ and $q_1, \ldots, q_n$ in the corresponding ideal and real document images.

4. Establish the correspondence between the points p_i and q_i.

5. Estimate the parameters of the transformation T that maps p_i to q_i.

6. Transform the ideal groundtruth information using the estimated transformation T.

The transformation T mentioned in the procedure above is a $2D$ to $2D$ mapping. That is $T : R^2 \rightarrow R^2$. Thus, if $(x, y) = T(u, v)$, where (u, v) is the ideal point and (x, y) is the scanned point, x in general may be a function of both u and v; and same is true regarding y.

Generation of the ideal document image and the corresponding groundtruth is achieved by our synthetic groundtruth generation software DVI2TIFF, which we described in Chapter 3. Given a transformation T, transforming the groundtruth information is trivial – all that needs to be done is transform the bounding box coordinates of the ideal groundtruth using the transformation T. Thus, there are two main problems: finding corresponding feature points in two document images, and finding the transformation T.

5.2 Geometric Transformations

Suppose we are given the coordinates of feature points p_i on an ideal document page, and the coordinates of the corresponding feature points q_i on the real document page. (How these feature points are extracted and matched is described in the next section.) The problem is to hypothesize a functional form for the transformation T, that maps the ideal feature points coordinates to the real point coordinates, and a corresponding noise model. To ensure that the transformation T is the same throughout the area of the document page, we choose the points p_i from all over the document page.

The possible candidates for the geometric transformation and pixel perturbation are similarity, affine, and projective transformations:

Similarity Transformation: This transformation is defined by the equation:

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{pmatrix} a & b \\ -b & a \end{pmatrix} \begin{pmatrix} u_i \\ v_i \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} + \begin{pmatrix} \eta_i \\ \psi_i \end{pmatrix} \tag{5.1}$$

where (u_i, v_i) is the ideal point and (x_i, y_i) is the transformed point.

In the above parameterization of the similarity transformation, a represents the product of a nonnegative isotropic scale and cosine of the rotation angle; b represents the product of the nonnegative scale and sine of the rotation angle; t_x and t_y represent the translation in x and y directions. This parametrization is linear and unconstrained in the parameters, unlike the parametrization in terms of scale, cosine and sine of rotation angle, and translation.

Affine Transformation: In this case we assume that the real image is an affine transformation of the ideal image. The affine transformation allows for rotation, translation, anisotropic scale, and shear. The functional form that maps the ideal point (u_i, v_i) onto the real point (x_i, y_i) is

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} u_i \\ v_i \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} + \begin{pmatrix} \eta_i \\ \psi_i \end{pmatrix}. \tag{5.2}$$

Projective Transformation: Here the assumption is that the real image is a perspective projection of an image on a plane onto another nonparallel plane. The functional form that maps the ideal point (u_i, v_i) onto the real point (x_i, y_i) is given below.

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} = \frac{1}{a_3 u_i + b_3 v_i + 1} \begin{pmatrix} a_1 u_i + b_1 v_i + c_1 \\ a_2 u_i + b_2 v_i + c_2 \end{pmatrix} + \begin{pmatrix} \eta_i \\ \psi_i \end{pmatrix} \tag{5.3}$$

After inspection it can be seen that the equations are linear and unconstrained in the eight parameters $a_1, b_1, c_1, a_2, b_2, c_2, a_3, b_3$. Discussion on the estimation of these parameters can be found in subsection 5.3. This parameterization accounts for rotation, translation and the center of perspectivity parameters.

The natural choice for noise is a Gaussian. That is, $(\eta, \psi)^t \sim N(0, \sigma^2 I)$. Furthermore, σ can be assumed to be known and a function of the image processing algorithm that is used to detect the feature points.

Each of these models can be used to fit the data. Nevertheless, the question is which model, if any, models the transformations correctly, and how does one go about proving the hypothesis quantitatively?

In the next section, we show how to estimate the parameters of the three models. In the section that follows we show how to statistically validate/invalidate the models.

5.3 Estimation of geometric transformation parameters

Note that all the three models are linear in the parameters. Each corresponding point provides two linear constraints on the parameters. Thus we need at least two corresponding points for estimating the similarity parameters, three corresponding points for affine, and four for projective. If we have more corresponding points than the minimum required, we can solve for the parameters of the transformation in a least squares sense, which also happens to be the maximum likelihood estimate of the parameters under the Gaussian noise model.

5.3.1 Similarity transformation

The similarity equations given in equation (5.1) can be rearranged into the following form:

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{pmatrix} 1 & 0 & u_i & v_i \\ 0 & 1 & v_i & -u_i \end{pmatrix} \cdot \begin{pmatrix} t_x \\ t_y \\ a \\ b \end{pmatrix} + \begin{pmatrix} \eta_i \\ \psi_i \end{pmatrix}, \tag{5.4}$$

where (u_i, v_i) is the ideal point and (x_i, y_i) is the transformed point. If there are n corresponding points, the above equation can be written as:

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & 0 & u_1 & v_1 \\ 1 & 0 & u_2 & v_2 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & u_n & v_n \\ 0 & 1 & v_1 & -u_1 \\ 0 & 1 & v_2 & -u_2 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & v_n & -u_n \end{bmatrix} \cdot \begin{pmatrix} t_x \\ t_y \\ a \\ b \end{pmatrix} + \begin{bmatrix} \eta_1 \\ \eta_2 \\ \vdots \\ \eta_n \\ \psi_1 \\ \psi_2 \\ \vdots \\ \psi_n \end{bmatrix}. \tag{5.5}$$

The above equation can be written in a compact matrix form as follows:

$$\mathbf{b} = \mathbf{A}\mathbf{p} + \mathbf{n} \tag{5.6}$$

where $\mathbf{b}$ is the $2n \times 1$ vector of x and y, $\mathbf{A}$ is the $2n \times 4$ form matrix, $\mathbf{p}$ is the 4×1 vector of unknown parameters, and $\mathbf{n}$ is the $2n \times 1$ vector of noise values. If the number of corresponding points n is two, we have four equations in four unknowns, and thus can solve for $\mathbf{p}$ uniquely by solving the system of equations:

$$\mathbf{b} = \mathbf{A}\hat{\mathbf{p}}. \tag{5.7}$$

However, if we have more correspondences, we can solve for $\mathbf{p}$ in a least squares sense.

$$\hat{\mathbf{p}} = \arg\min_{\mathbf{p}} \|\mathbf{A}\mathbf{p} - \mathbf{b}\|. \tag{5.8}$$

The least squares solution is obtained by solving the following linear system of equations:

$$\mathbf{A}^t\mathbf{b} = \mathbf{A}^t\mathbf{A}\hat{\mathbf{p}}. \tag{5.9}$$

The proof of the fact that the solutions of the two equations (5.8) and (5.9) is the same can be found in standard linear algebra texts such as [Str88]. Incidentally, the least squares solution is also the maximum likelihood estimate of $\mathbf{p}$ under the assumption that $\mathbf{n}$ is Gaussian distributed as $N(0, \sigma^2 I)$.

5.3.2 Affine transformation

The affine equations given in equation (5.2) can be rearranged into the following form:

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{pmatrix} u_i & v_i & 0 & 0 & 1 & 0 \\ 0 & 0 & u_i & v_i & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} a \\ b \\ c \\ d \\ e \\ f \end{pmatrix} + \begin{pmatrix} \eta_i \\ \psi_i \end{pmatrix}. \tag{5.10}$$

where (u_i, v_i) is the ideal point, and (x_i, y_i) is the transformed point. If there are n corresponding points, the above equation can be written as:

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} u_1 & v_1 & 0 & 0 & 1 & 0 \\ u_2 & v_2 & 0 & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_n & v_n & 0 & 0 & 1 & 0 \\ 0 & 0 & u_1 & v_1 & 0 & 1 \\ 0 & 0 & u_2 & v_2 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & u_n & v_n & 0 & 1 \end{bmatrix} \cdot \begin{pmatrix} a \\ b \\ c \\ d \\ e \\ f \end{pmatrix} + \begin{bmatrix} \eta_1 \\ \eta_2 \\ \vdots \\ \eta_n \\ \psi_1 \\ \psi_2 \\ \vdots \\ \psi_n \end{bmatrix}. \tag{5.11}$$

The above equation can be written in a compact matrix form as follows:

$$\mathbf{b} = \mathbf{A}\mathbf{p} + \mathbf{n} \tag{5.12}$$

where $\mathbf{b}$ is the $2n \times 1$ vector of x and y, $\mathbf{A}$ is the $2n \times 6$ form matrix, $\mathbf{p}$ is the 6×1 vector of unknown parameters, and $\mathbf{n}$ is the $2n \times 1$ vector of unknown noise values. If the number of corresponding points n is three, we have six equations in six unknowns, and thus can solve for $\mathbf{p}$ uniquely by solving the system of equations

$$\mathbf{b} = \mathbf{A}\hat{\mathbf{p}}. \tag{5.13}$$

However, if we have more correspondences, we can solve for $\mathbf{p}$ in a least squares sense:

$$\hat{\mathbf{p}} = \arg\min_{\mathbf{p}} ||\mathbf{A}\mathbf{p} - \mathbf{b}|| \, . \tag{5.14}$$

The least squares solution, which is also the maximum likelihood solution in this case, is obtained by solving the following linear system of equations:

$$\mathbf{A}^t\mathbf{b} = \mathbf{A}^t\mathbf{A}\hat{\mathbf{p}}. \tag{5.15}$$

5.3.3 Projective transformation

The projective transformation equations given in equation (5.3) can be rearranged in the following form.

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{pmatrix} u_i & v_i & 1 & 0 & 0 & 0 & -u_i x_i & -v_i x_i \\ 0 & 0 & 0 & u_i & v_i & 1 & -u_i y_i & -v_i y_i \end{pmatrix} \cdot \begin{pmatrix} a_1 \\ b_1 \\ c_1 \\ a_2 \\ b_2 \\ c_2 \\ a_3 \\ b_3 \end{pmatrix} + \begin{pmatrix} \eta_i \\ \psi_i \end{pmatrix}, \tag{5.16}$$

where (u_i, v_i) is the ideal point and (x_i, y_i) is the transformed point. If there are n corresponding points, the above equation can be written as:

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} u_1 & v_1 & 1 & 0 & 0 & 0 & -u_1x_1 & -v_1x_1 \\ u_2 & v_2 & 1 & 0 & 0 & 0 & -u_2x_2 & -v_2x_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_n & v_n & 1 & 0 & 0 & 0 & -u_nx_n & -v_nx_n \\ 0 & 0 & 0 & u_1 & v_1 & 1 & -u_1y_1 & -v_1y_1 \\ 0 & 0 & 0 & u_2 & v_2 & 1 & -u_2y_2 & -v_2y_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & u_n & v_n & 1 & -u_ny_n & -v_ny_n \end{bmatrix} \begin{pmatrix} a_1 \\ b_1 \\ c_1 \\ a_2 \\ b_2 \\ c_2 \\ a_3 \\ b_3 \end{pmatrix} + \begin{bmatrix} \eta_1 \\ \eta_2 \\ \vdots \\ \eta_n \\ \psi_1 \\ \psi_2 \\ \vdots \\ \psi_n \end{bmatrix} \tag{5.17}$$

The above equation can be written in a compact matrix form as follows.

$$\mathbf{b} = \mathbf{A}\mathbf{p} + \mathbf{n} \tag{5.18}$$

where $\mathbf{b}$ is the $2n \times 1$ vector of x and y, $\mathbf{A}$ is the $2n \times 8$ form matrix, $\mathbf{p}$ is the 8×1 vector of unknown parameters, and $\mathbf{n}$ is the $2n \times 1$ vector of noise values. If the number of corresponding points n is four, we have eight equations in eight unknowns, and thus can solve for $\mathbf{p}$ uniquely by solving the following system of equations:

$$\mathbf{b} = \mathbf{A}\hat{\mathbf{p}}. \tag{5.19}$$

However, if we have more correspondences, we can solve for for $\mathbf{p}$ in a least squares sense.

$$\hat{\mathbf{p}} = \arg\min_{\mathbf{p}} ||\mathbf{A}\mathbf{p} - \mathbf{b}||\,. \tag{5.20}$$

The least squares solution, which is also the maximum likelihood solution in this case, is obtained by solving the following linear system of equations:

$$\mathbf{A}^t\mathbf{b} = \mathbf{A}^t\mathbf{A}\hat{\mathbf{p}}. \tag{5.21}$$

5.4 *Validating geometric transformations*

Since the estimated parameters of the models are functions of real point coordinates, which are random variables, the estimated parameters are random variables. The distribution of estimated parameters can be derived in terms of the assumed distribution of the noise in the real point coordinates. To confirm that the geometric transformation model and the noise model are valid, we can test whether or not the theoretically derived distribution of the estimated parameter vector is the same as that computed empirically. If either the geometric transformation model or the noise model is incorrect, the test for equality of the empirically computed distribution and the theoretically derived distribution will not pass. Furthermore, instead of testing the distribution of the estimated parameters, we can test the distribution of the residual error, which in turn has a known distribution.

5.5 Dealing with nonlinearities

As we show in a later chapter on experimental results, none of the three geometric transformations model the transformation very accurately. That is, the real points are displaced from ideal transformed points in some nonlinear fashion. The mismatch seems to arise from nonlinearities in the optical system. Although we had first assumed that the nonlinearities are due to mechanical variations from scan to scan, it turned out that for our scanner this was not true. We proved this fact as follows. We scanned the same document multiple number of times without displacing the original. When the images were exclusive ORed, not much difference was found.

As in photogrammetry literature, we resorted to computing two interpolating functions in two variables that modeled the nonlinearities in the printer-scanner combination. In the next section we describe the details of the piecewise bilinear interpolation functions. A good discussion on image warping can be found in [Wol90].

5.5.1 Piecewise bilinear interpolation functions

Let the function $f : R^2 \rightarrow R^2$ be an image-to-image nonlinear function. We are given the ideal calibration points $p_1, \ldots, p_n$, and the corresponding observed points $q_1, \ldots, q_n$. That is, $q_i = f(p_i) + \eta$. The problem is to construct a piecewise bilinear function that approximates f in the sense that

$$\sum_{k=1}^{n} ||q_k - f(p_k)|| \tag{5.22}$$

is minimized.

The piecewise bilinear function is represented as follows. First, a grid of points $g_{i,j}$, with $i = 1, \ldots, l$ and $j = 1, \ldots, m$ on the first image are identified. The grid points are such that the y-coordinates of the points along a row of grid points are the same and the x-coordinates of points along a column of grid points is the same. That is, $y(g_{i,j}) = y(g_{i,k})$ for $j = 1, \ldots, m$, $k = 1, \ldots, l$. Furthermore, there is a natural ordering of the grid point coordinates: $x(g_{i,j}) < x(g_{i+1,j})$ and $y(g_{i,j}) < y(g_{i,j+1})$. Note that the number of grid points is much less than the number of calibration points: $l \times m < n$.

We represent the nonlinear function f by representing the transformation on the grid of points $g_{i,j}$. Let $g_{i,j} + \Delta g_{i,j}$ be the grid point after the function f transforms the grid point $g_{i,j}$. Let the point p lie within a grid cell whose four corner grid points are: $a = g_{i,j}, b = g_{i+1,j}, c = g_{i+1,j+1}, d = g_{i,j+1}$. The transformation of the point p is then approximated as follows. Let

$$\begin{aligned} t &= (x(p) - x(a))/(x(b) - x(a)), && (5.23) \\ s &= (y(p) - y(d))/(y(d) - y(a)). && (5.24) \end{aligned}$$

Then the point $q = f(p) + \eta$ after transformation is given by

$$q = p + (1-t)(1-s)\Delta a + t(1-s)\Delta b + ts\Delta c + (1-t)s\Delta d + \eta, \tag{5.25}$$

where $\Delta a = \Delta g_{i,j}$; and $\Delta b, \Delta c$, and Δd are defined similarly.

Let a_k, b_k, c_k, d_k be the corner points of the grid cell within which the point p_k lies, and let t_k and s_k be constants calculated using equations (5.23) and (5.24). Equation (5.22) can be stated as: Find Δa_k, Δb_k, Δc_k, Δd_k to minimize

$$\sum_{k=1}^{n} ||q_k - [p_k + (1-t_k)(1-s_k)\Delta a_k + t_k(1-s_k)\Delta b_k + t_k s_k \Delta c_k + (1-t_k)s_k\Delta d_k]|| \ . \tag{5.26}$$

In the above equation, out of the $n \times 4$ elements Δa_k, Δb_k, Δc_k, Δd_k, $k = 1, \ldots, n$, only $l \times m$ elements are unique. For example, Δc_9 and Δd_{20} both might represent the same grid point variation, $\Delta g_{4,5} : \Delta c_9 = \Delta d_{20} = \Delta g_{4,5}$. We can now give unique labels to the grid differences, setup a system of linear equations, and solve for the unique elements in a least squares sense.

5.6 Finding corresponding points

Corresponding points are required in two scenarios. The first scenario is where we are calibrating the scanner to find the geometric transformation model and the non-linearity model. In this case we create ideal binary images with patterns appropriate for calibration. In the second scenario, we have documents with text, figures and mathematics, and we need to identify features in the ideal image and the corresponding features in the real image. Both these situations are considered in the following two subsections.

5.6.1 Correspondence in documents with text

In a document image with text, figures and mathematics, there are no universal feature points in the interior of the document that are guaranteed to appear in each type of document. However, most documents have a rectangular text layout, whether they are in one-column format or in two-column format. We use the upper-left (UL), upper-right (UR), lower-right (LR), and lower-left (LL), corners of the text area as feature points.

The four feature points, $p_1, \ldots, p_4$, are detected on the ideal image as follows.

1. Compute the connected components in the image.

2. Compute the upper-left (a_i), upper-right (b_i), lower-right (c_i), and lower-left (d_i) corners of the bounding box of each connected component.

3. Find the four feature points using the following equations:

$$
\begin{aligned}
p_1 &= \arg\min_{a_i}(x(a_i) + y(a_i)), \\
p_2 &= \arg\max_{b_i}(x(b_i) - y(b_i)), \\
p_3 &= \arg\max_{c_i}(x(c_i) + y(c_i)), \\
p_4 &= \arg\min_{d_i}(x(d_i) - y(d_i)).
\end{aligned}
$$

   The above equations compute the upper-left (q_1), upper-right (q_2), lower-right (q_3), and lower-left (q_4).

The above algorithm is also used to compute the corresponding four feature points $q_1, \ldots, q_4$ on the real image. Since sometimes noise blobs can appear in a real image, we check to see that the bounding box sizes of the components are within a specified tolerance. A transformation T can be estimated using the corresponding points $p_1, \ldots, p_4$ and $q_1, \ldots, q_4$ by the methods described in section 5.3.

5.6.2 Correspondence in calibration documents

The geometric transformation T is independent of the content of the document image, and is a property of the printer-scanner combination, and the way the document is placed on the scanner bed. For the purposes of calibrating the printer-scanner combination we can eliminate the problems of feature extraction and finding the correspondences by simply using an ideal image, specially created for calibration, with features that can be accurately detected and matched. The ideal calibration page is made up of crosses whose dimensions and locations on the page are known. The intersection points of the vertical and horizontal lines of the crosses are the feature points. Detection of the intersection points in the real document image and in the ideal image is easily and reliably achieved using binary morphological image processing. Once the intersection points are detected, the bounding boxes of these feature points are computed by connected component analysis. An initial transformation is then computed using only four corresponding points that are detected and matched using the algorithm described in the previous subsection.

Given an estimated transform T, the points p_i are then transformed using this transformation. Next, for each transformed point $T(p_i)$ we find the real point q_j that is closest. This is done by a brute-force $O(n^2)$ algorithm. Since n is of the order of 1000, the computation required is of the order 10^6, which takes approximately three seconds on a Sparc 2.

5.7 Summary

In this chapter we presented a closed-loop method for producing character groundtruth for real document images. The method starts by generating ideal noise-free document images using a document typesetting software like LaTeX. These binary document images are printed, photocopied, and then scanned. Feature points are extracted from the ideal and the scanned document images, and their correspondences established. We showed that the similarity, affine and projective transformations alone cannot be used to represent the transformation between the ideal and the scanned documents. This fact was confirmed by using test images specially designed for calibration, and verifying that the statistical distribution of the registration error is not what the theory predicts. The local nonlinearities that exist can be accounted for by performing a local template match using the ideal character as the template, and searching a small neighborhood in the real image for the best match. The size of the local search neighborhood is decided by the calibration experiment. The calibration experiment gives us the maximum deviations that can occur between the ideal feature points after they have been transformed using the estimated transformation and the feature points on the scanned image.

Chapter 6

MODEL VALIDATION AND PARAMETER ESTIMATION

6.1 Statistical Problem Definition

In this section we formulate the degradation model parameter estimation problem and the model validation problem as statistical problems. Although degradation of the document is over the entire page, the degradation process itself is local. That is, degradation in one region does not influence the degradation process in another sufficiently far away region. More precisely, the degradation at a pixel is influenced only by pixels within a local neighborhood. Thus, one way to characterize the degradation process is to study the degradation of local patterns. Since the most common patterns that occur on a document page are characters, we statistically characterize the degradation of individual characters on the page and use this characterization to estimate the parameters of a degradation model that produces similar degradations.

Assume that a scanned character is represented by a 30×30 matrix with zeros or ones. This matrix can be represented as 1000×1 vector x. ($30 \times 30 \approx 1000$.) Let B be the space of $D = 1000$ dimensional binary vectors, that is, $B = \{0, 1\}^D$. Now, let $x_1, x_2, \ldots, x_N \in B$ be independent and identically distributed D-dimensional vectors representing instances of degraded characters produced from the same class ω. That is, each x_i is a degraded character that is produced from the same ideal pattern ω (say the ideal character 'e') and the same degradation process. In our case D is large, typically on the order of 1000, Thus, the number of possible x_i's is 2^{1000}, which is approximately equal to 10^{300} – a dauntingly large number. The available sample size, N, is typically on the order of 1000. Thus, the sample x_i occupy the space B extremely sparsely, which implies that the density function cannot be estimated reliably from the sample. This fact prohibits us from performing any standard statistical test based on density estimates.

The two problems we need to address are:

Model Validation: Suppose we are given a set of *real* degraded instances $x_1, \ldots, x_N$ $\in B$ of the pattern ω and another set of *synthetic* degraded instances $y_1, \ldots, y_M$ $\in B$ of the pattern ω. Test the null hypothesis that $y_1, \ldots, y_M$ and $x_1, \ldots, x_N$, are samples from the same underlying population, to a specified significance level ϵ.

Parameter Estimation: Suppose we are given a set of degraded instances $x_1, \ldots, x_N$ $\in B$ of the pattern ω. Estimate the degradation model parameter $\hat{\Theta}$ that can be used to generate degraded instances $y_1, \ldots, y_M \in B$ from the ideal pattern ω, such that $y_1, \ldots, y_M$ are close to $x_1, \ldots, x_N$ under a specified distance function.

6.2 Model Validation

In this section we describe a nonparametric validation procedure that can be used to statistically validate any document degradation model. Suppose we are given a sequence of real degraded characters $X = \{x_1, x_2, \ldots, x_N\}$, and another sequence of artificially degraded characters $Y = \{y_1, y_2, \ldots, y_M\}$ that were created by perturbing an ideal character with a document degradation model. We can assume that the characters x_i and y_i are binary matrices of size (approximately) 30×30. The question that needs to be addressed is whether or not the x_i's and the y_i's come from the same underlying population. At this point it does not matter where the x_i's and the y_i's came from. The x_i's and the y_i's could both be synthetically generated, or both be real instances, or one of them could be synthetic and the other real. A statistical hypothesis test can be performed to test the null hypothesis that the underlying population distributions of x_i's and y_i's are the same.

Standard parametric hypothesis testing procedures (χ^2 test etc.) are not applicable for many reasons: (i) the dimensions of x_i and y_i are not fixed, (ii) the vectors are binary and in particular not Gaussian, and (iii) the size of the space to which they belong is very large (approximately 2^{900} if we assume each character to be of dimension 30×30). Instead, we now describe a *nonparametric permutation test* (see [Goo94, ET93]) that performs this hypothesis test.

1. Given (i) the real data $X = \{x_1, x_2, \ldots, x_N\}$, (ii) the synthetic data $Y = \{y_1, y_2, \ldots, y_M\}$, (iii) a distance function, $\rho(X, Y)$, on sets, (iv) a distance func-

tion, $\delta(x, y)$, on characters, and (v) the size ϵ of the test (i.e. misdetection rate $= \epsilon$).

2. Compute $d_0 = \rho(X, Y)$.
3. Create a new sample $Z = \{x_1, \dots, x_N, y_1, \dots, y_M\}$. Thus Z has $N+M$ elements labeled z_i, $i = 1, \dots, N + M$.
4. Randomly permute (reorder) Z.
5. Partition the set Z into two sets X' and Y' where $X' = \{z_1, \dots, z_N\}$ and $Y' = \{z_{N+1}, \dots, z_{N+M}\}$.
6. Compute $d_i = \rho(X', Y')$.
7. Repeat steps 4, 5 and 6 K times to get K distances $d_1, \dots, d_K$.
8. Compute the distribution of d_i's empirically: $P(d \geq v) = \#\{k | d_k \geq v\}/K$
9. Compute the P-value: $\epsilon_0 = P(d \geq d_0)$.
10. Reject the null hypothesis that the two samples come from the same population if $\epsilon_0 < \epsilon$.

This method is also depicted in figure 6.1.

The above procedure computes the null distribution of the distance function $\rho(X, Y)$ nonparametrically. In a standard parametric hypothesis testing procedure, the form of the distributions of x and y are known (usually Gaussian) and so the null distribution of $\rho(X, Y)$ is known. In contrast, we compute the null distribution by randomly permuting the data set Z and creating a histogram of d_i's.

By design, the size of the test, ϵ, is fixed. Thus, irrespective of the distance function $\rho(X, Y)$, the percentage of time that the validation procedure rejects a true null hypothesis that the two samples are from the same underlying population is ϵ. In other words, the probability of misdetection is ϵ. What is not fixed is the probability of false alarm, γ. Moreover, although the use of various distance functions for ρ and δ gives rise to the same probability of misdetection, ϵ, each has a different probability

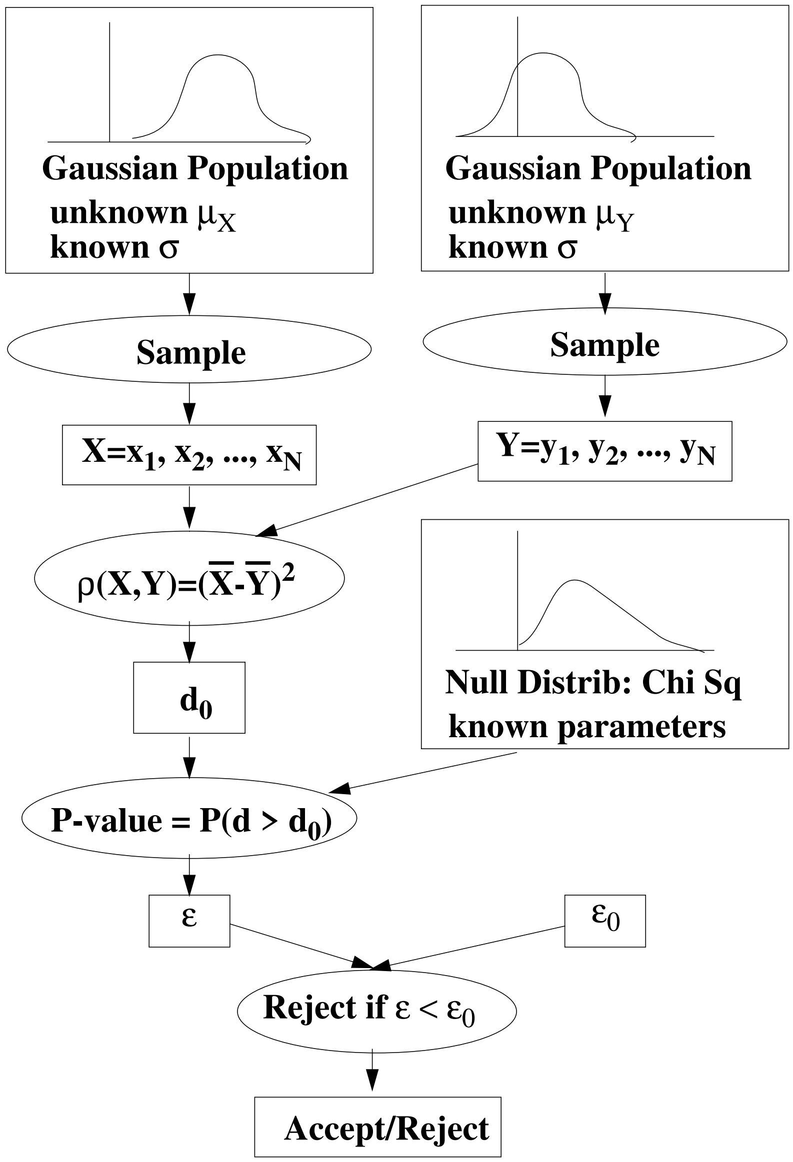


Figure 6.1: Here we show how the permutation test works when the two samples X and Y are know to be Gaussian distributed with unit variance. In this case the null distribution can be computed theoretically.

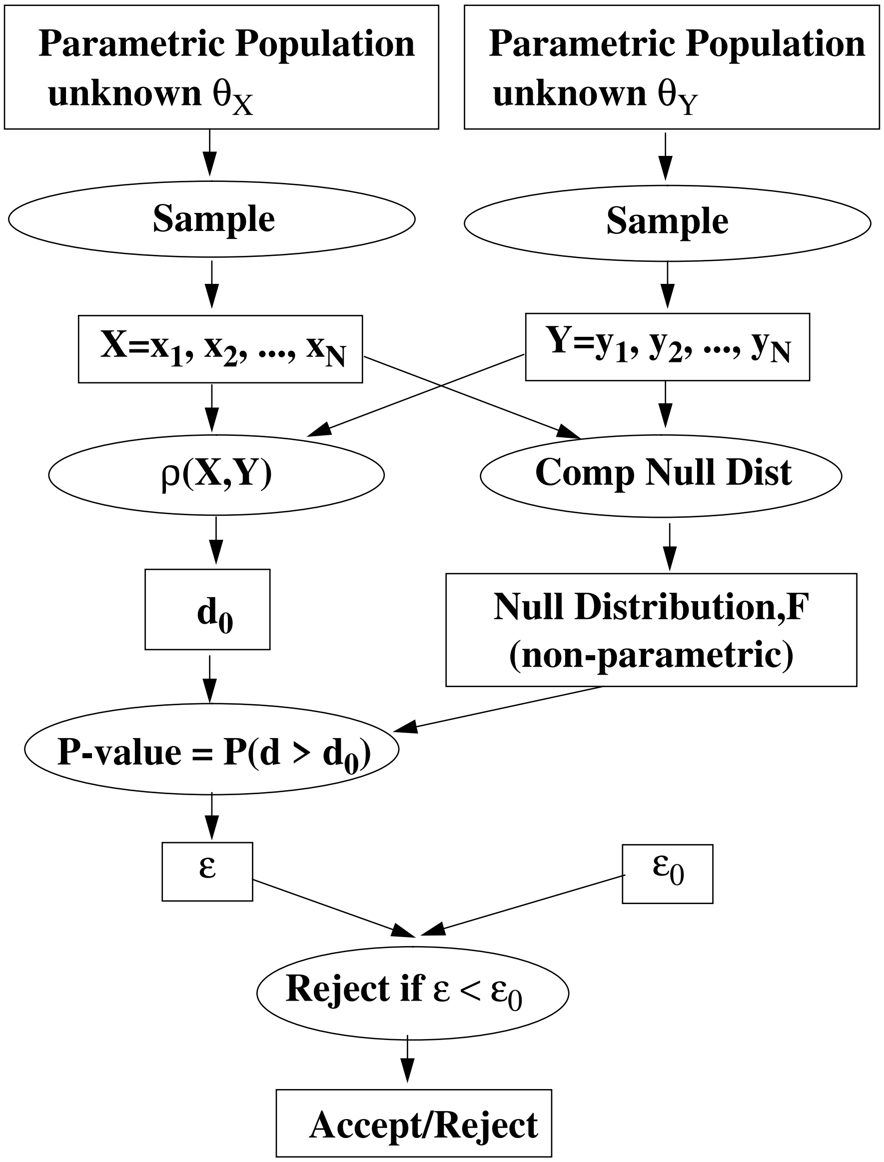


Figure 6.2: Here we show how the nonparametric test works when the two samples X and Y are from arbitrary distributions. For our problem, x_i and y_i are binary characters. In this case the null distribution cannot be theoretically.

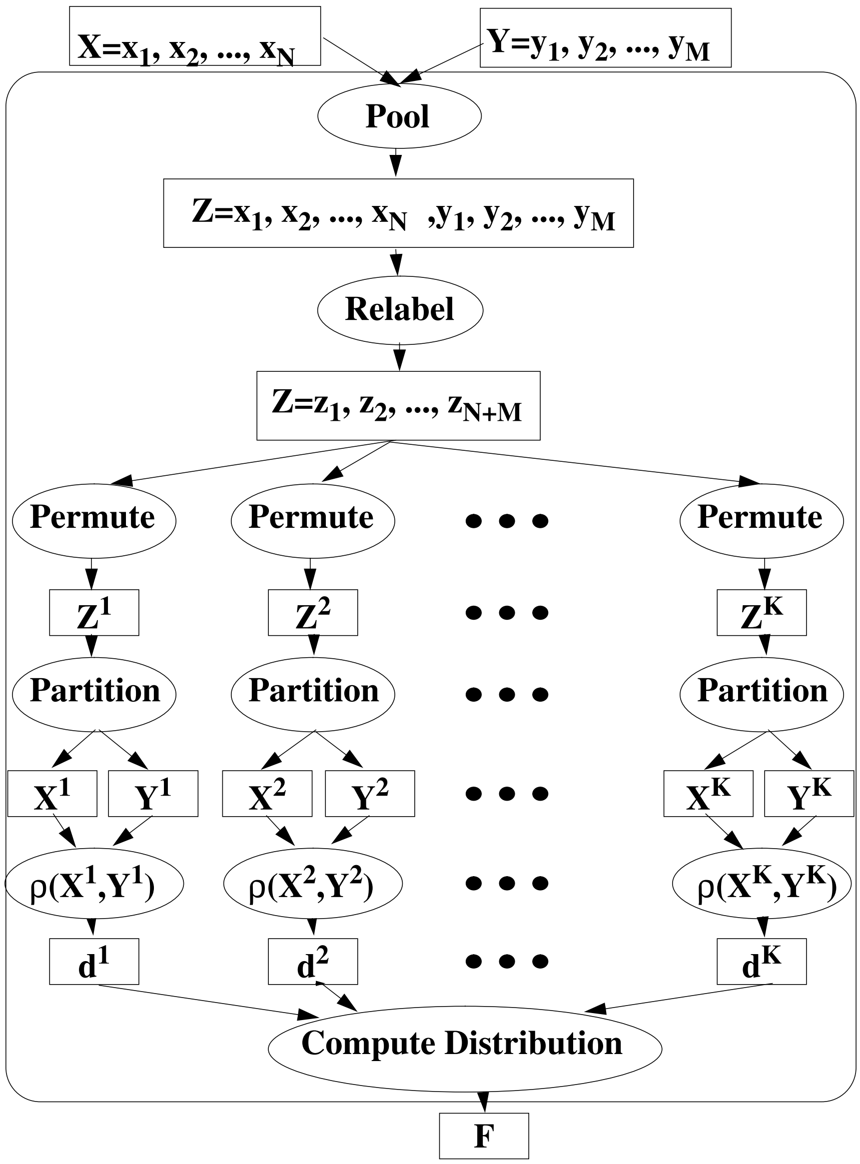


Figure 6.3: This figure shows the permutation procedure for computing the null distributions.

of false alarm, γ, which is the probability that the procedure claims that X and Y come from the same underlying populations when, in fact, they come from different underlying populations.

It is important to note that if two samples X and Y pass the validation procedure, it does not mean that we accept the null hypothesis. Rather, it means that we do not have enough statistical evidence to reject the null hypothesis. Nevertheless, when we reject a null hypothesis, it *does* mean that we have enough statistical evidence to reject it.

6.3 Power Functions

Let us assume that the x_i's are distributed as $F(\theta_X)$ and the y_i's are distributed as $F(\theta_Y)$, where θ_X and θ_Y are the parameters of the distributions. Let the null hypothesis, H_N, and the alternate hypothesis, H_A, be:

$$H_N \quad : \quad \theta_X = \theta_Y \tag{6.1}$$

$$H_A \quad : \quad \theta_X \neq \theta_Y \tag{6.2}$$

The size of the test, ϵ, is fixed by the algorithm designer and is given as

$$\epsilon = P(H_A | H_N \text{ is true}) \ . \tag{6.3}$$

The plot of 1 minus the probability of false alarm as a function of θ is the *power function.* Thus, if we fix the distribution parameter of the x_i's at $\theta_X = \theta_0$, and vary the distribution parameter value $\theta_Y = \theta$ for y_i's, the power function is denoted by $\gamma_{\theta_0}(\theta)$, and is given by:

$$\gamma_{\theta_0}(\theta) = P(H_A | \theta_X = \theta_0 \text{ and } \theta_Y = \theta) \ . \tag{6.4}$$

Thus $1 - \gamma_{\theta_0}(\theta)$ is the probability of false alarm. The power function should have a minimum at $\theta_X = \theta_Y = \theta_0$, with $\gamma_{\theta_0}(\theta_0) = \epsilon$, and should increase on either side and go up to 1 when $\theta_Y = \theta$ is very far from θ_0.

Let us say there are two validation schemes A and B with test size ϵ and power functions $\gamma^A_{\theta_0}(\theta)$ and $\gamma^B_{\theta_0}(\theta)$. Since the misdetection probability, ϵ, is the same for both schemes, A is better than B if the false alarm rate of A is less than the false alarm

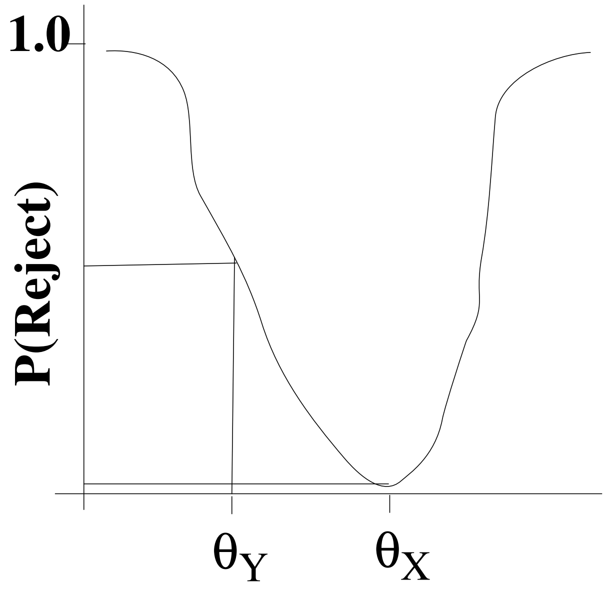


Figure 6.4: The true parameter of the sample X is Θ_X. The parameter Θ_Y of the sample Y is updated and the corresponding probability of the test rejecting the null hypothesis that X and Y are from same underlying distribution is plotted. The resulting curve is the power function.

rate of B. That is, A is better than B if $1 - \gamma^A_{\theta_0}(\theta) < 1 - \gamma^B_{\theta_0}(\theta)$ or $\gamma^A_{\theta_0}(\theta) > \gamma^B_{\theta_0}(\theta)$. If the above relation is true for all values of θ, then the procedure A is said to be uniformly more powerful than B. That is, the scheme A is better than scheme B if the power function plot of A is above the power function plot of B for all values of θ. Generalizing, if there are many validation schemes, the one whose power function is above all other power functions, is the best scheme. If the power functions intersect, there is no clear winner; for some regions in the parameter space one scheme is better while in other regions the other scheme is better.

For a given validation scheme, if we increase the sample sizes N and M, the power function changes and the new power function is higher than the old power function, and so by definition is more powerful. Thus, the sensitivity, i.e., the width of the notch at the minimum, is a function of the sample sizes N and M. When the sample size is small, the notch is broader and when the sample size is large, the notch is sharper. This fact is used in deciding what sample size should be used for the test: choose the sample size such that the desired probability of false alarm is attained when the parameters θ_X and θ_Y differ by a specified amount $\Delta\theta$.

Finally, our validation scheme described in the previous section is dependent on two distance functions ρ and δ. Thus, each choice of ρ and δ gives rise to a different power function. The combination that produces the highest power function is the best choice. See [Arn90] for details on power functions.

6.4 Distance Functions, Outliers, and Robust Statistics

Various distance functions $\rho(X, Y)$ can be used for computing the distance between the sets of characters X and Y. We use the following symmetric distance functions for ρ.

Mean Nearest Neighbor Distance:

$$\rho(X,Y) = \rho_{Mean}(X,Y) = \frac{(\rho_{Mean}(Y;X) + \rho_{Mean}(X;Y))}{(N+M)}$$

where,

$$\begin{aligned} \rho_{Mean}(Y;X) &= \sum_{x \in X} \left(\min_{y \in Y} \delta(x,y) \right) \\ \rho_{Mean}(X;Y) &= \sum_{y \in Y} \left(\min_{x \in X} \delta(x,y) \right) \end{aligned}$$

Trimmed Mean Nearest Neighbor Distance:

$$\rho(X,Y) = \rho_{Trim}(X,Y) = (\rho_{Trim}(Y;X) + \rho_{Trim}(X;Y))/2$$

where,

$$\begin{aligned} \rho_{Trim}(Y;X) &= \mathrm{Trim}_{x \in X} \left(\min_{y \in Y} \delta(x,y) \right) \\ \rho_{Trim}(X;Y) &= \mathrm{Trim}_{y \in Y} \left(\min_{x \in X} \delta(x,y) \right) \end{aligned}$$

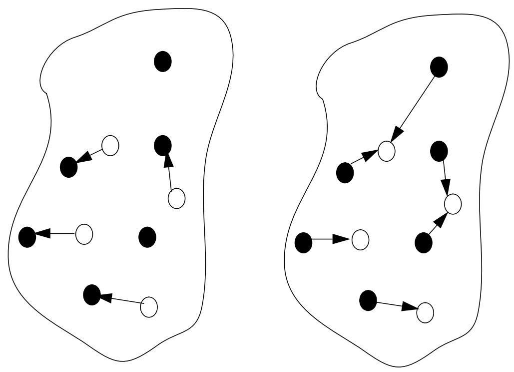
Here the *Trim* function accepts as input a set of real numbers, orders them in an increasing order, discards the top and bottom 10%, and returns the mean of the rest 80%.

Median Nearest Neighbor Distance:

$$\rho(X,Y) = \rho_{Med}(X,Y) = (\rho_{Med}(Y;X) + \rho_{Med}(X;Y))/2$$

where,

$$\begin{aligned} \rho_{Med}(Y;X) &= \mathrm{Median} \left(\min_{y \in Y} \delta(x,y) \right) \\ \rho_{Med}(X;Y) &= \mathrm{Median} \left(\min_{x \in X} \delta(x,y) \right) \end{aligned}$$



$$\rho(X;Y) = (u_1 + u_2 + \ldots + u_4)/4$$

$$\rho(Y;X) = (v_1 + v_2 + \ldots + v_5)/5$$

$$\rho(X,Y) = (\rho(X;Y) + \rho(Y;X))/2$$

Figure 6.5: The black dots are elements of the set X and the white dots are elements of the set Y. In the figure on the left, the distance $\rho(X;Y)$ from Y to X is computed by summing the distance of each y_i to the nearest x_i. Similarly on the right the distance $\rho(Y;X)$ is computed. The final symmetric distance $\rho(X,Y)$ is computed by taking the mean.

Notice that the mean nearest neighbor (NN) distance is not a robust distance measure. That is, if for some reason, a data point is far from norm, the P-value computation becomes very sensitive to this data point. This can occur when a character in the real data set X is actually a 'c' (instead of being an 'e,'), and is identified wrongly as 'e'. Yet another outlier source is connected characters: when characters are extracted from a real document, they might be touching other characters, pieces of which might slip in. The Median and the Trimmed Mean distance measures are robust against outliers since they do not look at the tails of the distribution. One would expect that these should work better in the cases where there are outliers.

The distance function, $\delta(x, y)$, mentioned earlier is the distance between two individual characters x and y. We use the Hamming distance for δ. This is computed by counting the number of pixels where the characters x and y differ after the centroids of x and y have been registered. A variety of other character distances, $\delta(x, y)$, and set distance functions, $\rho(X, Y)$, could have been used. (e.g. the Hausdorf distance, rank ordered Hausdorf distance, etc.) The combination of character distance $\delta(x, y)$, and set distance, $\rho(X, Y)$, that give rise to the best power function is the best pair of distances to use for the validation procedure.

6.5 Estimation of the Degradation Model Parameters

Given a degraded document we would like to estimate the parameters, $\hat{\Theta}$, of the degradation model that could be used to create degraded documents which are "similar" in the sense discussed earlier.

We use the following procedure to estimate the parameter vector $\hat{\Theta}$.

1. Given a fixed sample X of size N and an inital guess of Θ.

2. Generate a sample Y of size M and with model parameter Θ.

3. Check if the validation procedure accepts the null hypothesis that X and Y come from the same underlying population.

4. Repeat K times steps 2 and 3 and estimate the reject rate.

5. Change the parameter Θ of the sample Y and repeat steps 2 through 4, to get a reject rate function.

6. Find the parameter value Θ_0 where the reject rate function is minimum.

7. Θ_0 is the best estimate.

There is a subtle difference between the power function and the reject rate function generated in the estimation procedure. In the validation procedure, the power function is generated by creating new samples of X and Y, in each step. However, during estimation, we have only one given sample of X, which is fixed in all the experiments, while multiple samples of Y are generated. In chapter 7 we use this method to estimate the parameters of the degradation model.

6.6 Comparing Two models

Let us say there are two document degradation models M_1 and M_2. The problem is to find the model that is closer to the real process. We know that if the sample size N of the synthetic samples and the real samples is increased, after a certain point the validation procedure will start rejecting both the models. However, we will now give a procedure that will allow a researcher to decide which model is closer to reality for a fixed sample size N.

1. Fix the sample size N.

2. We are given the real sample D of size N.

3. Genrate synthetic samples S_1 and S_2 of size N using the models M_1 and M_2 respectively.

4. Conduct the two sample validation test using the real sample D and the synthetic sample S_1. Let the associated P-value be p_1.

5. Conduct the two sample validation test using the real sample D and the synthetic sample S_2. Let the associated P-value be p_2.

6. If $p_1 > p_2$, the model M_1 is closer to the real process for a sample size of N. Otherwise the model M_2 is closer.

Thus the above procedure allows a researcher to choose between models.

When we were choosing between parameter setting for a fixed model, we could use the power function to arrive at the best parameter sitting. However, two different models have different parameter space and hence they cannot compared using power functions. The P-value forms our means of comparing the models on a common basis.

6.7 Discussion

In this chapter we gave non-parametric procedures for validating and estimating degradation models. The validation procedure is a two-sample permutation test. One sample is a set of real characters, and the other sample is a set of synthetically degraded characters. The null distribution of a (given) sample distance function is constructed by a random permutation process.

We gave power functions for our validation procedures. The power functions enable give us a way of choosing between distance functions and other parameters that a validation procedure may have. The parameter value that gives rise to the lowest power function is the best parameter setting. They also allow us to study the probability of rejecting the null hypothesis over the parameter space as a function of the sample size.

In many scenarios, such as when there are outliers in the data set, an exact hypothesis test always ends up in rejecting the null hypothesis when the sample size is made large enough. Equality of distributions not really the kind of test one would like to conduct. Rather, one would like to know if two distributions are 'close enough.' We showed that using robust set distance functions is one way of performing such approximate tests.

We made use of a variant of the power function procedure for estimating the parameters of the model. Given a real sample, we generated synthetic samples and estimated the parameter of the model by sampling the parameter space. Two sampling procedures were used. First was a brute force search where we sampled each parameter at equal intervals and searched over all the possibilities. The second was a line search procedure where optimal value of the objective function was computed along an axis in the parameter domain. Once the optimum value was found, the particular parameter value was frozen, and the objective value was minimized along another axis. This procedure was repeated three times.

Chapter 7

EXPERIMENTAL PROTOCOL AND RESULTS

In this chapter we outline the protocol we use to conduct the experiments. Here we give all the sample sizes we use, the number of trials that are run at different stages, the exact model parameter values that are used for generating the synthetically degraded characters, etc. The purpose of this section is twofold: first, to design experiments that validate the theoretical formulations developed in the previous chapter; and second, to provide enough information so that anyone can replicate our experiments;

There are three types of experiments possible:

Synthetic vs. Synthetic: One sample X is synthetically created using the document degradation model, with a fixed model parameter value. Then many samples Y are generated, again using the model, but with different parameter setting. The validation procedure can be run on the samples X and Y, and the power function generated. This experiment is in part a sanity check for the methodology: if it does not work on controlled synthetic data, there is little point in trying it on real data. Also, the parameter estimation methodology can be studied in this way since the true parameter Θ is known and the variance of the estimated parameter $\hat{\Theta}$ can be calculated.

Real vs. Real: This experiment tests for systematic dissimilarities between two image populations (e.g. rotations, fonts, etc.). Note that this use of the validation procedure is independent of degradation models.

Real vs. Synthetic: Here the sample X consists of real degraded characters and the sample Y is generated by varying the degradation model parameter Θ. The validation procedure is run on the X and Y samples, and a power function is generated. This experiment tests whether or not the synthetic characters are actually close to the real characters.

7.1 Protocol for Synthetic vs. Synthetic

The following protocol is used for creating the samples X and Y. The distribution parameter Θ_X is fixed with the following parameter component values: $\eta_f = \eta_b = 0$, $\alpha_0 = \beta_0 = 1$, $\alpha = \beta = 1.5$, and the structuring element size $k = 5$. The distribution parameter Θ_Y is varied by varying α and β. In our experiments we make α equal to β. The other parameter components of Θ_Y – $\eta_f, \eta_b, \alpha_0, \beta_0, k$ – are made equal to the corresponding components of the model parameter Θ_X. In all cases the noise-free document is the same (a LaTeX document page formatted in IEEE Transaction style) and the same set of 340 character 'e' (Computer Modern Roman 10 point font) are extracted from the page, for creating the sample X and the sample Y.

The validation procedure parameters used are as follows:

1. Sizes of samples, X, and Y: $N = M = \{10, 20, 60\}$.
2. Number of permutations: $K = 1000$.
3. Significance level of the test: $\epsilon = 0.05$.
4. Number of repetitions, T, for computing the power function: $T = 100$.
5. The character-to-character distance, $\delta(x, y)$, used is the Hamming distance.
6. The set-to-set distance, $\rho(X, Y)$, used is the mean nearest-neighbor distance.

The noise-free document is shown in Figure 7.1(a). The degraded document generated with model parameter Θ_X is shown in Figure 7.1(b). The power function for the sample sizes 10, 20, 60 are shown in Figure 7.2. The power function corresponding to sample size 10 is the widest, and the power function corresponding to sample size 60 is the narrowest. Note all the three power functions give a misdetection (reject) rate close to $\epsilon = 0.05$ when the Θ_Y is close to Θ_X. (Only the α component, which is equal to 1.5 for Θ_X, is shown in the plot.) Furthermore, when the α component for Θ_Y is far from 1.5, the misdetection rate is close to 1.0, which implies that the validation procedure can distinguish the two samples with high probability. An image generated with $\alpha = \beta = 1.7$ that the validation procedure accepted with a probability close to 0.9, is shown in Figure 7.1(c). Two document images generated with parameter values $\alpha = \beta = 2.0$ and $\alpha = \beta = 0.9$ that are easily rejected by the validation procedure are shown in Figure 7.1(d) and Figure 7.1(e), respectively.

7.2 Protocol for Real vs. Real Experiment

In this section we outline the experimental protocol that is used to validate the real-degraded characters against real-degraded characters.

First, various European language texts are generated using the Adobe Times-Roman typeface at 8 point. Next, these documents are printed on a Cannon laser printer and then scanned at 400 pixels per inch using a Cannon scanner. Lower-case 'e's are extracted semiautomatically by OCR (thus some characters possess artifacts resulting from resegmentation). From among these, 3000 characters are selected by two persons working independently to avoid misclassifications.

Before selecting the two populations, we randomly shuffle the real data in order to obscure any systematic perpage dissimilarities (due to, for example, skew scale variations). The validation procedure does not reject the null hypothesis that the two samples are from the same underlying population. Repeated trials give a reject rate close to 0.05, the significance level designed into the test.

7.3 Outliers and Distance Function Comparisons

The validation procedure protocol is as follows: the significance level ϵ is fixed at 0.05; the sample sizes $N = M$ used are 10, 20, and 60; the number of permutations K for creating the empirical null distribution is 1000; the number of trials T for estimating the misdetection rate is 100.

We studied the sensitivity of the validation procedure to the set distance $\rho(X, Y)$ as follows. The data sets X and Y are collections of (synthetic) degraded character 'e'. Degradation parameter values for X are fixed at $\alpha = \beta = 1.5$, but the corresponding degradation parameters for Y are varied from 0.6 to 2.4. The Hamming distance is used for the character-to-character distance, $\delta(x, y)$. Sample size of X and Y is fixed at $N = M = 60$. The mean, trimmed mean and median distances are used to compute the power function, both, in the presence and in the absence of outliers.

Figures 7.3(a), 7.4(a), and 7.5(a), show the power functions in the absence of

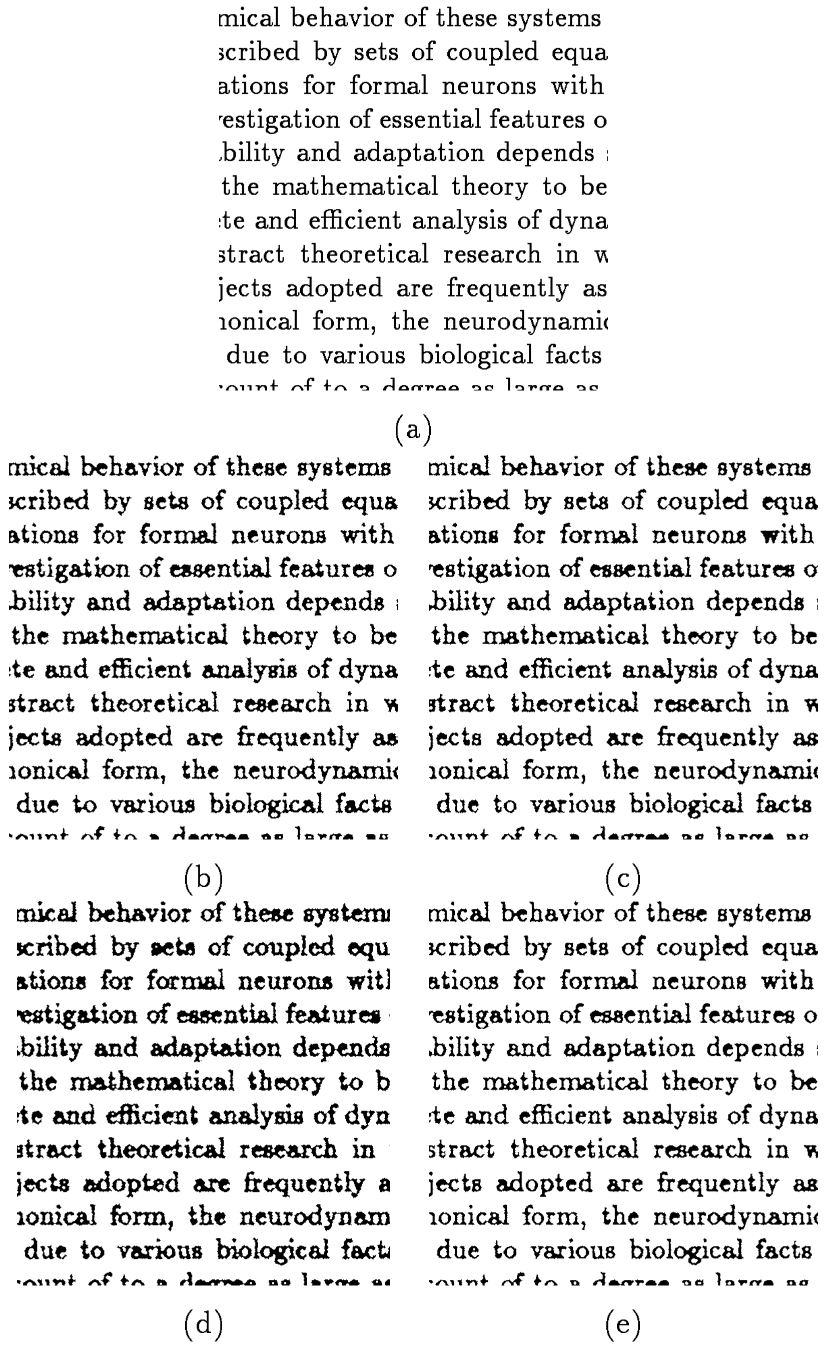


Figure 7.1: Local document degradation model. (a) Subimage of the noise free document. (b) Reference degraded document generated with $\alpha = \beta = 1.5$. (c) Probe sample accepted, $\alpha = \beta = 1.7$. (d) Probe sample rejected, $\alpha = \beta = 0.9$. (e) Probe sample rejected, $\alpha = \beta = 2.0$. Sample size used is 60.

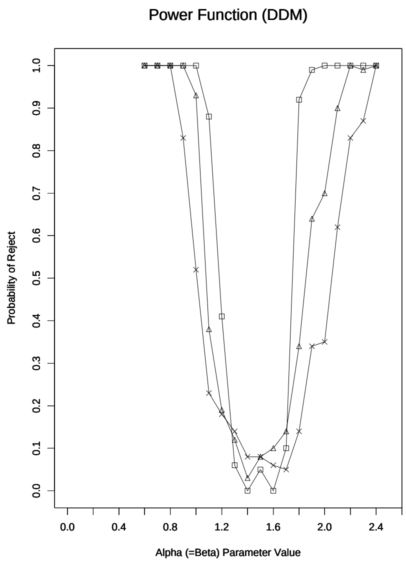
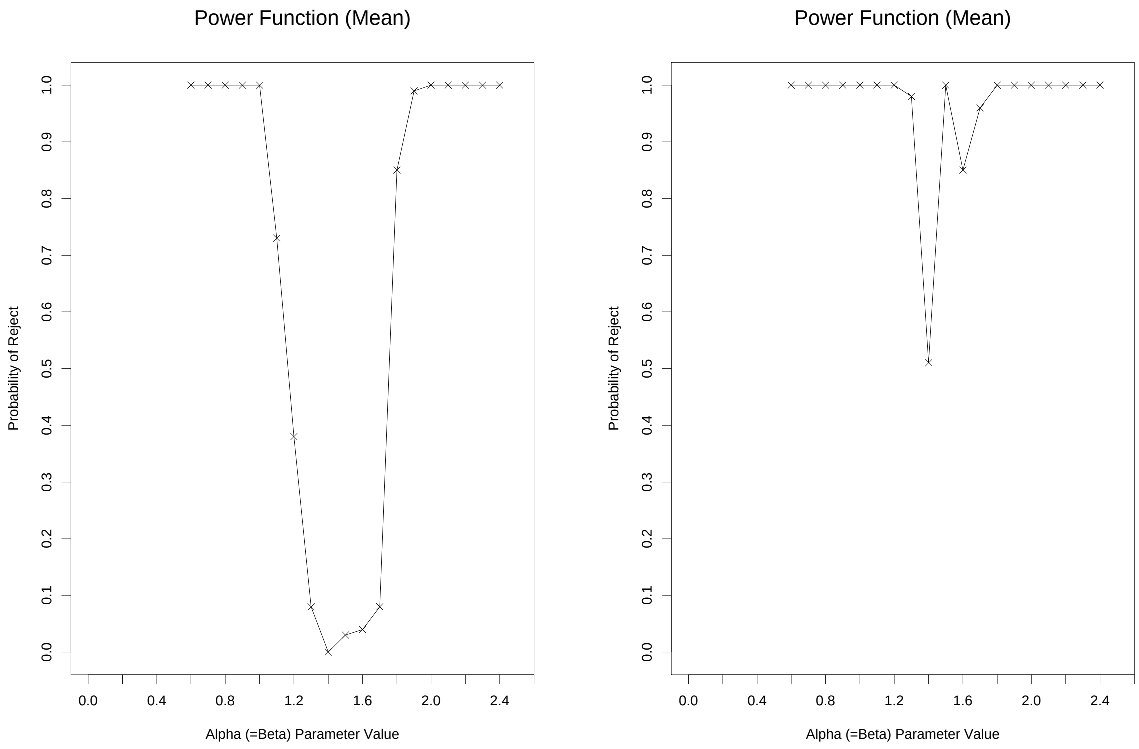


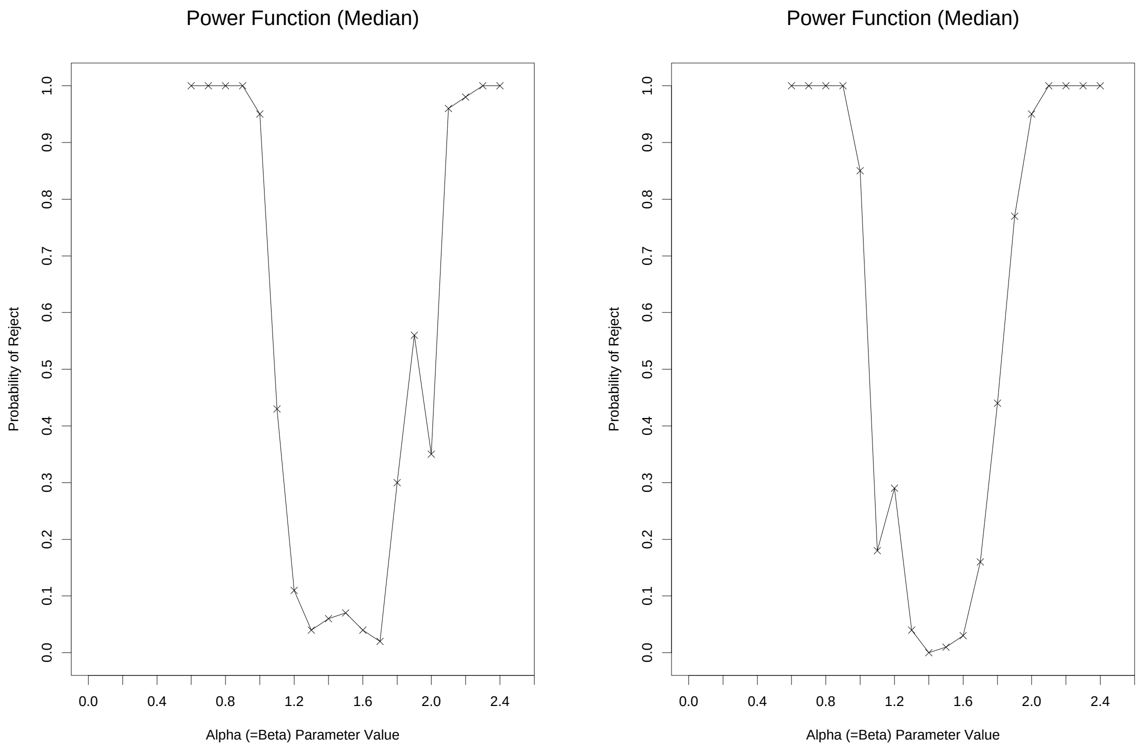
Figure 7.2: Power plots for the local document degradation model. The reference distribution had $\alpha = \beta = 1.5$. Notice that the power function has a minimum near $\alpha = \beta = 1.5$. The power function corresponding to sample size of 60 (boxes), is sharper; that corresponding to a sample size of 10 (crosses) is broader.



(a) (b)

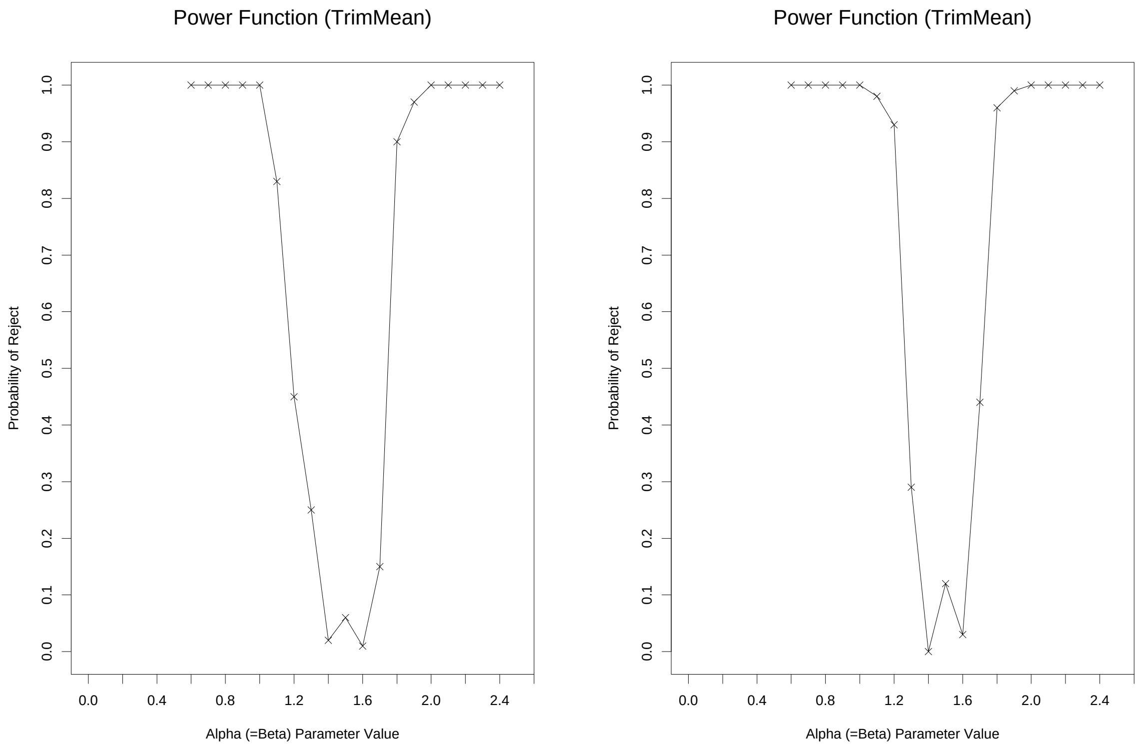
Figure 7.3: Power functions of the validation procedure when mean nearest neighbor distance is used for the set distance functions $\rho(X, Y)$. Figure (a) is when there are no outliers. Figure (b) corresponds to the situation when there are 5 outliers in one of the data sets.

outliers when the mean, trimmed mean distances are used. Next, we introduced outliers in the data set X by substituting 5 degraded 'e's with degraded 'c's. The Y data set is unchanged. Figures 7.3(b), 7.4(b), and 7.5(b), show the power functions in the presence of outliers. Clearly the median and trimmed mean nearest neighbor distances are more robust against outliers, since the corresponding power functions are not affected. Furthermore, it can be seen that the median NN distance function, in the outlier-free case, is less 'powerful' than the mean distance function since the function lies below the mean NN power function plot. Finally, it can be seen that the 10 % trimmed NN distance function is superior to the other two distance functions, since the corresponding power function is robust against outliers and at the same time higher.



(a) (b)

Figure 7.4: Power functions of the validation procedure when median nearest neighbor distance is used for the set distance functions $\rho(X, Y)$. Figures (a) is when there are no outliers. Figure (b) corresponds to the situation when there are 5 outliers in the X data set.



(a) (b)

Figure 7.5: Power functions of the validation procedure when 10% trimmed mean nearest neighbor distance is used for the set distance functions $\rho(X, Y)$. Figures (a) is when there are no outliers in the data X and Y. Figure (b) corresponds to the situation when there are 5 outliers in the X data set.

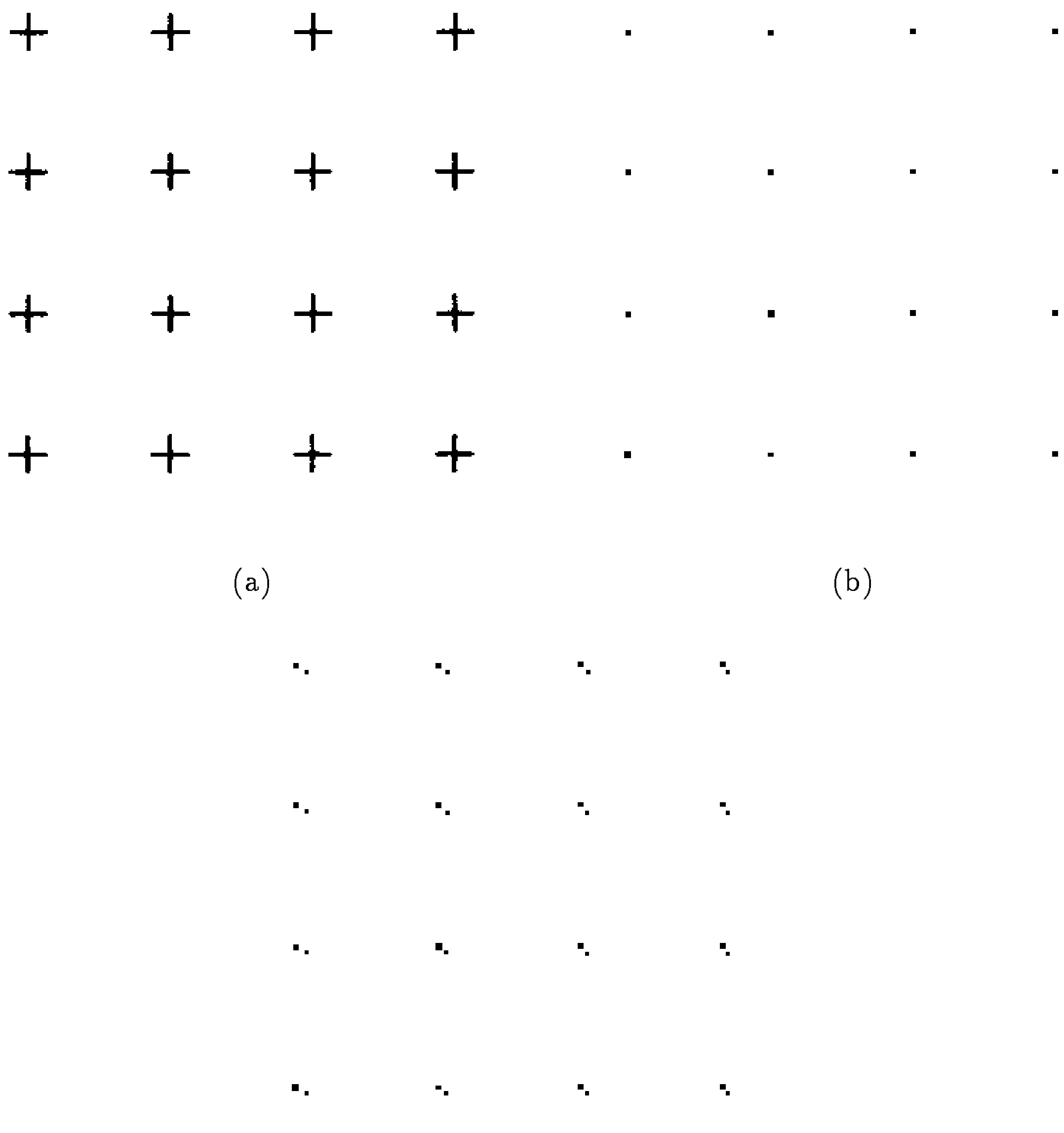
7.4 Protocol for Calibration

The ideal image for calibrating the printer-photocopier-scanner process is created as follows. First a grid of equally spaced "+" symbols is arranged on a 3300×2500 binary image. The vertical and horizontal bars of the "+" symbol are 25 pixels long and 3 pixels thick. The number of symbols on each row and column of the grid are 23 and 30, respectively.

The ideal image is then printed and scanned. The intersection points of the two bars of the "+" symbols are used as the calibration points. The calibration points are detected by a morphological algorithm: first the image is closed with a 3×3 square structuring element. Next, two images are created by opening the closed image with a vertical and horizontal structuring elements, respectively. Calibration points on the scanned image are detected by binary-anding these two images. A connected component algorithm is then run on the image with the detected calibration points. The centroids of the connected components are used as the coordinates of the calibration points. The calibration points in the ideal image are known since the ideal calibration image is created under experimenter's control.

To estimate the projective transform, four feature points are first detected using the algorithm described in chapter 5. Next, we estimate the projective transform parameters from the ideal and real points (correspondences are known since we order the four points in a counter clockwise order, starting with the upper left feature point, and assume that the orientation of the page is unchanged). The estimated transform parameters are then used to project all the ideal points. An exhaustive search is conducted to establish correspondences between the projected ideal calibration points and real calibration points. That is, for each projected ideal point, we find the closest real point, and assume the two points match. A registration error vector, which is the error between each real calibration point and the projected calibration point, is computed for each calibration point. The maximum error we attain is with ± 4 pixels in each coordinate.

In Figure 7.6(a) we show a subimage of the scanned calibration document. The detected calibration points are shown in Figure 7.6(b). In Figure 7.6(c) the ideal calibration points are transformed using the estimated projective transformation and overlaid on the real calibration points. A scatter plot of the error vectors is shown in Figure 7.7.



(a) (b)

(c)

Figure 7.6: (a) A subimage of the scanned calibration document. The detected calibration points are shown in (b). (c) The ideal calibration points are transformed using the estimated projective transformation and overlaid on the real calibration points.

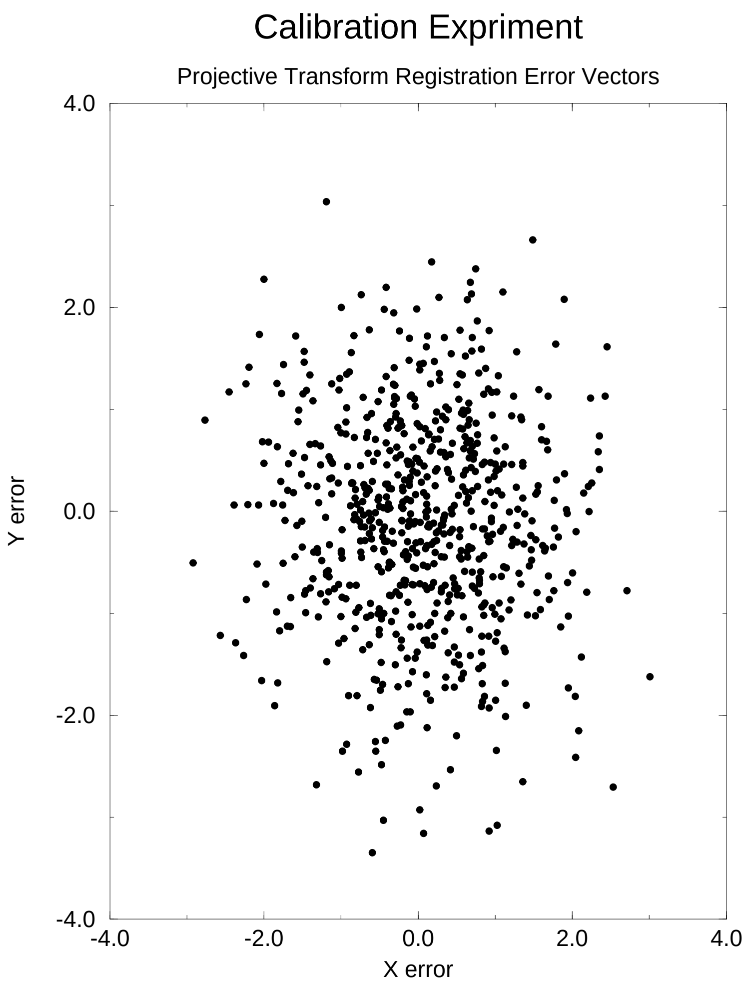


Figure 7.7: A scatter plot of the error vectors computed between real calibration points and projected ideal calibration points.

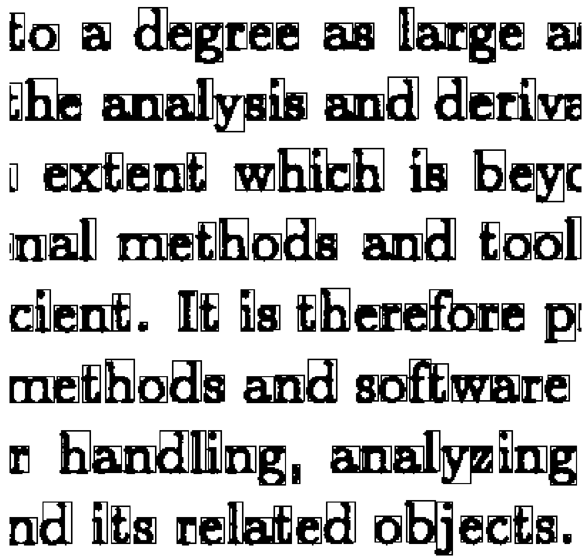
7.5 Data Collection

The ideal data is a LaTeX formatted document. The IEEE Transaction style is used for typesetting the document. The corresponding ideal binary image and character ground truth is created using the DVI2TIFF software. The ideal document is created at 300×300 dots/inch resolution and the size of the binary document in pixels is 3300×2550. This document is printed using a SparcPrinter II. Next, the original printed document is photocopied five times using a Xerox photocopier - once at the normal setting, twice with darker settings, and twice with lighter settings. Finally the five photocopied documents are scanned using a Ricoh scanner. The scanner is set at 300×300 dots/inch resolution. The rest of the scanner parameters are set at normal settings. The scanned binary image is of size 3307×2544.

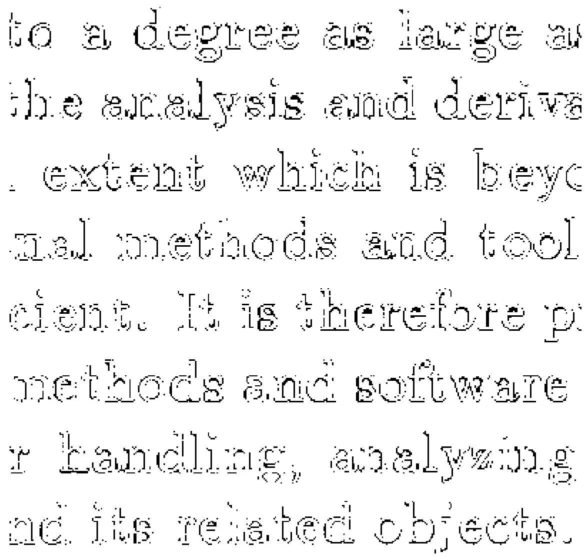
7.6 Protocol for Generating Real Ground Truth

Once the real scanned documents have been gathered as described in the previous section, we use the registration algorithm, described in chapter 5 to i) transform the ideal binary documents so that it registers to the scanned document and ii) to create the ground truth corresponding to the scanned document. The transformed ground truth also forms the ground truth for the transformed ideal document. The local nonlinearities of the transformation are accounted for by searching in a local neighborhood for a good match between the ideal character symbol and the real character symbol. The local template match window size is determined by the calibration experiment we performed earlier. Since the maximum error in the registration is ± 4 pixels, we used a window with $-7 \leq \Delta x, \Delta y \leq 7$. The ground truth generated by our algorithm is highly accurate. A subimage of the scanned image with the overlaid bounding box is shown in Figure 7.8. An exclusive or-ed image of the real scanned document and the registered ideal document is show in Figure 7.8. The time taken for this procedure on a SUN SPARC 5, is 2 minutes.

Our model validation methodology requires a sample of degraded bit patterns corresponding to characters. We extract the degraded bit patterns corresponding to the characters 'e,' 'a,' and 's,' of 10 point size and Computer Modern Roman font. The number of 'e's on a typical IEEE Transactions page is 300.

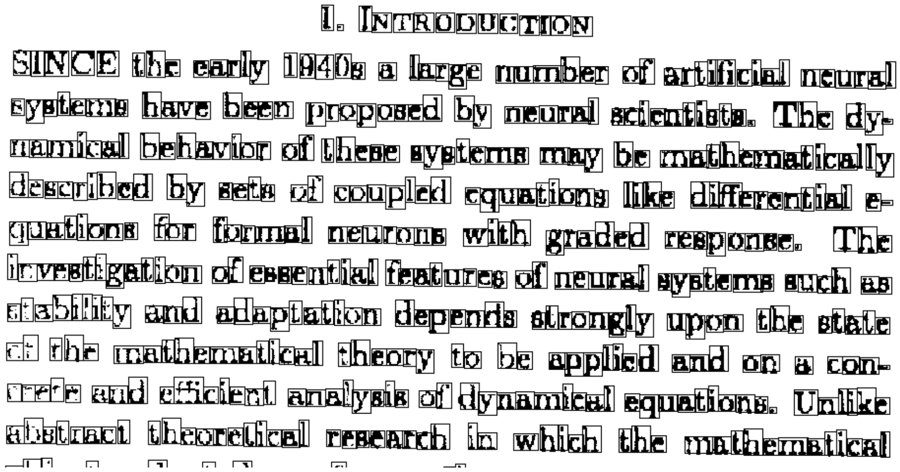


(a)



(b)

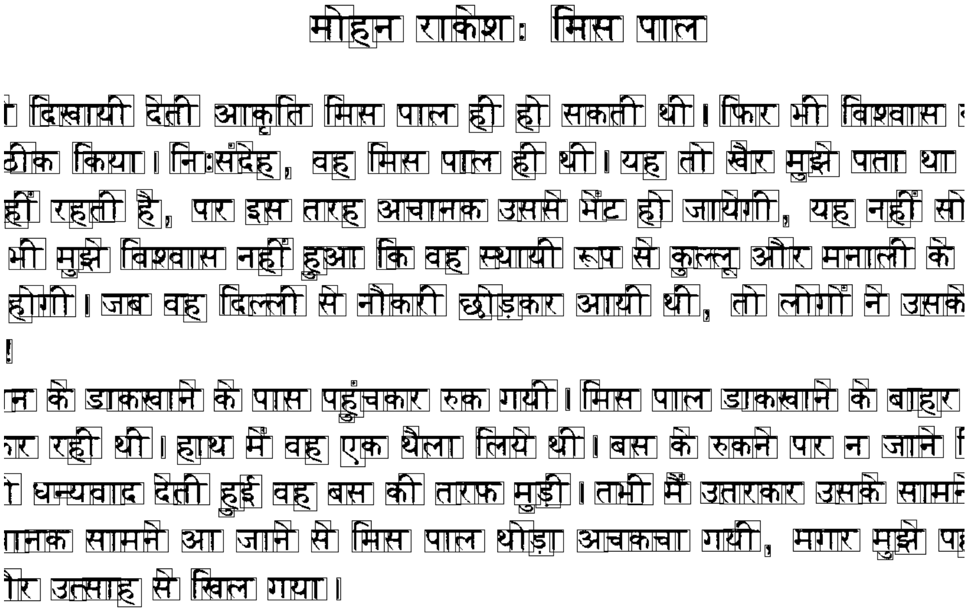
Figure 7.8: Ground truth for real documents. (a) shows a subimage of a document with the estimated bounding boxes of each character. (b) shows the result of exclusive-OR between the real document and the registered ideal document.



I. INTRODUCTION

SINCE the early 1940s a large number of artificial neural systems have been proposed by neural scientists. The dy-namical behavior of these systems may be mathematically described by sets of coupled equations like differential e-quations for formal neurons with graded response. The investigation of essential features of neural systems such as stability and adaptation depends strongly upon the state of the mathematical theory to be applied and on a con-crete and efficient analysis of dynamical equations. Unlike abstract theoretical research in which the mathematical

Figure 7.9: A submimage of a FAXed document with the ground truth overlaid.



मोहन राकेश: मिस पाल

ी दिखायी देती आकृति मिस पाल ही हो सकती थी। फिर भी विश्वास
ठीक किया। निःसंदेह, वह मिस पाल ही थी। यह तो खैर मुझे पता था
हीं रहती है, पर इस तरह अचानक उससे भेंट हो जायेगी, यह नहीं सो
भी मुझे विश्वास नहीं हुआ कि वह स्थायी रूप से कुल्लू और मनाली के
होगी। जब वह दिल्ली से नौकरी छोड़कर आयी थी, तो लोगों ने उसके
!

न के डाकखाने के पास पहुंचकर रुक गयी। मिस पाल डाकखाने के बाहर
ार रही थी। हाथ में वह एक थैला लिये थी। बस के रुकने पर न जाने
ो धन्यवाद देती हुई वह बस की तरफ मुड़ी। तभी मैं उतरकर उसके साम
ानक सामने आ जाने से मिस पाल थोड़ा अचकचा गयी, मगर मुझे प
ौर उत्साह से खिल गया।

Figure 7.10: A submimage of a Hindi document in Devanagri script with the ground truth overlaid.

7.7 Protocol for Estimating Real Model Parameters

The seven dimensional parameter space is sampled according to the protocol given below. For each parameter setting, a synthetically degraded document is created, and the statistic value is computed between the scanned degraded and the synthetic degraded images. The parameter setting for which the statistic value is minimum is used as the optimal estimate of the model parameters. The sampling protocol for the parameter space we used is:

$$\alpha_0 = \{0.0 + 0.2i, i = 0, ..., 5\} \tag{7.1}$$
$$\alpha = \{0.5 + 0.17i, i = 0, .., 15\} \tag{7.2}$$
$$\beta_0 = \{0.0 + 0.2i, i = 0, ..., 5\} \tag{7.3}$$
$$\beta = \{0.5 + 0.17i, i = 0, .., 15\} \tag{7.4}$$
$$k = \{1, 2, 3, 4, 5, 6, 7\} \tag{7.5}$$
$$d = \{0, 1, 2\} \tag{7.6}$$
$$\eta_0 = 0.0. \tag{7.7}$$

In Figure 7.11(a) a subimage of a scanned document is shown. The estimated parameter for this documents is

$$(\alpha_0 = 0.4, \alpha = 2.0, \beta_0 = 0.0, \beta = 0.0, d = 1, k = 2, \eta = 0.0).$$

In Figure 7.11(b) the corresponding synthetically degraded document is shown.

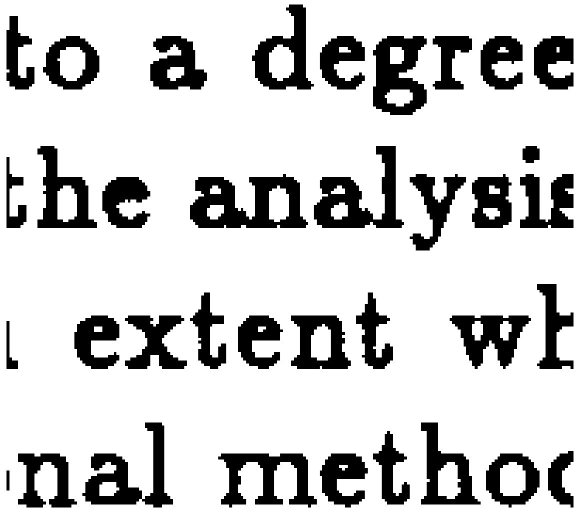
7.8 Protocol for Validating Real vs. Synthetic Degradations

Real data is first collected using the protocol outlined in section 7.6. The parameters are then estimated using the protocol specified in section 7.7.

In all cases the noise free document is the same (a LaTeX document page formatted in IEEE Transaction style) and the same set of 340 character 'e' (Computer Modern Roman 10 point font) are extracted from the page, for creating the synthetic population Y.

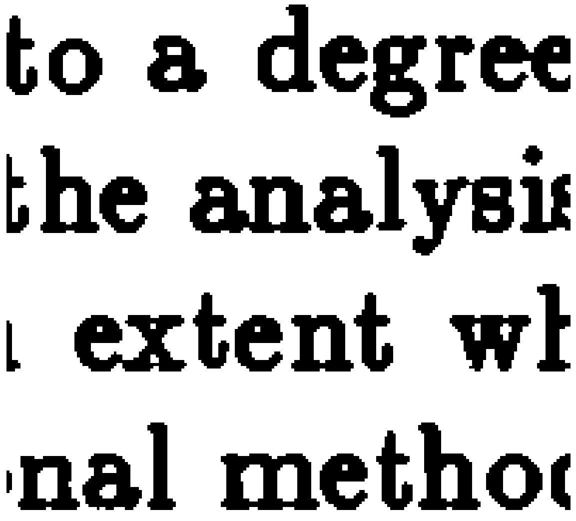
The validation procedure parameters used are as follows:

1. Sample sizes of scanned characters, X, and synthetic characters, Y: $N = M = \{10, 20, 60\}$.



to a degree
the analysis
extent wh
nal method

(a)



to a degree
the analysis
extent wh
nal method

(b)

Figure 7.11: (a) Real document. (b) Synthetically degraded document. The parameters used for the simulation are estimated from the real document. The parameter values used are: $\alpha_0 = 0.4$, $\alpha = 2.0$, $\beta_0 = 0.0$, $\beta = 0.0$, $d = 1$, $k = 2$, $\eta = 0.0$.

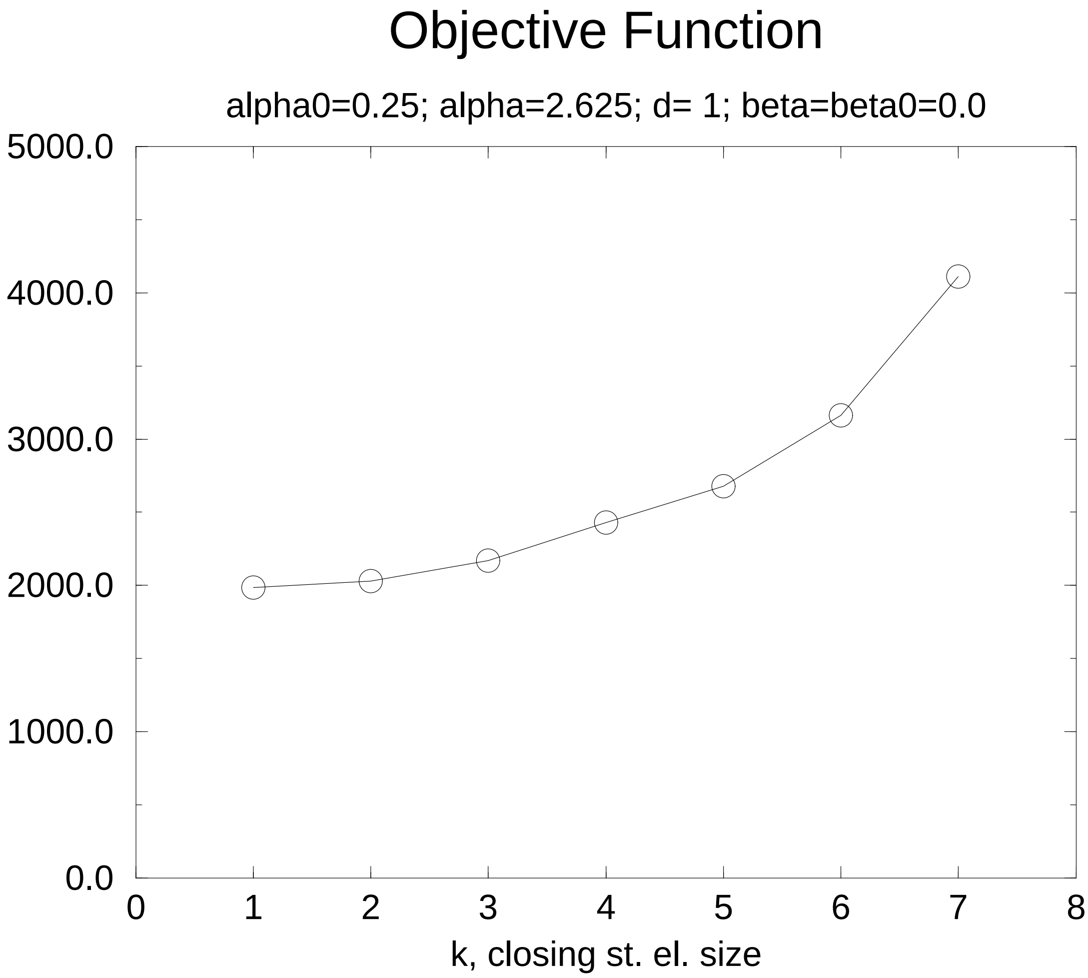


Figure 7.12: The objective value as a function of the structuring element size, k. Rest of the parameters are fixed at the optimal solution.

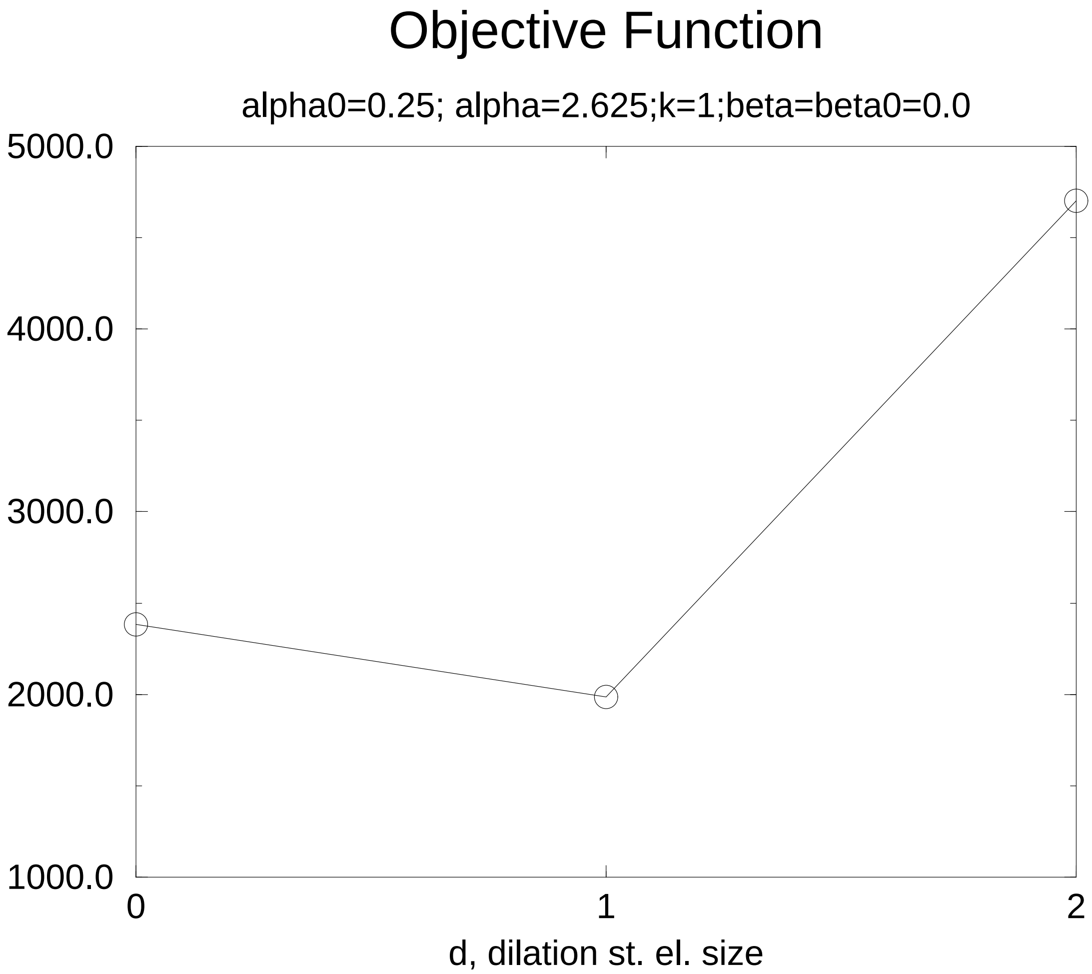


Figure 7.13: The objective value as a function of the structuring dilation structuring element size, k. Rest of the parameters are fixed at the optimal solution.

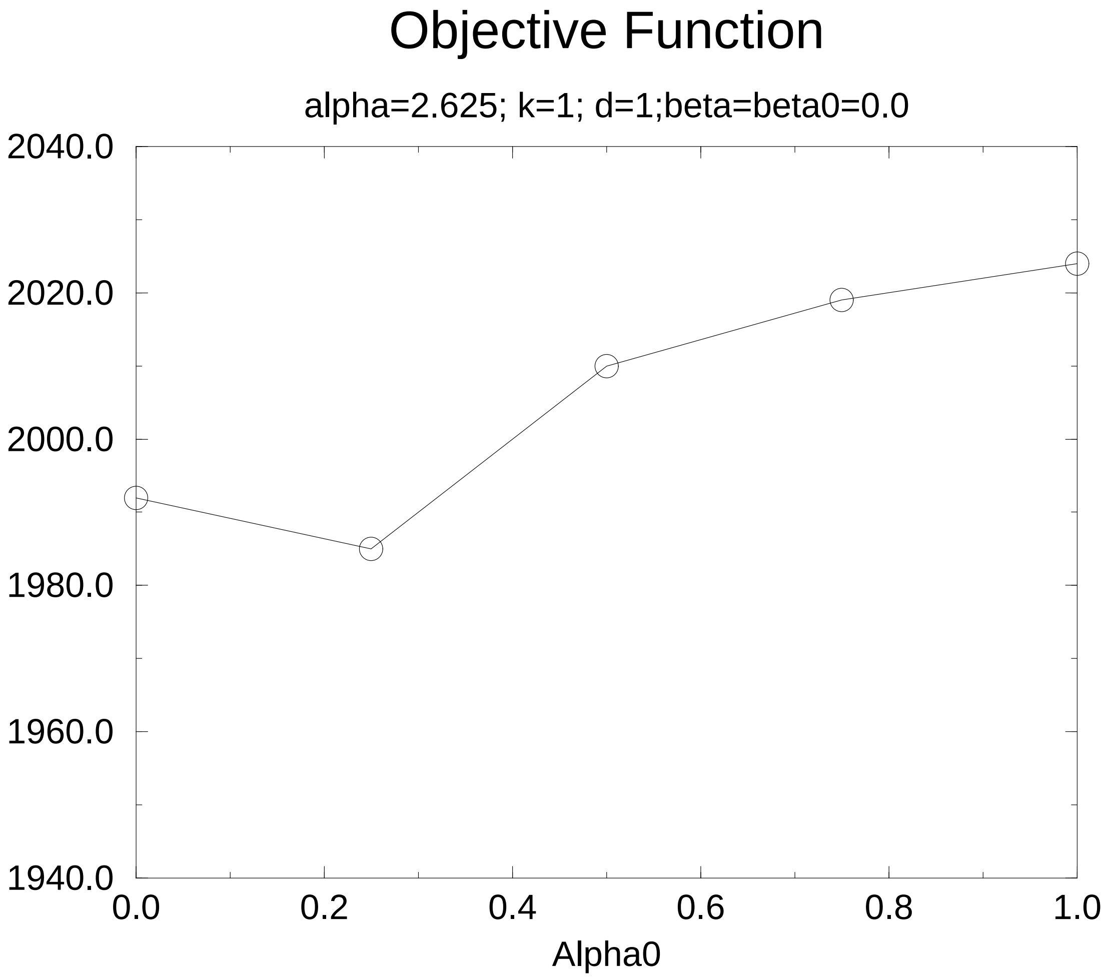


Figure 7.14: The objective value as a function of α_0. Rest of the parameters are fixed at the optimal solution.

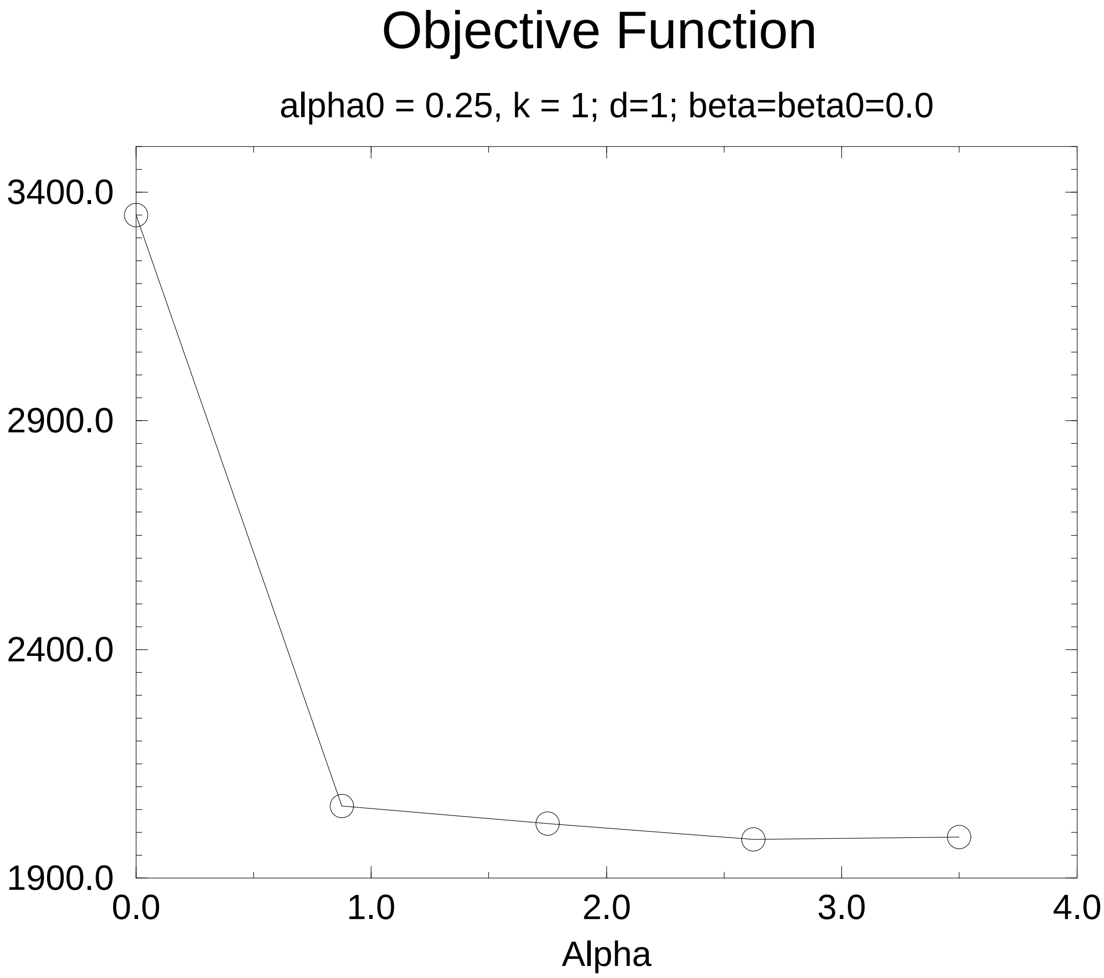


Figure 7.15: The objective function as a function of the decay constant α. Rest of the parameters are fixed at the optimal solution.

2. Number of permutations, K, for creating the empirical null distribution: $K = 1000$.

3. Significance level of the test: $\epsilon = 0.05$.

4. Number of bootstrap repetitions, T, for computing the reject rate of the test: $T = 100$.

5. The bootstrap samples are sampled (with replacement) from a pool of size $N_b = 100$.

6. The character-to-character distance, $\delta(x, y)$, used is the Hamming distance.

7. The set-to-set distance, $\rho(X, Y)$, used is the mean nearest-neighbor distance.

The above test is conducted on 'e's. The test did not reject the null hypothesis that the samples are from the same population for a sample size of 10. That is, the reject rate is lower than 5%. For the sample size of 20, 46% percent of the time the test rejected the null hypothesis. For sample size of 60, the null hypothesis is rejected 100% of the times.

7.9 Discussion

In the previous section we used a two sample permutation procedure to test the null hypothesis that the sample of real degraded characters and the sample generated by the estimated degradation model are from the same underlying population. We found that when the sample size is forty, the test procedure rejects the null hypothesis.

In fact, in a two sample test, if one of the samples is from a distribution that is even slightly different from the second sample's distribution, the statistical testing procedure will be able to reject the null hypothesis that the samples are from the same underlying population if the sample size is large enough.

Since we know that any model of a real process, with very high likelihood, is just an approximation to the real process, the samples generated from the model will be different from the real samples. Thus, any validation procedure will be able to distinguish the real and synthetic samples if the sample sizes are large enough. In other words, it is futile to test the equality of the distribution of the synthetic samples and the real samples; they will be always proved to be unequal if a sample size that is large enough is used. Even if some other validation procedure is used, for example any method based on comparison of confusion matrices, the equality test is always going to give a negative result when the sample size is made large enough.

The next question is: How can one use the validation procedure in practice if the models are always going to be proved incorrect?

There are two ways one can approach this impasse. First method is to use the validation procedure for comparing two models. That is, given two models, and a fixed sample size, we use the validation procedure to quantitatively judge which of the two synthetic samples is closer to the real sample.

The second method is to use an approximate hypothesis test instead. That is, given two samples, we will use a test to say whether the two distributions are within ϵ instead of being equal.

In the following subsections we discuss the two methods in more detail.

7.9.1 Comparing Two models

Let us say there are two document degradation models M_1 and M_2. The problem is to find the model that is closer to the real process. We know that if the sample size N of the synthetic samples and the real samples is increased, after a certain point the validation procedure will start rejecting both the models. However, we will now give a procedure that will allow a researcher to decide which model is closer to reality for a fixed sample size N.

1. Fix the sample size N.

2. We are given the real sample D of size N.

3. Generate synthetic samples S_1 and S_2 of size N using the models M_1 and M_2 respectively.

4. Conduct the two sample validation test using the real sample D and the synthetic sample S_1. Let the associated P-value be p_1.

5. Conduct the two sample validation test using the real sample D and the synthetic sample S_2. Let the associated P-value be p_2.

6. If $p_1 > p_2$, the model M_1 is closer to the real process for a sample size of N. Otherwise the model M_2 is closer.

Thus the above procedure allows a researcher to choose between models.

When we were choosing between parameter setting for a fixed model, we could use the power funciton to arrive at the best parameter sitting. However, two different models have different parameter space and hence they cannot compared using power functions. The P-value forms our means of comparing the models on a common basis.

7.9.2 Approximate Tests

Let $X = \{x_1, x_2, \ldots, x_n\}$ be a sample of real 'e's and let $Y = \{y_1, y_2, \ldots, y_n\}$ be sample of synthetic 'e's.

Let us say we are given a distance function $\rho(X, Y)$ between the sets X and Y. Since X and Y are random variables, $\rho(X, Y)$ is a random variable. In section 6.2 we described a method for computing the null distribution of $\rho(X, Y)$ which is the distribution of $\rho(X, Y)$ under the assumption that X and Y come from same underlying population. Now, instead of shuffling the two samples X and Y, as was done in section 6.2, and then computing the null distributions, we will proceed slightly differently. We will compute two empirical distributions. The first distribution is generated using the same procedure but instead of using samples from both X and Y, data only from X is used. That is, the set X is split into X_1 and X_2 and the permutation procedure is applied J times to generate the empirical histogram. Let the empirical null distribution obtained only using X be $F_X = (\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_L)$ and similarly let the null distribution obtained only using Y be $F_Y = (\hat{q}_1, \hat{q}_2, \ldots, \hat{q}_L)$. These empirical distributions are computed by breaking the real line into L intervals, counting the number of elements that fall into each bin, and then dividing each count by the total number of trials J.

We will use the Bayes error as a distance measure between the two density functions F_X and F_Y. Bayes error is defined by:

$$BE = \frac{1}{2} \sum_{l=1}^{L} \min(p_l, q_l) \tag{7.8}$$

Notice that if the distributions X and Y are close, BE will be close to 0.5. If they differ, the Bayes error will decrease. Thus the Bayes error is bounded: $0 \leq BE \leq 0.5$. We will test the hypothesis $BE < \epsilon$. Now, in order to use the Bayes error as a test statistic, we require its null distribution. To test this hypothesis at the α significance level we need to determine β such that

$$\alpha = P(\sum_{l=1}^{L} \min(\hat{q}_l, \hat{p}_l) > \beta | \sum_{l=1}^{L} \min(q_l, p_l) < \epsilon) \tag{7.9}$$

and we will reject the hypothesis if the test statistic $T = \sum_{l=1}^{L} \min(\hat{q}_l, \hat{p}_l)$ is greater than β.

Here ϵ is the user-specified threshold on the Bayes error, and β is the location such that the area under the null distribution and to the right of β is α, the significance level of the test. To compute the null distribution we proceed as follows:

1. Choose q_l, p_l at random such that they satisfy the constraint $\sum_{l=1}^{L} \min(q_l, p_l) < \epsilon$.

2. Generate J samples according to multinomials $(p_1, \ldots, p_L)$ and $(q_1, \ldots, q_L)$, where J is the number of trials that were performed to estimate F_X and F_Y. Then use the samples to estimate the empirical probabilities $\hat{q}_l, \hat{p}_l$.

3. Compute the statistic $T = \sum_{l=1}^{L} \min(\hat{q}_l, \hat{p}_l)$.

4. Repeat steps 1 through 3 K times and compute T_k, $k = 1, \ldots, K$.

5. The normalized frequency distribution of T_k is the null distribution.

Notice that the Kolmogorov test is a special case of the Bayes error distance. In Kolmogorov test, the maximum distance between the two binned cumulative distributions is used as the test statistic. The maximum distance is used as a test statistic because the corresponding null distribution is known (derived theoretically). In the case of Bayes error, however, the null distribution is unknown. Thus we create an empirical null distribution using the data.

Chapter 8

DISCUSSION

In this chapter we will discuss many issues and concerns that are related to the thesis topic.

8.1 The Scientific Method

The *scientific method* dictates that any explanation of a particular phenomenon should have the following two components. First, there should be a model for the phenomenon; and second, an experiment should be conducted to verify if the model is correct. The proposed model itself should be such that experiments can be designed to test the model's effectiveness in explaining the data. The model should then be used to predict the consequences of a hypothetical experiment. If the model-based predictions agree with the empirical data gathered by actually conducting the experiment, we declare that the model is a scientific explanation for the phenomenon.

If one can conduct an experiment and show that the model-based prediction is not close to the experimental results, we say that the model is not valid, or not a good model. Furthermore, if we have two models and the first model predicts the outcomes of an experiment more accurately than the second model, we declare that the first model is a better model or explanation of the phenomenon. Thus this hypothesize-and-test loop allows us to weed out models and compare models against each other.

An important point to note is that if the model accurately predicts the results of an experiment, we can only say that currently we do not have evidence against the model. This does not mean that the model is correct. For instance, it might so happen that although the model predicts the results of one experiment accurately, it does not predict the results of another experiment accurately. Thus a model is correct until proven otherwise. In physics, Newtonian mechanics was considered a correct model for motion until it was noticed that the behavior of bodies at high velocity could not be explained by the theory. Relativity theory is the currently accepted theory for motion.

So what about mathematics? Mathematics is also a science, albeit of a different type. In mathematics there are axioms that are assumed correct without question. Various relations are then derived from the axioms using the symbol manipulation rules of a particular mathematical system. For instance, in group theory, all the results are derived from the definition of groups. Since the definitions and axioms have nothing to do with natural phenomenon, the results of the subsequent symbol manipulations have nothing to do with the natural world. Thus, mathematical results stand on their own, and no experiment, in the sense of measuring something, needs to be done to verify the results.

The Webster's dictionary gives the following definition for 'scientific method': "principles and procedures for the systematic pursuit of knowledge involving the recognition and formulation of a problem, the collection of data through observation and experiment, and the formulation and testing of hypotheses."

8.1.1 Electrical Engineering and the Scientific Method

The scientific method, which was briefly summarized in the previous section, does not restrict its applicability to natural phenomena alone. Let us say an engineer designs a new OCR algorithm that is supposed to work for certain class of scanned documents. Here one can model the class of documents, the printing and scanning processes, and finally the algorithm. The designed experiment, once performed, indicates whether or not the algorithm works according to our prediction. If it does not, the engineer individually checks the object model, the imaging model, and the algorithm itself. The algorithm is improved by this process of scientific analysis. In fact, the Webster's dictionary defines the word 'engineering' as: "the application of science and mathematics by which the properties of matter and the sources of energy in nature are made useful to people in structures, machines, products, systems, and processes."

On occasions when an algorithm shows potential for monetary profit, or when there are time constraints, scientific analysis may be bypassed – the rigorous scientific method is not applied to the designed algorithm to check if the claimed features of the algorithm agree with the experimental findings. However, the engineering tradition does not encourage such practice. In fact, many research areas such as document understanding have hit a road block due to such practices.

8.1.2 Computer Science and Scientific Method

There has been much debate whether computer science is a scientific discipline or an engineering discipline. It is quite amazing that well-respected computer scientists believe that computer science *is not* about validating and comparing models. For instance, Juris Hartmanis, a physicist-turned-computer-scientist, who received the 1994 Turing award for developing the concept of computational complexity, says [Har94b]:

> "...Computer science deals with information, its creation and processing, and with the systems that perform it, much of which is not directly restrained and governed by physical laws. Thus computer science is laying the foundations and developing the real search paradigms and scientific methods for the exploration of the world of information and intellectual processes that are not directly governed by physical laws. This is what sets it apart from the other sciences ..."

We disagree with Hartmanis. The information that is gathered (using various types of sensors) in document understanding, computer vision, et cetera is clearly at the mercy of the imaging process. The imaging process itself is governed by the physical laws and thus we are forced to model the sensed data, which is not created by a computer, and which can be noisy. This in turn leads to various competing models for the sensing process and various optimal algorithms for each model. However, Hartmanis thinks otherwise:

> "Thinking about the previously mentioned (and other) theoretical work in computer science, one is lead to the very clear conclusion that theories do not compete with each other as to which better explains the fundamental nature of information. Nor are new theories developed to reconcile theory with phenomena, as in physics. In computer science there is no history of critical experiments experiments that decide between the validity of various theories, as there are in physical sciences."

Our impression is that Hartmanis has focussed on a narrow area of computer science that is referred to as 'theoretical computer science' (mainly automata theory, computability and complexity theory). True, these are mathematical fields that start with definitions and abstract mathematical models of computers, languages and algorithms and use the rules of mathematics to arrive at relationships and properties of the abstract model being studied. Research in these areas is as scientific as a mathematician's research. However, all areas of computer science that have to sense the natural world (most artificial intelligence topics), are also scientific disciplines, but like physics and other natural sciences. Fortunately many others feel similarly [H+95]

8.1.3 Performance Evaluation and Characterization

Now that we have talked about the scientific method, we can easily see how performance evaluation forms a part of scientific investigation. In the context of computer vision, suppose we are given an algorithm that is claimed to perform a certain task, on a specified population of images and to a specified accuracy level. The algorithm typically has various algorithm parameters, and the image population too can be specified by certain image population parameters. We have two problems at hand. The first problem is to verify that the algorithm works as claimed over the entire population of images it is supposed to work on. The second problem is to provide the algorithm users some way of predicting the results of the algorithm on a given set of data.

8.2 Automatic Groundtruth: Making the Scientific Method Practical

Having talked about the scientific method, it is important to realize that in many cases it may not be possible to apply the scientific method. For example, in the case of OCR, to check whether an OCR algorithm actually performs at the advertised accuracy level, we have to know what is the correct answer (that is, the groundtruth). Furthermore, since the population on which the algorithm is applied is large, a large database of scanned documents with groundtruth is required. Such large databases, until now, were not feasible since they required a person to manually annotate a document image with groundtruth information. Since manual groundtruth is prone to errors, expensive, laborious, time consuming and a health hazard, the project was not attempted.

However, this thesis has changed the situation. Our closed loop procedure allows us to automatically generate groundtruth information for scanned document images. Not only that, it works for any language. Thus, now it is possible for the research community to generate large databases with thousands of scanned characters with groundtruth. In fact, we have already produced such a database with 62000 characters.

In contrast, the photogrammetry and the medical imaging communities still manually enter groundtruth data. Whereas a team of about seven persons manually collected groundtruth for seventy aerial images in over two years, our automatic model-based groundtruth procedure generated groundtruth for 33 document images in two and half hours, without any human intervention. There is no reason why our procedure cannot be used to groundtruth aerial images.

It is interesting to note that all the components of the registration and groundtruth generation algorithm – scanners, printers, document typesetting languages, regression and model-based matching algorithms – have all existed almost as long as researchers have been working on OCR.

8.3 Degradation Models

So if we have a way of generating large size real data sets with groundtruth, one can question why is there any need for a generative model? Why not let a large sample of degraded documents represent the degradation model instead of a functional or algorithmic model? That is, why not set the parameters of the OCR algorithm by finding the parameter setting that minimizes the classification error over the entire database of real images?

As usual, there are pros and cons. From a practical point of view, if a product is to be delivered in a short time, one might be better off just finding the parameter setting that minimizes the classification error. In such a situation, only obtaining the optimal accuracy matters, and an explanation of why the accuracy is what it is does not matter. Usually the search space in this case is very large and finding the optimal solution is time consuming.

However, when the optimal performance rates do not meet the requirements, the OCR algorithm designer asks the question why? An explanation is sought as to what went wrong where. The components of the systems are investigated with the hope of possible improvements.

In a model-based approach, one can derive the optimal solution by assuming an underlying model. One can say a solution is the optimal and no further improvements can be made – e.g. the Kalman filter is optimal for Gaussian noise. Furthermore, using a noise model allows a designer to propagate the error through the various components and predict what the final performance will be even before implementing the system and running experiments.

8.4 Validating What?

There is a proposal by another group that the way to validate a model is by testing the difference in error patterns produced by OCRing a real image and OCRing a synthetically degraded image. If the errors are 'similar' under some distance measure, there is not enough statistical evidence to invalidate the model. Else, if there *is* substantial difference, the model is invalid.

We, on the other hand, work much earlier on in the whole OCR process. Our method creates a null distribution of a test statistic that is a distance function between two sets of degraded characters. One set of degraded characters is real, the other is synthetic. If the p-value associated with the test is less than the significance level set by the user, the null hypothesis that the two samples come from the same underlying distribution is rejected. Otherwise, we say that there is not enough evidence to invalidate the model.

There are many problems of using an OCR in the procedure for validating a degradation model. First, it is important to remember that what is being validated is the model and the OCR system together and not just the model. Similarly, in our methodology, what is being validated is the model and the distance function together and not just the degradation process. Thus, using an OCR for validation implies that the validation results become a function of the OCR package and all its parameters. A typical OCR is quite complicated, and so for each parameter setting of the OCR package, one gets a different validation result. In contrast, in our methodology the distance functions are much simpler entities than OCR systems. Besides, we provide a rigorous statistical procedure based on the power function that can decide which distance function gives a more powerful test. On the other hand, computing a power function for an OCR-based validation scheme would be, to say the least. quite time consuming.

Furthermore, OCR packages usually perform noise removal, skew compensation, etc. as part of a preprocessing stage. Thus, something is awkward in the OCR-based evaluation – it is the degradation that is getting validated but a component in the process is getting rid of it. If for example, only rotation degradation is being validated, a validation scheme that uses an OCR with skew correction will never be able to detect a rotation degradation because it will get undone by the preprocessing step.

Chapter 9

CONCLUSIONS

Two document degradation models are proposed. The first model is applied on binary images at a page level and it accounts for local degradations that occur while printing, photocopying and scanning documents. The model is motivated by studying the spatial properties of the degradations and using the morphological operations that best model such spatial distortions. The reason for representing the distortions using morphological operations is simple: since most noise removal and restoration algorithms today are morphological, it is best to have a degradation model that fits into this framework. In particular, the local degradations are modeled by making the pixels flip from zero to one and vice-versa according to a probability that depends inversely on the distance between the pixel and the boundary of the character. The correlation due to the optical point spread function is modeled by a morphological closing operation. The model is parametrized and thus it can be used to synthetically generate a large number of degraded documents. Moreover, since the input to the model is a binary image, binary document images in any language can be degraded using this model. The implemented software takes approximately two minutes on a SUN sparc 10 to degrade a 3300×2550 binary image.

The second model accounts for the perspective and illumination distortions that occur while photocopying or scanning a thick, bound book. The model is based on the physics of imaging, and takes into account the optics of the imaging system, the shape of the book surface and its reflectance properties. As a result, it can model the defocus, illumination change and the gradual skew in the imaged document. This model is also parametrized and allows us to synthetically generate distorted document images. Furthermore, a model such as this allows researchers to 'undo' the perspective and illumination distortions. This model was also simulated and on a sun machine.

A methodology for producing groundtruth information for the synthetically degraded documents (the identity, location, bounding box, and font type of individual characters) is described. The general method is to (i) start with a document in a symbolic form, where the text, formatting and layout is known without ambiguity, (ii) create the ideal bit map, (iii) create the ideal groundtruth from the knowledge of the typesetting language, and finally (iv) degrade the document image using the model. Since a document can be typeset in various styles and formats and degraded using this model, the methodology gives us access to a vast variety of degraded documents. In this thesis the text is represented using ASCII and the layout is typeset using LaTeX. The groundtruth for synthetically degraded document images is generated effortlessly, and at no cost. Since documents of various types and in various languages can be typeset using LaTeX or any similar typesetting language, groundtruth for text in any language can be generated using our methodology. In fact, we used the model and the groundtruth method to generate, synthetically degraded music, mathematics, Arabic, Hindi and engineering linedrawing document images with the corresponding groundtruth. An implementation of this methodology takes less than a second on a SUN sparc 10 to generate groundtruth for a synthetically degraded document page of size 3300×2550 with approximately 2000 characters.

Accurate character groundtruth information for real document images has always been difficult to obtain. In this thesis a methodology is described to generate highly accurate groundtruth for real document images. The steps of the procedure are: (i) generate an ideal document image with the associated ideal groundtruth, (ii) print the ideal document image, (ii) scan the printed page, (iii) find a transformation that registers the ideal document image to the real document image, and finally (iv) transform the ideal groundtruth using the estimated transformation to get the groundtruth for the real document image. The procedure takes about five minutes on a SUN spac 10, for an image of size 3300×2550 and with about 2000 characters per page. Again, this methodology is independent of the language in which the text is written, and so can be used to generate groundtruth for real documents in any language. A database of 33 real document images with a total of 6200 characters, and their corresponding groundtruth is created using this methodology. Creation of such accurate databases of real character groundtruth was not possible until now.

Two methods for estimating the parameters of the degradation model is described and implemented. Thus, given a real document and its corresponding ideal document, the parameters of the model that generates 'similar' looking degraded documents from ideal documents can be estimated. The first method samples the six-dimensional parameter space coarsely and chooses the parameter value that yields the lowest objective function value. The second method starts with an initial estimate and then searches for a minimum along each parameter one at a time. The process is stopped after three iterations. The estimation procedure is extremely useful for creating large database of synthetically degraded documents from a small sample of real documents. This obviates the need for manual printing, photocopying and scanning of documents on a large scale.

The degradation model validation problem is posed as a two-sample statistical hypothesis testing problem. A non-parametric permutation test is adopted for this purpose. The user specifies a test statistic, which is essentially a distance function on the two sets of degraded characters. The null distribution, which is the distribution of the test statistic under the hypothesis that the two samples come from the same underlying population, is created using a permutation procedure. The p-value corresponding to the test statistic associated with the the two sets is computed and compared with the user-specified significance level to reject or accept the null hypothesis. This procedure and several robust variants are implemented. The distance functions used for the test statistic are somewhat heuristic and questionable. This issue is addressed by using the power functions – a standard statistical device – to find the distance function that is more powerful. The local degradation model passes the validation test when the sample size is small but rejects it when sample size is increased. This is so because any model of a real world process is an approximation and thus will not pass the test if the sample size is increased. A method of conducting approximate tests instead of equality tests is also described.

Another way of using the validation procedure is for choosing between models. After the validation procedure is run, a p-value is obtained. Thus if two different models are tested on the same real data, each validation procedure gives rise to a p-value for each model. The model whose associated p-value is larger is in closer agreement with the real data and thus should be preferred.

A summary of the major contributions in this thesis now follows.

9.1 Summary of Contributions

The main contributions that are presented in this thesis are:

1. A model for the local degradations that are introduced while printing, photocopying and scanning a document. This model is motivated by studying the spatial properties of the degradations and the natural morphological operations that can represent such spatial characteristics. Documents of various types and in various languages can be degraded at user-specified levels.

2. A physical model that accounts for the perspective and illumination distortions that occur while photocopying or scanning a thick, bound book. The model takes into account the optics of the imaging system, the shape of the book surface, defocus, illumination and reflectance properties of the surfaces. This model is also parametrized and allows us to synthetically generate distorted document images.

3. A methodology for automatically generating groundtruth for synthetically degraded documents. The method uses the original symbolic text and the typesetting information and produces the groundtruth for the synthetically degraded document image in any language.

4. A methodology for automatically generating groundtruth for *real* degraded documents. The method registers the ideal document image to the real document image and then transforms the ideal groundtruth using the estimated transform such that the groundtruth overlays the real document image very accurately. A database of 33 real document images with 6200 characters and the corresponding groundtruth information is created using this methodology. Now researchers can evaluate their OCR systems at character level on large databases of real documents. This was not possible until now.

5. A methodology for degradation model parameter estimation is given. Given a sample of real images, the nonparametric estimation procedure finds the parameter values that make the simulated samples closest to the real ones. Thus, from the user's point of view, a person having a small sample of real images can create a large sample by first estimating the parameters of the model and then synthetically generating a large data set.

6. A methodology for degradation model validation. Given a sample of real documents, and a sample of synthetic documents, this nonparametric hypothesis testing procedure tests the null hypothesis whether or not the two populations come from the same underlying distribution. The local degradation model passed the validation procedure for small sample sizes but rejects it when the sample size increases.

7. A method for comparing models is discussed. After the model parameters are estimated, the validation procedure can be run to obtain the associated p-values. The model with higher p-value is better.

8. A methodology based on the power function that allows to optimize the validation procedure is described and implemented. The validation procedure has variables such as the choice of distance functions. This power function procedure allows us to select the distance function that makes the validation procedure more powerful (in a statistical sense).

9. All the software and the real character groundtruth data sets will be made available to researchers on a CD-ROM.

9.2 Future Research

This work has opened up many new areas of research that need to be explored.

- Since there are two models, experiments need to be conducted to compare both models and see which one is closer to the real degradations.

- Our parameter estimation methodology requires us to have the ideal image. In many cases this is not possible. How does one approach the problem then? One way might be to collect symbols from the image and make an 'ideal font' by averaging a large number of instances of the same symbol.

- A bootstrap procedure for estimating the covariance of the estimated parameters. Other techniques could be compared.

- Efficient search/optimization procedures for estimating the parameters of the models would help. The objective function is not continuous and differentiable and so currently we have computed the estimate by evaluating the objective function at sampled locations in the parameter space.

- Since there is another group having another validation procedure, both validation procedures can be be compared to find which one is more sensitive.

- A method for validating the perspective distortion model.

BIBLIOGRAPHY

[Ame82] The American Society for Mechanical Engineers, New York. *ANSI Y14.5M, Dimensioning and Tolerancing*, 1982.

[And84] T. W. Anderson. *Introduction to Multivariate Statistical Analysis.* John Wiley & Sons, Inc., New York, NY, 1984.

[Arn90] S. F. Arnold. *Mathematical Statistics.* Prentice-Hall, New Jersey, 1990.

[Bai90] H. Baird. Document image defect models. In *Proc. of IAPR Workshop on Syntactic and Structural Pattern Recognition*, pages 38–46, Murray Hill, NJ, June 1990.

[Bai92] H. S. Baird. Document image defect models. In *Structured Document Image Analysis.* Springer-Verlag, New York, 1992.

[Bai93] H. Baird. Calibration of document image defect models. In *Proc. of Second Annual Symposium on Document Analysis and Information Retrieval*, pages 1–16, Las Vegas, Nevada, April 1993.

[BL] B. Brown and J. Lovato. RANLIB.C: Library of C routines for random number generation. Technical report, Department of Biomathematics, The University of Texas, Houston, TX 77030. anonymous ftp: odin.mda.uth.tmc.edu:/pub/unix/ranlib.c.tar.Z.

[BLR94] B. Brown, J. Lovato, and K. Russell. DCDFLIB: A library of C routines for cumulative distribution functions, inverses, and other parameters. Technical report, Department of Biomathematics, The University of Texas, Houston, TX 77030, 1994. anonymous ftp: odin.mda.uth.tmc.edu:/pub/unix/dcdflib.c.tar.Z.

[Bor86] G. Borgerfors. Distance transforms in digital images. *Computer Vision, Graphics, and Image Processing*, 34:344–371, 1986.

[CB90] G. Casella and R. L. Berger. *Statistical Inference.* Wadsworth, CA, 1990.

[CF90] R. G. Casey and D. R. Ferguson. Intelligent forms processing. *IBM Systems Journal*, 29(3):435–50, 1990.

[DAR92] DARPA. *Proceedings of DARPA Workshop on Document Understanding.* Palo Alto, CA, 1992.

[DR93] D. S. Doermann and A. Rosenfeld. The processing of form documents. In *Proc. of Int. Conf. on Document Analysis and Recognition*, pages 497–501, Tsukuba, Japan, October 1993.

[ET93] B. Efron and R. J. Tibshirani. *An introduction to the bootstrap.* Chapman and Hall, New York, 1993.

[FvDFH90] J.D. Foley, A. van Dam, S.K. Feiner, and J.F. Hughes. *Computer Graphics.* Addison-Wesley, Reading, Massachesetts, 1990.

[Goo94] P. Good. *Permutation Tests: A Practical Guide to Resampling Methods for Testing Hypotheses.* Springer-Verlag, New York, 1994.

[H+95] J. Hartmanis et al. Computing surveys symposium on computational complexity and the nature of computer science. *ACM Computing Surveys*, 1995. A special issue with fourteen articles.

[Har89] R.M. Haralick. Performance assessment of near perfect machines. *Journal of machine vision and applications*, 2:1–16, 1989.

[Har94a] R. M. Haralick. Propagating covariance in computer vision. In *Proc. of IAPR Int. Conf. on Pattern Recognition*, pages 493–498, Israel, October 1994.

[Har94b] J. Hartmanis. Turing award lecture: on computational complexity and the nature of computer science. *Communications of the ACM*, 1994.

[Hor86] B.K.P. Horn. *Robot Vision.* The MIT Press, Cambridge, MA, 1986.

[Hou83] H. S. Hou. *Digital Document Processing.* John Wiley, New York, 1983.

[HP+] R. M. Haralick, I. Phillips, et al. UW-CDROM-I.

[HS92] R.M. Haralick and L.G. Shapiro. *Computer and Robot Vision (vols. 1 and 2).* Addison-Wesley Publishing Co., Inc., Reading, Massachusetts, 1992.

[KH95] T. Kanungo and R. M. Haralick. Morphological degradation parameter estimation. In *SPIE Proceedings*, San Jose, CA, February 1995.

[KHB+94] T. Kanungo, R. M. Haralick, H. S. Baird, W. Stuetzle, and D. Madigan. Document degradation models: Parameter estimation and model validation. In *Proc. of Int. Workshop on Machine Vision Applications*, Kawasaki, Japan, December 1994.

[KHP93] T. Kanungo, R. M. Haralick, and I. Phillips. Global and local document degradation models. In *Proc. of Second International Conference on Document Analysis and Recognition*, pages 730–734, Tsukuba, Japan, October 1993.

[KHP94] T. Kanungo, R. M. Haralick, and I. Phillips. Non-linear local and global document degradation models. *Int. Journal of Imaging Systems and Technology*, 5(4), 1994.

[KJPH93a] T. Kanungo, M. Y. Jaisimha, J. Palmer, and R. M. Haralick. A quantitative methodology for analyzing the performance of detection algorithms. In *IEEE International Conference on Computer Vision*, Berlin, Germany, 1993.

[KJPH93b] T. Kanungo, M.Y. Jaisimha, J. Palmer, and R.M. Haralick. A methodology for quantitative performance evaluation of detection algorithms. *IEEE Transactions on Image Processing (to appear)*, 1993.

[Knu88] D. E. Knuth. *TEX: the program.* Addison-Wesley, Reading, Mass., 1988.

[Koc87] K. Koch. *Parameter Estimation and Hypothesis Testing in Linear Models.* Springer-Verlag, New York, NY, 1987.

[Lam86] L. Lamport. *LATEX: a document preparation system.* Addison-Wesley, Reading, Mass., 1986.

[LLT94] Y. Li, D. Lopresti, and A. Tomkins. Validation of document defect models for optical character recognition. In *Proc. of Third Annual Symposium on Document Analysis and Information Retrieval*, pages 137–150, Las Vegas, Nevada, April 1994.

[LLT96] Y. Li, D. Lopresti, and A. Tomkins. Validation of document defect models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, January 1996.

[Nag94] G. Nagy. Validation of ocr data sets. In *Proc. of Third Annual Symposium on Document Analysis and Information Retrieval*, pages 127–135, Las Vegas, Nevada, April 1994.

[Pen88] A.P. Pentland. A new sense of depth of field. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pages 523–531, 1988.

[PFTV90] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C.* Cambridge University Press, New York, NY, 1990.

[SG88] M. Subbarao and N. Gurumoorthy. Depth recovery from blurred edges. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pages 498–503, Ann Arbor, MI, 1988.

[Str88] G. Strang. *Linear Algebra and its applicatins.* Harcourt Brace Jovanovich, 1988.

[SW86] B. Smith and J. Willington. *Initial Graphics Exchange Specification (IGES), Version 3.0.* U.S. Department of Commerce, National Institution of Standards, NBSIR 86-3359, 1986.

[V$^+$90] P. Vojta et al. XDVI Software, 1990.

[Wol90] G. Wolberg. *Digital Image Warping.* IEEE Press, 1990.

Appendix A

NULL DISTRIBUTION FOR GAUSSIAN POPULATIONS

In this appendix we compute the null distributions of two set distances $\rho(X, Y)$ when x_i and y_i are Gaussian distributed. We show that when x and y are each Gaussian distributed with a known variance σ^2, the two distance functions considered are χ^2 distributed under the null hypothesis. Such closed form solutions for the null distributions are possible only when the underlying distributions are known *a priori*. However, this is not the case in general – the Gaussian assumptions might be appropriate in some settings but could be completely wrong in other settings. Thus, the non-parametric permutation method described in chapter 6 is a much better approach to computing the null distributions when the forms of the sample distributions are not known. Nevertheless, for the purpose of validating the software and algorithm for computing the empirical null distribution, the Gaussian case is very useful since it allows us to compare the empirical distributions against known (theoretically computed) distributions.

A.1 Inter Cluster Mean Distance

Let $X = \{x_1, x_2, \ldots, x_N\}$ be a set such that $x_i \in R$ and $x_i \sim N(\mu_X, \sigma^2)$. Similarly, let $Y = \{y_1, y_2, \ldots, y_N\}$ be a set such that $y_i \in R$ and $y_i \sim N(\mu_Y, \sigma^2)$. The problem is to test the null hypothesis that $\mu_X = \mu_Y$, when σ^2 is known.

Now, we know that

$$\hat{\mu}_X = \frac{1}{N}\sum_{i=1}^{N} x_i \sim N(\mu_X, \sigma^2/N) \tag{A.1}$$

$$\hat{\mu}_Y = \frac{1}{N}\sum_{i=1}^{N} y_i \sim N(\mu_Y, \sigma^2/N) \ . \tag{A.2}$$

Therefore,

$$\hat{\mu}_X - \hat{\mu}_Y \sim N(\mu_X - \mu_Y, 2\sigma^2/N) \tag{A.3}$$

and

$$\sqrt{N/2}(\hat{\mu}_X - \hat{\mu}_Y)/\sigma \sim N(\mu_X - \mu_Y, 1) \ . \tag{A.4}$$

Now, let

$$t = \rho(X, Y) = \frac{N}{2\sigma^2}(\hat{\mu}_X - \hat{\mu}_Y)^2.$$

Thus under the null hypothesis that $\mu_X = \mu_Y$, we have

$$t = \rho(X, Y) \sim \chi_1^2 \ . \tag{A.5}$$

Thus, instead of empirically computing the distributions as described in chapter 6 we can use the above analytic form of the distribution to accept or reject the null hypothesis. Moreover, we see that the empirical method has reduced to a standard statistical technique when the underlying distribution is known to be Gaussian.

A.2 Likelihood distance

In the previous section we picked a particular distance function $\rho(X, Y)$ and showed that its null distribution is χ_1^2. In this section we pick a distance function based on the likelihood function of the data. It turns out that this distance function is the same as the one used in the previous section.

Let $X = \{x_1, x_2, \ldots, x_N\}$, where $x_i \in R$, and $x_i \sim N(\mu_X, \sigma^2)$. Similarly, let $Y = \{y_1, y_2, \ldots, y_N\}$, where $y_i \in R$, and $y_i \sim N(\mu_Y, \sigma^2)$. The problem is to test the null hypothesis that $\mu_X = \mu_Y = \mu$.

Let $\rho_Y(X)$ denote the distance of set X from set Y. Here we use a function of the likelihood for ρ.

$$\rho_X(Y) = f(P(y_1, \ldots, y_N | x_1, \ldots, x_N, \sigma)) \tag{A.6}$$

$$\rho_Y(X) = f(P(x_1, \ldots, x_N | y_1, \ldots, y_N, \sigma)) \ . \tag{A.7}$$

In general, the above distances need not be symmetric in X and Y. Hence, we also consider symmetric distances of the form

$$\rho(X, Y) = f(P(y_1, \ldots, y_N | x_1, \ldots, x_N, \sigma) P(x_1, \ldots, x_N | y_1, \ldots, y_N, \sigma)) \ . \tag{A.8}$$

We can also consider the right hand side in the equation above divided by $\log \max_\mu P(x_1,\ldots,x_N,y_1,\ldots,y_N|\mu,\sigma)$. That is,

$$\rho(X,Y) = \log\left(\frac{P(y_1,\ldots,y_N|x_1,\ldots,x_N,\sigma)P(x_1,\ldots,x_N|y_1,\ldots,y_N,\sigma)}{\max_\mu P(x_1,\ldots,x_N,y_1,\ldots,y_N|\mu,\sigma)}\right). \quad \text{(A.9)}$$

We can use the standard rules of probability theory to manipulate the above equation as follows.

$$\begin{aligned} & P(y_1,\ldots,y_N|x_1,\ldots,x_N,\sigma) \\ = & \int_{-\infty}^{\infty} P(\mu,y_1,\ldots,y_N|x_1,\ldots,x_N,\sigma)d\mu \\ = & \int_{-\infty}^{\infty} \frac{P(y_1,\ldots,y_N,x_1,\ldots,x_N,\mu,\sigma)}{P(x_1,\ldots,x_N,\sigma)}d\mu \\ = & \int_{-\infty}^{\infty} \frac{P(y_1,\ldots,y_N|x_1,\ldots,x_N,\mu,\sigma)P(x_1,\ldots,x_N,\mu,\sigma)}{P(x_1,\ldots,x_N,\sigma)}d\mu \\ = & \int_{-\infty}^{\infty} \frac{P(y_1,\ldots,y_N|\mu,\sigma)P(x_1,\ldots,x_N|\mu,\sigma)P(\mu,\sigma)}{\int_{-\infty}^{\infty} P(x_1,\ldots,x_N|\lambda,\sigma)P(\lambda,\sigma)d\lambda}d\mu\,. \end{aligned} \quad \text{(A.10)}$$

Now, we make the assumption that μ and σ are independent so that $P(\mu,\sigma) = P(\mu)P(\sigma)$. Furthermore we assume that μ and σ have a uniform prior. Although this implies the prior is improper (since its integral is not equal to 1), the posterior distribution integrates to one. Thus, $P(\mu,\sigma) = P(\mu)P(\sigma) \propto \epsilon$. But the ϵ in the numerator and the denominator of equation (A.10) cancel out and the numerator can now be written as follows.

$$\begin{aligned} & P(y_1,\ldots,y_N|\mu,\sigma)P(x_1,\ldots,x_N|\mu,\sigma)P(\mu,\sigma) \\ = & \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^N e^{-\frac{1}{2\sigma^2}\sum_{i=1}^N (y_i-\mu)^2} \cdot \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^N e^{-\frac{1}{2\sigma^2}\sum_{j=1}^N (x_j-\mu)^2} \\ = & \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^{2N} e^{-\frac{1}{2\sigma^2}\left[\sum_{i=1}^N (y_i-\mu)^2 + \sum_{j=1}^N (x_j-\mu)^2\right]}\,. \end{aligned} \quad \text{(A.11)}$$

Since the denominator is not a function of either μ or $y_1,\ldots,y_N$, it is a constant. The denominator can be computed by integrating out $\mu, y_1,\ldots,y_N$ from the probability density in equation (A.11). Thus,

$$P(y_1,\ldots,y_N|x_1,\ldots,x_N,\sigma)$$

$$= C\int_{-\infty}^{\infty}\left(\frac{1}{\sqrt{2\pi}\sigma}\right)^{2N} e^{-\frac{1}{2\sigma^2}\left[\sum_{i=1}^{N}(y_i-\mu)^2+\sum_{j=1}^{N}(x_j-\mu)^2\right]} \tag{A.12}$$

where the constant of integration C can be found by equating the right hand side to 1. In order to compute the integral, we simplify the exponent inside the integral.

$$\begin{aligned}
&\sum_{i=1}^{N}(y_i-\mu)^2+\sum_{j=1}^{N}(x_j-\mu)^2 \\
&= \sum_{i=1}^{N}(y_i-\bar{y}+\bar{y}-\mu)^2+\sum_{j=1}^{N}(x_i-\bar{x}+\bar{x}-\mu)^2 \\
&= \sum_{i=1}^{N}(y_i-\bar{y})^2+\sum_{i=1}^{N}(y_i-\bar{y})(\bar{y}-\mu)+N(\bar{y}-\mu)^2 \\
&\quad \sum_{j=1}^{N}(x_i-\bar{x})^2+\sum_{j=1}^{N}(x_i-\bar{x})(\bar{x}-\mu)+N(\bar{x}-\mu)^2 \\
&= \sum_{i=1}^{N}(y_i-\bar{y})^2+\sum_{j=1}^{N}(x_i-\bar{x})^2+N(\bar{y}-\mu)^2+N(\bar{x}-\mu)^2\ .
\end{aligned} \tag{A.13}$$

But,

$$\begin{aligned}
(\bar{y}-\mu)^2+(\bar{x}-\mu)^2 &= \bar{x}^2+\bar{y}^2+2\left[\mu^2-2\mu\left(\frac{\bar{x}+\bar{y}}{2}\right)\right] \\
&= \bar{x}^2+\bar{y}^2-2\mu\left(\frac{\bar{x}+\bar{y}}{2}\right)^2 \\
&\quad +2\left[\mu^2-2\mu\left(\frac{\bar{x}+\bar{y}}{2}\right)+\left(\frac{\bar{x}+\bar{y}}{2}\right)^2\right] \\
&= \frac{(\bar{x}^2+\bar{y}^2-2\bar{x}\bar{y})}{2}+2\left[\mu-\left(\frac{\bar{x}+\bar{y}}{2}\right)\right]^2 \\
&= \frac{(\bar{x}-\bar{y})^2}{2}+2\left[\mu-\left(\frac{\bar{x}+\bar{y}}{2}\right)\right]^2\ .
\end{aligned} \tag{A.14}$$

Thus, from equations (A.14) and (A.13)

$$\begin{aligned}
&\sum_{i=1}^{N}(y_i-\mu)^2+\sum_{j=1}^{N}(x_j-\mu)^2 \\
&= \sum_{i=1}^{N}(x_i-\bar{x})^2+\sum_{j=1}^{N}(y_j-\bar{y})^2+\frac{N}{2}(\bar{x}-\bar{y})^2+2N\left(\mu-\left(\frac{\bar{x}+\bar{y}}{2}\right)\right)^2\ .
\end{aligned} \tag{A.15}$$

Also, since

$$\int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi}(\sigma/\sqrt{2N})}e^{-\frac{1}{2\sigma^2/2N}\left(\mu-\frac{\bar{x}+\bar{y}}{2}\right)^2}d\mu = 1, \tag{A.16}$$

we have,

$$P(y_1, \ldots, y_N|x_1, \ldots, x_N, \sigma) \\ = C\left(\frac{1}{\sqrt{2\pi}\sigma}\right)^{2N} \frac{\sqrt{2\pi}\sigma}{\sqrt{2N}} \cdot e^{\frac{1}{2\sigma^2/2N}\left[\sum_{i=1}^N (x_i-\bar{x})^2 + \sum_{j=1}^N (y_j-\bar{y})^2 + \frac{N}{2}(\bar{x}-\bar{y})^2\right]} \quad \text{(A.17)}$$

Now to get the value of C, we proceed as follows.

$$\begin{aligned} 1 &= C\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^{2N} e^{-\frac{1}{2\sigma^2}\left[\sum_{i=1}^N (y_i-\mu)^2 + \sum_{j=1}^N (x_j-\mu)^2\right]} dy_1 \ldots dy_N d\mu \\ &= C\int \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^{N} e^{-\frac{1}{2\sigma^2}\left[\sum_{i=1}^N N(x_i-\bar{x})^2 + N(\mu-\bar{x})^2\right]} d\mu \\ &= C\left(\frac{1}{\sqrt{2\pi}\sigma}\right)^{N} \frac{\sqrt{2\pi}\sigma}{\sqrt{N}} e^{-\frac{1}{2\sigma^2/N}\left[\sum_{i=1}^N (x_i-\bar{x})^2\right]} \end{aligned} \quad \text{(A.18)}$$

Thus, we have computed C to be

$$C = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^{-(N+1)} \sqrt{N} e^{\frac{1}{2\sigma^2/N}\left[\sum_{i=1}^N (x_i-\bar{x})^2\right]} . \quad \text{(A.19)}$$

Finally we now can write the complete conditional density as

$$\begin{aligned} &P(y_1, \ldots, y_N|x_1, \ldots, x_N, \sigma) \\ &= \left[\left(\frac{1}{\sqrt{2\pi}\sigma}\right)^{-(N+1)} \sqrt{N} e^{\frac{1}{2\sigma^2/N}\left[\sum_{i=1}^N (x_i-\bar{x})^2\right]}\right] \\ &\quad \cdot \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^{2N} \frac{\sqrt{2\pi}\sigma}{\sqrt{2N}} \cdot e^{\frac{1}{2\sigma^2/2N}\left[\sum_{i=1}^N (x_i-\bar{x})^2 + \sum_{j=1}^N (y_j-\bar{y})^2 + \frac{N}{2}(\bar{x}-\bar{y})^2\right]} \\ &= \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^{N} \cdot \sqrt{2} \cdot e^{-\frac{1}{2\sigma^2}\left[\sum_{i=1}^N (y_i-\bar{y})^2 + \frac{N}{2}(\bar{x}-\bar{y})^2\right]} . \end{aligned} \quad \text{(A.20)}$$

Thus, we can use the $2\sigma^2$ times the negative exponent of the conditional probability, as given in equation (A.20), as the test statistic $\rho_X(Y)$. Notice that it is not symmetric in X and Y.

$$\begin{aligned} \rho_X(Y) &= f(P(y_1, \ldots, y_N|x_1, \ldots, x_N, \sigma)) & \text{(A.21)} \\ &= -\log P(y_1, \ldots, y_N|x_1, \ldots, x_N, \sigma) + \frac{N}{2}\log(2\pi\sigma^2) - \frac{1}{2}\log(2) & \text{(A.22)} \end{aligned}$$

$$
\begin{aligned}
&= (\sum_{i=1}^{N}(y_i-\bar{y})^2+\frac{N}{2}(\bar{x}-\bar{y})^2 && \text{(A.23)}\\
\rho_Y(X) &= f(P(x_1,\ldots,x_N|y_1,\ldots,y_N,\sigma)) && \text{(A.24)}\\
&= -\log P(x_1,\ldots,x_N|y_1,\ldots,y_N,\sigma)+\frac{N}{2}\log(2\pi\sigma^2)-\frac{1}{2}\log(2) && \text{(A.25)}\\
&= \sum_{i=1}^{N}(x_i-\bar{x})^2+\frac{N}{2}(\bar{y}-\bar{x})^2 && \text{(A.26)}\\
& && \text{(A.27)}
\end{aligned}
$$

In order to get a symmetric test statistic, we can look at the product of the conditional probabilities so that

$$
\rho_X(Y)+\rho_Y(X)=\sum_{i=1}^{N}(y_i-\bar{y})^2+\frac{N}{2}(\bar{x}-\bar{y})^2+\sum_{i=1}^{N}(x_i-\bar{x})^2+\frac{N}{2}(\bar{y}-\bar{x})^2 \qquad \text{(A.28)}
$$

But we know that the sum of within cluster scatter and the between cluster scatter is equal to the total scatter. Thus,

$$
\sum_{i=1}^{N}(y_i-\bar{y})^2+\frac{N}{2}(\bar{x}-\bar{y})^2+\sum_{i=1}^{N}(x_i-\bar{x})^2=\sum_{i=1}^{N}\left(x_i-\left(\frac{\bar{x}+\bar{y}}{2}\right)\right)^2+\sum_{j=1}^{N}\left(y_j-\left(\frac{\bar{x}+\bar{y}}{2}\right)\right)^2 .
$$

Notice that for given data sets, the above summation is the same constant regardless of which points go with x_i and which with y_i. Thus,

$$
\rho_X(Y)+\rho_Y(X)=C+\frac{N}{2}(\bar{y}-\bar{x})^2 \qquad \text{(A.29)}
$$

where C is a constant. Thus a symmetric test statistic based on likelihood is

$$
\rho(X,Y)=\frac{N}{2\sigma^2}(\bar{y}-\bar{x})^2 . \qquad \text{(A.30)}
$$

The reason for normalizing by σ^2 will become clear shortly.

The Monte Carlo hypothesis tests can now be conducted with the distance functions ρ defined in this appendix. In Figure A.1 we show that the theoretically computed null distribution agrees with the null distribution computed empirically by random permutations.

It is important to statistically compare the test statistics $\rho_X(Y), \rho_Y(X)$, and $\rho_Y(X)$ computed in this section. Notice that,

$$\begin{aligned}\bar{x} &\sim N(0, \sigma^2/N) \\ \bar{y} &\sim N(0, \sigma^2/N) \\ \bar{x} - \bar{y} &\sim N(0, 2\sigma^2/N)\end{aligned}$$

Thus,

$$(\bar{x} - \bar{y})^2 \sim 2\sigma^2/N\chi_1^2$$

and,

$$\rho(X, Y) = \frac{N}{2\sigma^2}(\bar{x} - \bar{y})^2 \sim \chi_1^2 \tag{A.31}$$

Thus, $\rho(X, Y)$ has a mean of 1 and variance of 2

Similarly,

$$\frac{1}{\sigma^2}\sum_{i=1}^{N}(y_i - \bar{y})^2 \sim \chi_{N-1}^2$$

Thus,

$$\frac{1}{\sigma^2}\sum_{i=1}^{N}(y_i - \bar{y})^2 + \frac{N}{2\sigma^2}(\bar{x} - \bar{y})^2 \sim \chi_{N-1}^2 + \chi_1^2,$$

so that

$$\rho_X(Y) \sim \chi_N^2 \, . \tag{A.32}$$

We see that $\rho_X(Y)$ has a mean of N and variance of $2N$. This implies that the $\rho(X, Y)$ is a more powerful test statistic (in terms of false alarm) than $\rho_X(Y)$ or $\rho_Y(X)$.

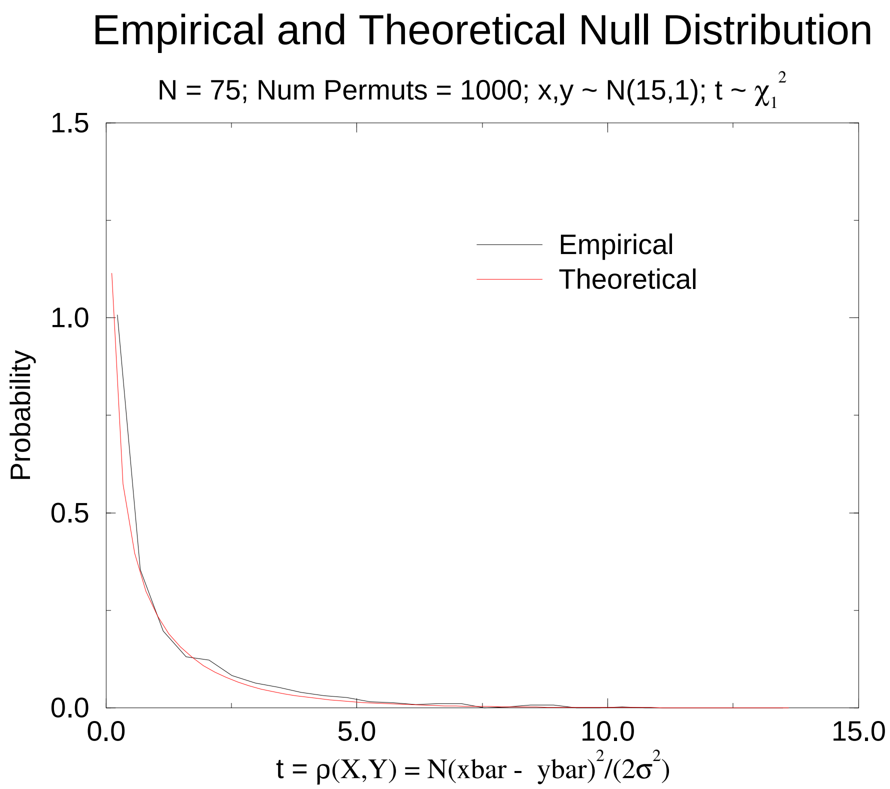


Figure A.1: Empirical and theoretical null distributions for two sample tests. Samples X and Y of size $N = 75$ are drawn from $N(15, 1)$. The empirical null distribution is computed as described in chapter 6. We use 1000 random permutations for computing the distribution. The distance function used is $t = \rho(X, Y) = N(\bar{x} - \bar{y})^2/(2\sigma^2)$. The theoretical distribution of t is χ_1^2. The empirical and theoretical plots have been plotted together in this figure.

Appendix B

POWER FUNCTIONS FOR GAUSSIAN POPULATIONS

Let us consider the case where we have a sample $x_1, x_2, \ldots, x_n$ drawn from a Gaussian population with a known variance, σ_0^2. That is, $x_i \sim N(\mu, \sigma_0^2)$. Suppose we have to test whether or not the mean of the population is equal to a specified value: i.e., $\mu = \mu_0$. If we have multiple ways of testing, it is fair to ask which one is the best. This can be ascertained by considering the probability of Type I and Type II errors incurred while performing each test and selecting the one that has lower Type I and Type II errors. Let H_N be the null hypothesis and H_A be the alternate hypothesis. That is,

$$H_N \; : \; \mu = \mu_0. \tag{B.1}$$

$$H_A \; : \; \mu \neq \mu_0. \tag{B.2}$$

When we reject H_N given H_N is true, we call the error to be *Type I*, or a *misdetection.* When we accept H_N given H_A is true, we call the error to be *Type II*, or a *false alarm.* In most statistical hypothesis testing procedures, one fixes the size ϵ of the test, which is the probability of Type I error. If ϵ is fixed for all the tests, the probability of Type I error is the same for all tests, and thus cannot be used for comparison. What *can* be used for further comparison amongst the tests is the Type II errors. The test that has the lowest Type II error is the best test.

We are given that under the null hypothesis $x \sim N(\mu_0, \sigma_0^2)$. Thus we can find x^α and α such that

$$P(x \geq x^\alpha | H_N) = \frac{1}{\sqrt{2\pi}\sigma} \int_{x^\alpha}^{\infty} e^{-\frac{1}{2\sigma_0^2}(x-\mu_0)^2} dx = \alpha \tag{B.3}$$

Thus as a test, one would use the following rule: if $x < x^\alpha$ accept H_N, otherwise accept H_A. By design we are assured that the misdetection rate is going to be ϵ. That is, if one runs this test T times with true null hypothesis, on the average, ϵT number of times the null hypothesis will be rejected.

Now consider the case when the alternate hypothesis, H_A, is true. That is, $x \sim N(\mu, \sigma_0^2)$, where $\mu \neq \mu_0$. Now the probability of rejecting the null hypothesis H_N as a function of μ is given below.

$$\gamma(\mu) \quad = \quad P(x \geq x^\alpha | H_A) = \frac{1}{\sqrt{2\pi}\sigma} \int_{x^\alpha}^{\infty} e^{-\frac{1}{2\sigma_0^2}(x-\mu)^2} dx. \tag{B.4}$$

The function $\gamma(\mu)$ is called the power function. The false alarm rate, which is probability that the null hypothesis is accepted given that the alternate hypothesis is true, is given by $P(x < x^\alpha | H_A) = 1 - \gamma(\mu)$.

If two tests have power functions $\gamma_1(\mu)$ and $\gamma_2(\mu)$, the test whose power function is higher for all values of μ, is a more powerful test, and is considered better. The power function has a minimum at $\mu = \mu_1$ where it attains a value ϵ and gradually increases as we go away from μ_1, and attains a value of 1 when we go far enough on either side. The sensitivity, i.e, the width of the notch, is a function of the sample sizes N and M and the various metrics used. When the sample size is small, the notch is broader and when the sample size is large, the notch is sharper.

Appendix C

MULTIVARIATE HYPOTHESIS TESTING FOR GAUSSIAN DATA

Multivariate hypothesis testing plays a central role in statistical analysis, which is an integral part of computer vision and image processing. Although theory of univariate hypothesis testing and software implementations are readily available, the theory of multivariate testing is usually not available in one book, and we are not aware of software implementations of multivariate tests. In this appendix we summarize various hypotheses that can be made about the population parameters of multivariate Gaussian distributions, and describe the various tests that can be conducted to reject these hypotheses. These tests have been implemented in C and we describe the interface to these functions. The theory and software have been validated by statistically testing empirical distributions against their theoretical distributions. The software is available free.

C.1 Computer vision and multivariate testing

Many computer vision problems can be posed as either parameter estimation problems (for example, estimate the pose of the object), or hypothesis testing problems (for example, the image represents which of the N objects in a database.) Since the input data (such as, images, or feature points) to these algorithms is noisy, the estimates produced by the algorithm are noisy. In other words, there is an inherent uncertainty associated with the results produced by any computer vision algorithm. These uncertainties are best expressed in terms of statistical distributions, and the distributions' means and covariances. Details of the theory and application of covariance propagation can be found in [Har94a], and the references cited in that paper.

Usually, implementations of vision algorithms run into thousands of lines of code. Furthermore, the algorithms are based on many approximations, and numerous mathematical calculations. One way to check whether the software implementation and the theoretical calculations are correct is by providing the algorithm input data with known (controlled) statistical characteristics, which is possible since the input data can be artificially generated, and then checking if the estimated output is actually distributed as what was predicted by theoretical calculations.

Since many of the estimation problems are multidimensional, testing whether the means and covariances of the empirical distribution and predicted distribution are the same is easier than testing whether or not the shapes of the two distributions are the same. In this article, we summarize statistical tests for the case when the random estimates can be assumed to be multivariate Gaussian. We also describe the function interfaces to software we have implemented for conducting these tests. Although the software libraries and environments (e.g. Splus, numerical recipes) are available for conducting the tests for one-dimensional samples, we are unaware of similar software libraries for multivariate case. In fact, most of the statistics books do not give all the five tests we have give (for example, Koch [Koc87] does not address the fifth testing problem). A description of how the software and the theory are tested using statistical techniques is also included.

C.2 The hypotheses

Let $x_1, x_2, \ldots, x_n$ be a sample from a multivariate Gaussian distribution with population mean μ and population covariance Σ. That is, $x_i \in R^p$ and $x_i \sim N(\mu, \Sigma)$, where p is the dimension of the vectors x_i.

We can make various hypotheses regarding the population mean and covariance depending on what is known and what is unknown. The data x_i are then used to test whether or not the hypothesis is false. Notice that each population parameter (here we have two – μ and Σ) can be either (i) tested, or (ii) unknown and untested, (iii) or known. If a parameter is being tested, then a claim regarding its value is being made. If a parameter is unknown and untested, no claim is being made about the value of that parameter; its value is not known and therefore we cannot use it in any computation. If the value of a parameter assumed to be known, then its value is known without error and cannot be questioned or tested, just like the normality assumption is not questioned. Furthermore, when a parameter value is known, the value itself can be used in computation of test statistics for other parameters.

In general, if the distribution has q parameters, then there can be $3^q - 2^q$ tests. The reasoning is as follows. Since each parameter can be either tested, or unknown and untested, or known, the number of possibilities is 3^q. But, of these the number of combinations in which none of the parameters are tested (that is, they are either known, or unknown and untested – and so do not represent a test) is 2^q. Thus, the total number of distinct hypotheses that can be made about a sample from a q-parameter distribution is $3^q - 2^q$.

In the case when the data comes from multivariate normal distribution, $N(\mu, \Sigma)$, we have $q = 2$ and thus can have $3^2 - 2^2 = 5$ possible hypotheses. Now we describe each of the five tests when the data comes from a multivariate normal population.

H_1 : $\mu = \mu_0$, ($\Sigma = \Sigma_1$ **known.**) In this test, the question is whether or not the sample is from a Gaussian population whose mean is μ_0. The population covariance Σ is assumed to be known and equal to Σ_1. Thus, no question can be asked regarding Σ_1 – the spread of the population from which the sample comes from is known without error, and can be used in the computation of any test statistic. The only thing that is unknown about the population is its mean. The data is used to reject the null hypothesis that the mean μ is actually equal to μ_0.

H_2 : $\mu = \mu_0$, (Σ **unknown, untested.**) In this test, the question is whether or not the sample is from a Gaussian population whose mean is μ_0. No statement is made regarding the population covariance Σ and since its value is unknown, it cannot be used in any computation of a test statistic. Thus, one is concerned whether or not the location of the sample is around μ_0, the spread can be anything and we do not care about that.

H_3 : $\Sigma = \Sigma_0$, ($\mu = \mu_1$ **known.**) In this test, the question is whether or not the sample is from a Gaussian population whose covariance is Σ_0. The population mean μ is assumed to be known and equal to μ_1. Thus, no question can be asked regarding μ_1 – the location of the population from which the sample comes from is known without error, and can be used in the computation of any test statistic. The only thing that is not known about the population is its covariance, and the data is used to reject any hypothesis about the population covariance.

H_4 : $\Sigma = \Sigma_0$, (μ **unknown, untested.**) In this test, the question is whether or not the sample is from a Gaussian population whose covariance is Σ_0. No statement is made regarding the mean μ and since its value is unknown, it cannot be used in computation of any test statistic. Thus, one is concerned whether or not the spread of the sample is around Σ_0, the location (mean) can be anywhere and we do not care about that.

$H_5: \ \mu = \mu_0, \ \Sigma = \Sigma_0$ In this test, the question is whether or not the sample is from a Gaussian population whose mean is μ_0 and covariance is Σ_0.

C.3 Definitions

In this section we briefly describe the terms used in the rest of the appendix. For a lucid explanation of the basic univariate concepts please see [CB90]. A more rigorous treatment of the univariate and multivariate test is given in [Arn90]. Multivariate tests are treated in great detail in [Koc87]. The most authoritative reference on multivariate statistics is [And84]. Although this book has most of the results, it is not very readable, and the results are scattered all over the book.

A *statistic* of the data $x_1, \ldots, x_n$ is any function of the data. For example, sample mean, $\bar{x}$, is a statistic, and so is the sample variance, S. The statistic need not be one-dimensional – $(\bar{x}, S)^t$ together form another statistic of the same data. A *sufficient statistic* is a statistic that contains all the information about the data; any inference regarding the underlying population can be made using just the sufficient statistic – the individual data points do not add any more information to the inference process. For example, the vector of original data $(x_1, \ldots, x_n)^t$ is a sufficient statistic – it contains all the information regarding the data. Another sufficient statistic is $(\bar{x}, S)^t$. Sufficient statistic is not unique. A *minimal sufficient statistic* is a sufficient statistic that has smallest number of entries. For example, for Gaussian data, $(\bar{x}, S)$ is the minimal sufficient statistic.

A *hypothesis* is any statement about a population parameter that is either true or false. The *null hypothesis,* H_0, and the *alternate hypothesis,* H_A, form the two complementary hypothesis in a statistical hypothesis testing problem.

A *test statistic* is just another statistic of the data that is used for testing a hypothesis. The *null distribution* is the distribution of the test statistic when the null hypothesis is true. The *alternate distribution* is the distribution of the test statistic when the alternate hypothesis is true.

There are two types of errors: misdetection and false alarm. If the null hypothesis is true but the test procedure decides the null hypothesis to be false, it is called a *misdetection.* When the alternate hypothesis is true but the test procedure accepts the null hypothesis, it is called a *false alarm.* The misdetection probability, α, of a test procedure is also referred to as the *significance level.* Typical value for α is 0.05.

The *power function* of a hypothesis test is a function of the population parameter θ, and value of the function $\beta(\theta)$ is equal to 1 minus the probability of false alarm. Ideally, the power function should be zero for θ where the null hypothesis is true and one for all θ where the alternate hypothesis is true. For most realistic testing problems one cannot create a test procedure with such an ideal power function. Power functions are very useful for evaluating hypothesis testing procedures, as was shown in this thesis. A *uniformly most powerful test* is a test procedure whose power function is higher than all other test procedures.

There are many methods for designing tests and corresponding test statistics. The test statistics given in this appendix were derived in [And84] by maximizing the likelihood ratio. Please refer to the cited literature for the derivation.

C.4 Test statistics, null distributions and power

In this section we summarize all the test statistics and their distributions under true null hypothesis and, if known, their distributions under the alternate hypothesis. For a detailed discussion and derivations please refer to [And84].

In the following discussion we use the following definitions of $\bar{x}$ and S :

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

and

$$S = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x})^t,$$

where we have assumed that the data vectors x_i are p-dimensional and the sample size is n.

C.4.1 Test 1: $\mu = \mu_0$ with known $\Sigma = \Sigma_1$

Test statistic:

$$T = n(\bar{x} - \mu_0)^t \Sigma_1^{-1}(\bar{x} - \mu_0). \tag{C.1}$$

Distribution under null hypothesis is Chi-squared (Anderson, page 73):

$$T \sim \chi^2_p.$$

The alternate hypothesis is $H_A : \mu \neq \mu_0$; the distribution under the alternate hypothesis is noncentral Chi-squared (Anderson, page 77):

$$T \sim \chi^2_{p,d}$$

where $d = n(\mu - \mu_0)^t \Sigma_1^{-1} (\mu - \mu_0)$ is the noncentrality parameter.
Reference: Anderson, pages 73, 77.

C.4.2 Test 2: $\mu = \mu_0$ with unknown Σ

Hotelling's Test statistic:

$$T = \frac{n(n-p)}{p(n-1)} (\bar{x} - \mu_0)^t S^{-1} (\bar{x} - \mu_0). \tag{C.2}$$

Distribution under null hypothesis (F):

$$T \sim F_{p,n-p}.$$

The alternate hypothesis is $H_A : \mu \neq \mu_0$; the distribution under the alternate hypothesis is noncentral F:

$$T \sim F_{p,n-p,d}$$

where $d = n(\mu - \mu_0)^t \Sigma^{-1} (\mu - \mu_0)$ is the noncentrality parameter.
Reference: Anderson page 163.

C.4.3 Test 3: $\Sigma = \Sigma_0$ with known $\mu = \mu_1$

Let

$$C = \sum_{i=1}^{n} (x_i - \mu_1)(x_i - \mu_1)^t = (n-1)S + (\bar{x} - \mu_1)(\bar{x} - \mu_1)^t .$$

and

$$\lambda = (e/n)^{pn/2} |C\Sigma_0^{-1}|^{n/2} \exp(-tr(C\Sigma_0^{-1})/2) .$$

Test statistic:

$$T = -2\log\lambda. \tag{C.3}$$

Distribution under null hypothesis is Chi-squared:

$$T \sim \chi^2_{p(p+1)/2}.$$

The alternate hypothesis is $H_A : \Sigma \neq \Sigma_0$; the distribution under the alternate hypothesis is unknown.
Reference: Anderson page 249, 434, 436.

C.4.4 Test 4: $\Sigma = \Sigma_0$ with unknown μ

Let

$$B = (n-1)S,$$

and

$$\lambda = (e/(n-1))^{p(n-1)/2}|B\Sigma_0^{-1}|^{(n-1)/2}\exp(-tr(B\Sigma_0^{-1})/2)$$

Test statistic:

$$T = -2\log\lambda. \tag{C.4}$$

Distribution under null hypothesis is Chi-squared:

$$T \sim \chi^2_{p(p+1)/2}.$$

The alternate hypothesis is $H_A : \Sigma \neq \Sigma_0$; the distribution under the alternate hypothesis is unknown.
Reference: Anderson page 249, 434, 436.

C.4.5 Test 5: $\Sigma = \Sigma_0$ and $\mu = \mu_0$

Define

$$B = (n-1)S$$

and

$$\lambda = (e/n)^{pn/2}|B\Sigma_0^{-1}|^{n/2}\exp\left(-[tr(B\Sigma_0^{-1}) + n(\bar{x}-\mu_0)^t\Sigma_0^{-1}(\bar{x}-\mu_0)]/2\right).$$

Test statistic:

$$T = -2\log\lambda \tag{C.5}$$

Distribution under true null hypothesis is Chi-squared:

$$T \sim \chi^2_{p(p+1)/2+p}$$

The alternate hypothesis is $H_A : \Sigma \neq \Sigma_0$, and $\mu \neq \mu_0$; the distribution under the alternate hypothesis is unknown.
Reference: Anderson page 442.

C.5 Validating theory and software

Two checks have to be performed. First check is that the theory is correct: the theoretically derived null distributions of the test statistics are actually correct. The second check is that the software is correct: the implementation is exactly what the theory dictates. Both the checks can be done by computing the empirical distributions and comparing them with the theoretically derived distributions. In the next subsection we describe how we empirically compute the null distributions of the five test statistics, and in the following section we describe how we use the Kolmogorov-Smirnov test to check if the empirical distribution and the theoretically-derived distributions are the same.

For our implementation we used public domain software for generating random numbers (ranlib library [BL]) and for computing P-values (cdflib library [BLR94]). Few other basic routines were borrowed from [PFTV90].

C.5.1 Empirical null distributions

In order to generate the empirical null distributions we proceed as follows.

1. Choose some values for the multivariate Gaussian population parameters p, μ and Σ.

2. Generate n samples from the population.

3. Compute the value of the statistic, T, for the test you are verifying.

4. Repeat steps 2 and 3 M times to get T_i, $i = 1, \ldots, M$.

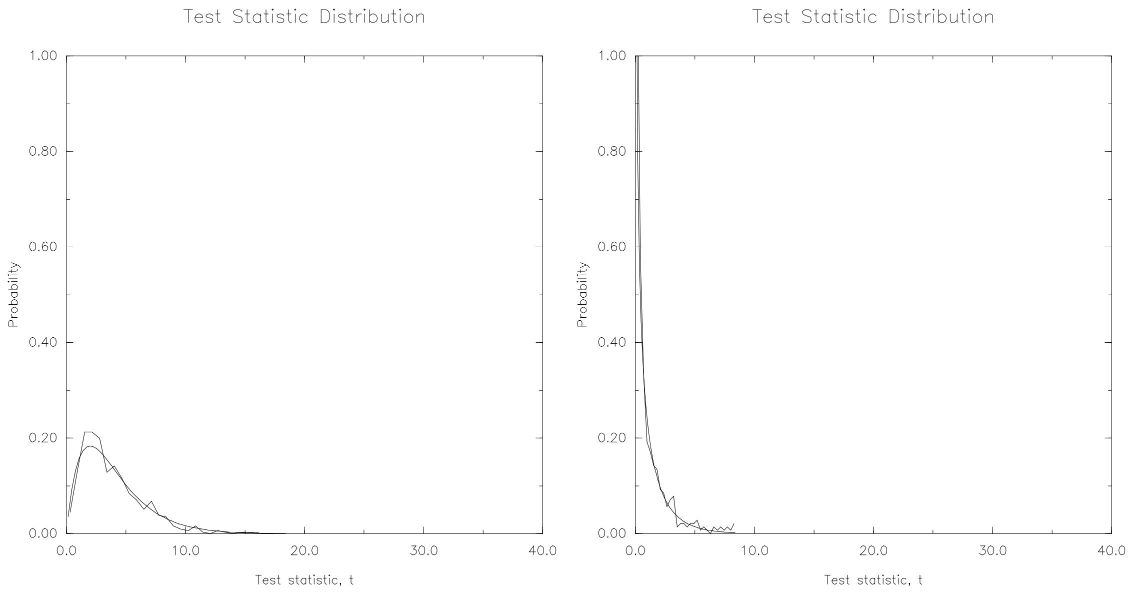
5. The empirical distribution T can be computed by computing the histogram of T_i.

We ran the above procedure for tests 1 through 5 described in the previous section and the plots of the empirical distributions of the corresponding test statistics are given in figures C.1 through C.5. The population parameters were: The dimension of data, $p = 4$ and $p = 1$; sample size, $n = 100$; population mean, $\mu = (1\ 2\ 3\ 4)^t$ for $p = 4$ and $\mu = 10$ for $p = 1$; population covariance, $\Sigma = 5I$ for $p = 4$ and $\Sigma = 5$ for $p = 1$; the number of repetitions $M = 500$. The histogram of the statistic and the theoretically derived function are shown in the figures. In the cases that the null distribution is distributed as χ^2, one can check the empirical distribution by using the fact that the mean, variance and mode of a χ^2_k random variable are k, $2k$, and $k - 2$, respectively.

C.5.2 Kolmogorov-Smirnov tests

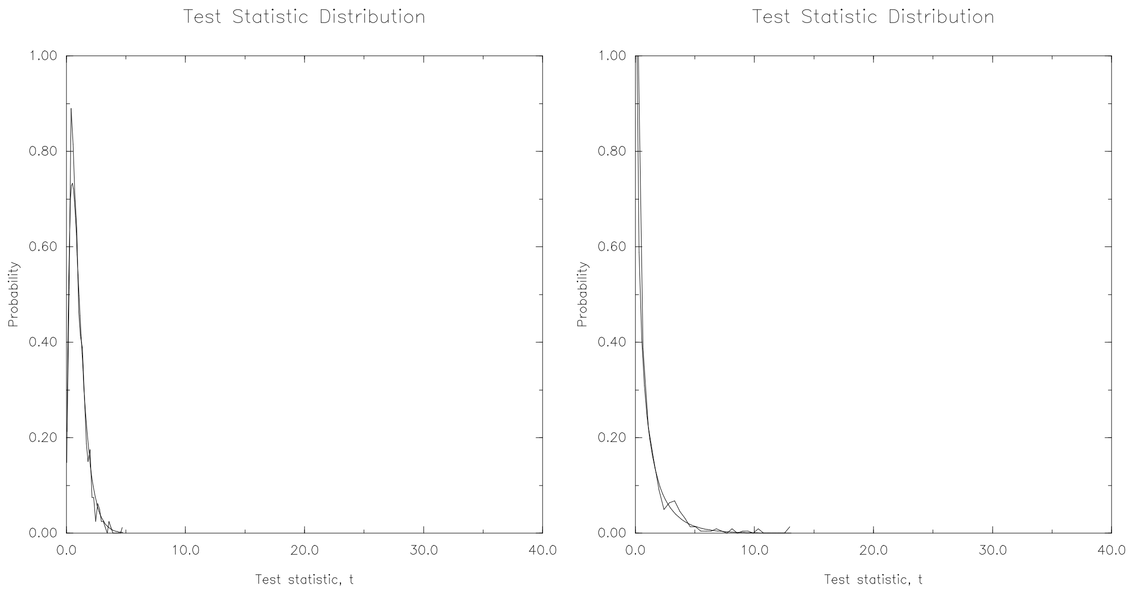
The Kolmogorov-Smirnov (KS) procedure tests whether two distributions are alike. The KS test uses the fact that the maximum absolute difference between the empirical cumulative distribution (the KS test statistic) and the theoretical cumulative distribution has a known distribution (the null distribution). For a more detailed discussion on the KS test see [PFTV90].

The Kolmogorov-Smirnov test was performed to check if the empirical distributions and the theoretical distributions were close enough. The p-value for the KS test are given in table C.1. All the empirically computed null distributions passed the KS test. Thus we have confirmed that the theoretical derivations of the null distributions are correct and the software implementing the theory is also correct.



(a) (b)

Figure C.1: Empirical and theoretical distributions of the Test 1 test statistic under true null hypothesis. In this case the theoretically derived null distribution is χ_p^2 where p is the dimension of the data. The histogram was computed by computing the test statistic T given in equation (C.1) $M = 500$ times. A sample size of $n = 100$ was used in each trial. In (a) dimension $p = 4$, population mean $\mu = (1\ 2\ 3\ 4)^t$, and the population covariance $\Sigma = I$. In (b) dimension $p = 1$, mean $\mu = 10$, and variance $\Sigma = 5$.



(a) (b)

Figure C.2: Empirical and theoretical distributions of the Test 2 test statistic under true null hypothesis. In this case the theoretically derived null distribution is $F_{p,n-p}$ where p is the dimension of the data. The histogram was computed by computing the test statistic T given in equation (C.2) $M = 500$ times. A sample size of $n = 100$ was used in each trial. In (a) dimension $p = 4$, population mean $\mu = (1\ 2\ 3\ 4)^t$, and the population covariance $\Sigma = I$. In (b) dimension $p = 1$, mean $\mu = 10$, and variance $\Sigma = 5$.

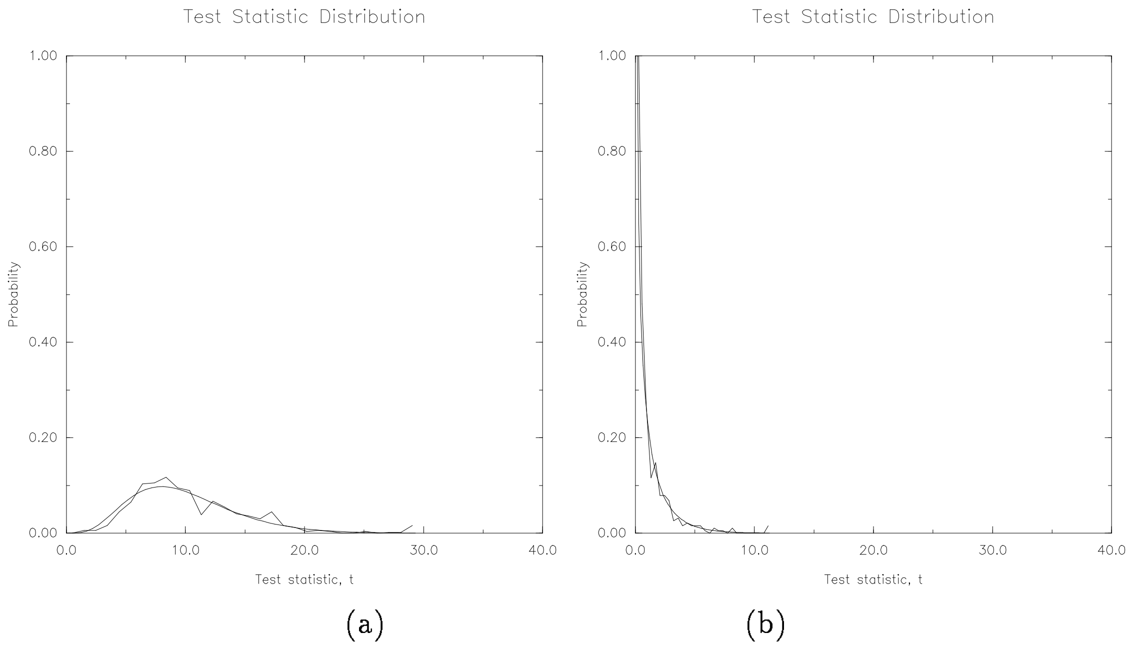
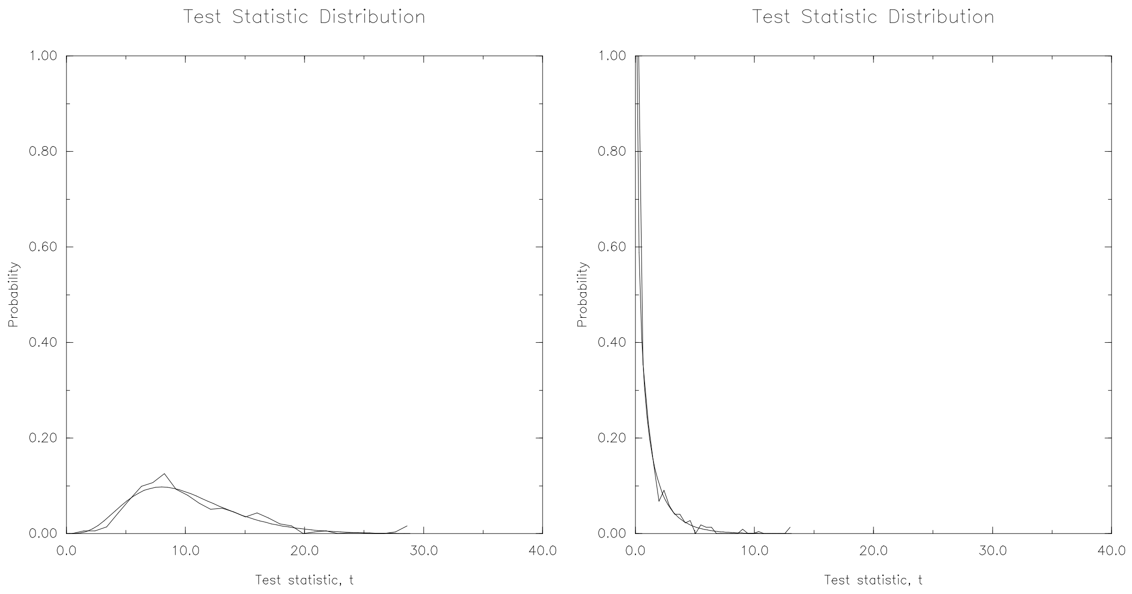


Figure C.3: Empirical and theoretical distributions of the Test 3 test statistic under true null hypothesis. In this case the theoretically derived null distribution is $\chi^2_{p(p+1)/2}$ where p is the dimension of the data. The histogram was computed by computing the test statistic T given in equation (C.3) $M = 500$ times. A sample size of $n = 100$ was used in each trial. In (a) dimension $p = 4$, population mean $\mu = (1\ 2\ 3\ 4)^t$, and the population covariance $\Sigma = I$. In (b) dimension $p = 1$, mean $\mu = 10$, and variance $\Sigma = 5$.



(a) (b)

Figure C.4: Empirical and theoretical distributions of the Test 4 test statistic under true null hypothesis. In this case the theoretically derived null distribution is $\chi^2_{p(p+1)/2}$ where p is the dimension of the data. The histogram was computed by computing the test statistic T given in equation (C.4) $M = 500$ times. A sample size of $n = 100$ was used in each trial. In (a) dimension $p = 4$, population mean $\mu = (1\ 2\ 3\ 4)^t$, and the population covariance $\Sigma = I$. In (b) dimension $p = 1$, mean $\mu = 10$, and variance $\Sigma = 5$.

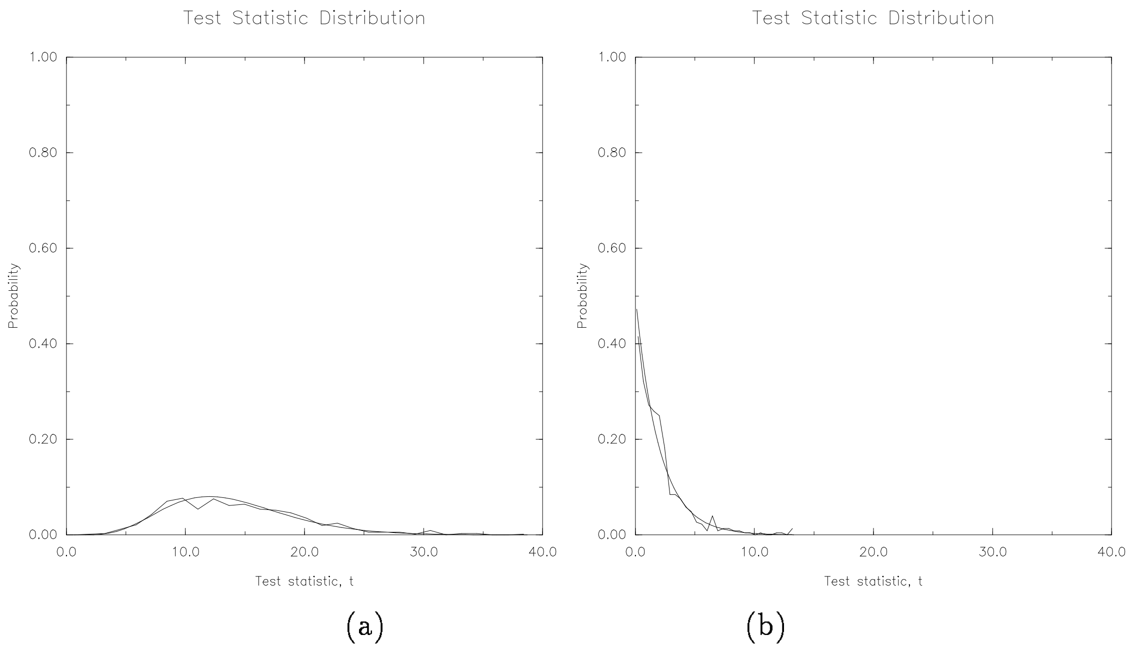


Figure C.5: Empirical and theoretical distributions of the Test 5 test statistic under true null hypothesis. In this case the theoretically derived null distribution is $\chi^2_{p(p+1)/2+p}$ where p is the dimension of the data. The histogram was computed by computing the test statistic T given in equation (C.5) $M = 500$ times. A sample size of $n = 100$ was used in each trial. In (a) dimension $p = 4$, population mean $\mu = (1\ 2\ 3\ 4)^t$, and the population covariance $\Sigma = I$. In (b) dimension $p = 1$, mean $\mu = 10$, and variance $\Sigma = 5$.

Table C.1: Kolmogorov-Smirnov test results for empirical null distributions shown in figures C.1 through C.5. Each empirical distribution was computed using $M = 500$ test statistic values. See the text for the corresponding population parameters.

Test	Dimension p	KS P-value	Pass at α=0.05?
1	4	0.322895	Yes
1	1	0.753276	Yes
2	4	0.563564	Yes
2	1	0.820641	Yes
3	4	0.337940	Yes
3	1	0.652343	Yes
4	4	0.338493	Yes
4	1	0.827236	Yes
5	4	0.157761	Yes
5	1	0.261129	Yes

VITA

Tapas Kanungo was born on the shores of river Ganges in the ancient Indian city of Varanasi – a city that was old when Buddha was young.

He earned his Bachelors degree in Electronics and Communication Engineering from Regional Engineering College, Tiruchirapalli, India, in 1986. He recieved his M.S. and Ph.D. in Electrical Engineering from the University of Washington, Seattle in 1990 and 1996, respectively.

From 1986 to 1988 Tapas Kanungo was with the Computer Science Group at the Tata Institute of Fundamental Research, Bombay, India. Since 1988 he has been with the Intelligent Systems Laboratory, at the University of Washington. Tapas Kanungo worked at the IBM Almaden Research Center, San Jose, CA, during the summer of 1993 and at the AT&T Bell Laboratories, Murray Hill, NJ, during the summer of 1994.

In 1990 Tapas Kanungo was a recipient of the Watamull scholarship and in 1992 he recieved the second prize at the Annual Industrial Affiliate's Poster Competition. He is a member of IEEE, ACM, and SIAM. His research interests are in the application of the scientific method to the areas of document understanding and information retrieval, computer vision and human vision.