

✓
D19
4/5

FOR REFERENCE ONLY

20 JUL 1998

40 0668550 2



ProQuest Number: 10183032

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



ProQuest 10183032

Published by ProQuest LLC (2017). Copyright of the Dissertation is held by the Author.

All rights reserved.

This work is protected against unauthorized copying under Title 17, United States Code
Microform Edition © ProQuest LLC.

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346

Algorithms for the Recognition of Poor Quality Documents

Ghulam Raza

**A thesis submitted to The Nottingham Trent University in
partial fulfillment of the requirements for the degree of
Doctor of Philosophy**

March 1998

*Department of Computing, The Nottingham Trent University
Burton Street, Nottingham, NG1 4BU
England, UK.*

Abstract

Optical Character Recognition (OCR) has engaged a number of researchers in developing suitable algorithms and systems that could translate human readable characters into machine readable codes accurately at high speed. Extensive work has presented successful results in recognizing good quality documents such as printed text but does not show satisfactory results for the poor quality documents, like low quality prints, photocopies, screen images, scanned old documents and facsimile messages.

Considering the limitations observed in past work, the present research investigates a suitable recognizer which could satisfactorily recognize poor quality documents. The work to date includes finding text lines, object extraction techniques, finding word gaps and finding words. It also includes methods for the extraction of different independent features. The features extracted during the current research include top side open, bottom side open, left side open, right side open, holes, top left corner open, top right corner open, bottom left corner open, bottom right corner open, vertical bars, horizontal bars, centre of gravity, dots of 'i' and 'j', and zones. These features are expected to be the same in the characters of different fonts and sizes and are tolerant to noise and hence can be used for the recognition of poor quality documents. Each feature contains some important information such as position in the object, length and width.

A method for the automatic creation of a database for both single and touching letters of any font and point size has been developed. Two methods (undercut and adding noise) for joining different letter combinations artificially and hence obtaining touching objects have been developed. A word recognition algorithm, based on object identification and dictionary lookup, for the recognition of poor quality documents has been described. The recognizer has two steps: finding object alternatives and making words using the alternatives and dictionary. The recognizer has been tested on fifty different facsimile messages containing 6029 machine printed words of different fonts, sizes and varied print qualities. The data was also tested using a commercial OCR software to obtain a comparative study of the recognizer and the commercial software. An overall 61.8% and 55.5% recognition rates are obtained for all facsimile messages using the recognizer and commercial software respectively. An improvement of 6.3% is found using the developed recognizer. The recognizer has also been tested on fifteen artificially created sample documents of different fonts and it gave an overall improvement of 10.3% compared with the commercial software. The results obtained confirm the effectiveness of the developed recognizer compared with the commercial system.

To improve the efficiency of the developed recognizer for a wide range of the poor quality documents further work is proposed. It involves improving existing methods for line and word extraction, feature extraction methods and extraction of additional new features. Future work considers finding methods for dealing with touching objects, document and context layout analysis, integration of a postulation algorithm into the developed recognizer and postprocessing.

The work presented in this thesis is original and author's own, unless otherwise specified by reference.

Acknowledgments

I would like to express my hearty gratitude to my supervisors Dr. N. Sherkat and Professor R. J. Whitrow for their invaluable guidance, help, encouragement, technical discussions and support in various problems concerning research during my time within the department. I am also very grateful to the Head of the Department of Computing for the use departmental facilities. I am also thankful to my parents and other family members for their continuous support and encouragement during my PhD study.

Dedicated to my respected parents

Table of Contents

Chapter 1

Introduction	1
---------------------------	----------

Chapter 2

Character recognition - a review	7
2.1 Introduction	7
2.2 Historical background	10
2.3 Classification of field	14
2.3.1 On-line or Dynamic recognition	14
2.3.2 Off-line or Static recognition	17
2.4 Character recognition problems	27
2.5 Applications of character recognition technology	30
2.6 Summary	33

Chapter 3

Object extraction	34
3.1 Introduction	34
3.2 Finding text line	35
3.3 Edge-following algorithm	37
3.4 Finding objects	39
3.4.1 Finding objects from bottom to top	39
3.4.2 Finding objects by colouring connected components	41
3.5 Finding word gap	46
3.6 Finding words	48
3.7 Summary	51

Chapter 4

Feature extraction	52
4.1 Introduction	52
4.2 Finding bars	56
4.3 Finding corners open	64
4.4 Finding centre of gravity	71
4.4.1 Centre of gravity using whole object	71
4.4.2 Centre of gravity using boundary of an object	72
4.5 Finding open sides using multi-colouring method	75
4.5.1 Finding the seed point	76
4.5.2 Multi-colouring to green	77
4.5.3 Finding a green/white pixel pair	79

4.5.4 Multi-colouring to red	81
4.5.5 Multi-colouring from red to green	82
4.5.6 Further steps	82
4.6 Finding zones	84
4.7 Finding dots of 'i' and 'j'	90
4.8 Summary	95

Chapter 5

Implementation	96
5.1 Introduction	96
5.2 Finding existing combinations of letters in a dictionary	97
5.2.1 Introduction	97
5.2.2 Method	98
5.2.3 Analysis	99
5.3 Automatic database development	100
5.3.1 Introduction	100
5.3.2 Overview of method	100
5.3.3 Converting to postscript format	103
5.3.4 Converting postscript to TIFF file	108
5.3.5 Creating and updating the database	108
5.3.6 Database format	114
5.3.7 Completion of database creation method	114
5.4 Artificially joining characters and objects	116
5.4.1 Introduction	116
5.4.2 Undercut	117
5.4.3 Adding noise	118
5.4.4 Summary	120
5.5 Recognizer	121
5.5.1 Introduction	121
5.5.2 Finding object alternatives	122
5.5.3 Building words	124
5.6 Summary	127

Chapter 6

Results and Conclusions	128
6.1 Introduction	128
6.2 Experiment 1	129
6.2.1 Sample data collection	129
6.2.2 Results	130
6.3 Experiment 2	141
6.3.1 Sample data collection	141
6.3.2 Results	141
6.4 Conclusions	146
6.5 Summary	150

Chapter 7	
Future work	151
7.1 Introduction	151
7.2 Line and word extraction	152
7.3 Feature extraction	152
7.3.1 Improving existing methods	153
7.3.2 Extracting new features	153
7.4 Document and context layout analysis	154
7.5 Postulation algorithm	154
7.5.1 Introduction	154
7.5.2 Proposed method	156
7.6 Ideas for dealing with touching objects	158
7.6.1 Introduction	158
7.6.2 Extracting features from upper half of the objects	158
7.6.3 Extracting features from upper and lower halves separately	159
7.6.4 Segmenting touching objects based on features	159
7.7 Postprocessing	161
7.8 Summary	162
References	163
Appendix A. OCR for machine-printed documents	207
Appendix B. OCR for hand-printed documents	214
Appendix C. Sample facsimile messages	220
Appendix D. Published papers	271

List of Tables

Table 4.1: Letters with different sizes having same features	54
Table 4.2: Letters with different fonts having same features	55
Table 4.3: Criteria to identify dot-shaped objects [Hennig et al 97]	93
Table 5.1: Total number of existing combinations for 2 and 3 letters in each type of dictionary and their net total	99
Table 5.2: Total number of possible combinations for 2 and 3 letters in each type of dictionary and their net total	99
Table 5.3: Setting of the code characters	105
Table 6.1: Recognition results for facsimile messages obtained using commercial software (READIRIS)	131
Table 6.2: Recognition results for facsimile messages obtained using developed recognizer	133
Table 6.3: Recognition results for artificially created sample documents obtained using commercial software (READIRIS)	142
Table 6.4: Recognition results for artificially created sample documents obtained using the developed recognizer	143

List of Figures

Figure 1.1: Outline of the word recognition approach [Raza et al 97b]	3
Figure 2.1: Illustration of a peephole method [Mori et al 92]	12
Figure 2.2: The different areas of character recognition	15
Figure 2.3: Different fonts and standards	27
Figure 2.4: Few characters in different point sizes	29
Figure 3.1: Searching for lines of text	36
Figure 3.2: Rectangle coordinate of an object	37
Figure 3.3: The movement vectors describing anticlockwise movement.	38
Figure 3.4: Finding objects from bottom to top.	40
Figure 3.5: An image in which method fails to extract all objects	40
Figure 3.6: A sample document image	41
Figure 3.7: Direction diagram.	42
Figure 3.8: Seed point, colouring connected components effect and bounding rectangle	43
Figure 3.9: Sample image after complete colouring process	43
Figure 3.10: Order of the objects after colouring process	44
Figure 3.11: Removing huge and tiny objects and obtaining writing sequence	45
Figure 3.12: Sample image after removing huge objects	45
Figure 3.13: Method for finding word gap	47
Figure 3.14: Word finding method	49
Figure 3.15: Sample poor quality facsimile message containing hand written text, tables and overlapping lines	50
Figure 3.16: Manually marking of word boundaries	50
Figure 4.1: The objects with missing and unwanted extra parts	53
Figure 4.2: A sample input image.	56
Figure 4.3: Direction indicator	57
Figure 4.4: Finding a line of black pixels	58
Figure 4.5: Sample images for setting minimum bar length	59
Figure 4.6: Different settings for minimum bar length (a) 80%, (b) 30% and (c) 40%	60
Figure 4.7: Continuing the search for a vertical bar	61
Figure 4.8: A vertical bar	62
Figure 4.9: A vertical bar in close up	62
Figure 4.10: Direction diagram	64
Figure 4.11: A sample input image with initial starting point	65
Figure 4.12: Forming a rectangle	66
Figure 4.13: Adding a line and expanding the rectangle	66
Figure 4.14: Starting point of the next line, finding line and expanding rectangle	67
Figure 4.15: The latest situation	67
Figure 4.16: Direction exhausted as black pixel is found after start point	68
Figure 4.17: Next pixel	68

Figure 4.18: Another line added to the rectangle	69
Figure 4.19: Maximum successful growth of the rectangle.	69
Figure 4.20: Direction exhausted as start pixel is black	70
Figure 4.21: Storing feature details	70
Figure 4.22: Sample input object images	71
Figure 4.23: Centre of gravity using whole object	72
Figure 4.24: Sample images obtained from the boundary the real objects	73
Figure 4.25: Origin	73
Figure 4.26: (a) A binary input image, (b) Colour image including border	76
Figure 4.27: Seed point	76
Figure 4.28: Coloured image	77
Figure 4.29: Corners	78
Figure 4.30: The colouring process	78
Figure 4.31: The effect of colouring after using all necessary pair directions	79
Figure 4.32: Corner obtained from the direction pair.	80
Figure 4.33: New inside seed point	81
Figure 4.34: The effect of colouring by using new seed point	81
Figure 4.35: Image after setting all red pixels to green	82
Figure 4.36: Another open side found by colouring white to red.	83
Figure 4.37: Finding the origin and extents.	83
Figure 4.38: Text with different fonts and sizes	85
Figure 4.39: Text line bigger in beginning and smaller at the end	85
Figure 4.40: Line of text with first letter much bigger than others	86
Figure 4.41: A sample word image	86
Figure 4.42: A horizontal projection histogram derived from colour transitions for the example word (a) Example word image (b) Original histogram (c) Smoothed histogram	87
Figure 4.43: Calculation of upper, middle upper, middle lower and lower lines.	89
Figure 4.44: Estimation of upper and lower lines in a word with no ascenders and descenders	89
Figure 4.45: Sample image for finding dots	90
Figure 4.46: The effect of aspect ratio constraint.	91
Figure 4.47: The effect of area to bounding box ratio constraint.	91
Figure 4.48: The effect of circumference to area ratio constraint.	92
Figure 4.49: The effect of concavity area constraint	92
Figure 4.50: Identifying dot-shaped regions: (a) height/width ratio; black pixels/area of bounding rectangle ratio; boundary length/region area ratio; (b) area of concavities/convex hull area ratio (c) no white pixels are allowed inside the area [Hennig et al 97].	93
Figure 4.51: The effect of all constraints	94
Figure 4.52: Storing feature details	94
Figure 5.1: Overview of the method	101
Figure 5.2: A sample dictionary	102

Figure 5.3: Possible objects in the sample dictionary.	102
Figure 5.4: postscript file format	104
Figure 5.5: A sample encoding	105
Figure 5.6: Font dependent features example.	107
Figure 5.7: TIFF file image for the two letter combinations	108
Figure 5.8: Text page	109
Figure 5.9: A text line.	112
Figure 5.10: Object boxes on a line.	112
Figure 5.11: Database entry format	114
Figure 5.12: An example entry database	114
Figure 5.13: A two letter object.	116
Figure 5.14: Two touching characters	116
Figure 5.15: The effect of different negative undercut values on some two letter combinations	118
Figure 5.16: Steps in creating noisy TIFF file	119
Figure 5.17: The effect of different noise and blur values on some two letter combinations	120
Figure 5.18: Outline of the word recognition approach [Raza et al 97b].	121
Figure 6.1: Some poor quality words from different facsimile messages	130
Figure 6.2: Performance of the commercial software and developed recognizer (considering Top1 and Top2 choices)	135
Figure 6.3: Average recognition rate for all facsimile messages using the READIRIS software and the developed recognizer by considering different choices.	136
Figure 6.4: Recognition rate obtained using developed recognizer by considering different choices (Top1, Top2, Top5, Top10)	139
Figure 6.5: Recognition rate obtained using developed recognizer by considering different choices (Top5, Top10, Top15, Top20)	139
Figure 6.6: Comparing output for some sample words	140
Figure 6.7: Performance of the commercial software and the developed recognizer on artificially created sample documents (considering Top1, Top2 and Top5 choices).	144
Figure 6.8: Average recognition rate for different fonts using the READIRIS software and the developed recognizer by considering top1 choice	145
Figure 7.1: Sample word for recognition consisting of strong and weak objects	157
Figure 7.2: Recognized strong objects and unknown weak objects	157
Figure 7.3: Touching characters with no top side open feature.	160
Figure 7.4: Objects containing the letters 'U' or 'u'	161

Chapter 1

Introduction

Recent years have seen significant advances in the fields of image analysis and pattern recognition [Rosenfeld 87] and [Mantas 87]. One of the most important areas of pattern recognition is Optical Character Recognition (OCR), which has a number of industrial, business and scientific applications.

Bearing in mind the importance of this area of research and its application in practical life, a lot of research effort has been devoted to the subject during the past 30 years. The ultimate objective of the research has been to translate machine printed human readable documents into machine readable codes with the same or better accuracy than human beings. Such documents may contain text in a variety of font styles, point sizes and print qualities.

Because of substantial research effort by the researchers in different parts of the world, many character recognition algorithms and systems have been designed and produced. The performance of existing algorithms and systems depends mainly upon the quality of the document which is to be recognized. If the quality of the document is good, then existing systems perform reasonably well, and a good

recognition rate (more than 99% using some OCR systems) is obtained, for example [Baird 88].

The recognition rate of poor quality documents decreases dramatically when using existing systems. Poor quality data appears very commonly in practical life. It comes from different sources such as scanned old documents, low quality prints, photocopies, facsimile messages, etc. Furthermore, existing OCR techniques are unsuitable for dealing with screen images [Harness et al 93] and [Sherkat et al 93].

The present research has been carried out to develop a reading machine that can be used for the recognition of documents containing a wide range of font styles, point sizes, and of highly variable print quality.

Most poor quality documents contain touching, broken and part-missing characters. Many existing OCR algorithms and systems rely on segmenting the touching characters. It has been observed that half of the errors in recognition occur because of incorrect segmentation of touching characters [Chen and DeCurtins 93]. It is also reported in the literature that methods of segmenting touching characters appear to be ad hoc in nature and thus not applicable to the general text [Elliman and Lancaster 90], especially in poor quality documents.

Bearing in mind the main drawbacks of poor quality documents and the unsatisfactory results achieved by applying segmentation of touching characters on such documents, the present research has steered away from a segmentation based approach. In the present work, we rely on the higher level information of the document such as word shape and lexical information.

Several alternative OCR methods have been proposed by different researchers such as subword image matching [Hong and Hull 95], sliding windows [Bhate et al 95], first-last character decisions [Zhao and Srihari 95a], and word-shape

recognition [DeCurtins and Chen 95], [Zhao and Srihari 94]. A generalization of the Levenshtein edit distance, more suitable for OCR, was proposed in [Seni et al 95]. A methodology integrating segmentation with recognition via Markov source modeling was elaborated in the series of papers [Chou and Kopec 95], [Kam and Kopec 95], [Kopec et al 95] with applications to printed music as well as text. [Chen et al 95] discussed the detection of multi-word phrases. An improved method of text recognition via deciphering was presented by [Fang and Hull 95]. The results obtained using these methods by no means suggest that the problem is solved. The proposed work aims to continue this approach and solve the existing problems related to poor quality documents.

The outline of the developed Word Recognition (WR) approach based on object recognition and dictionary lookup is shown in Figure 1.1. An object is a black pixel or group of black pixels completely surrounded by white pixels. In our case, an object is generally either a single character, two or more touching characters or a punctuation mark.

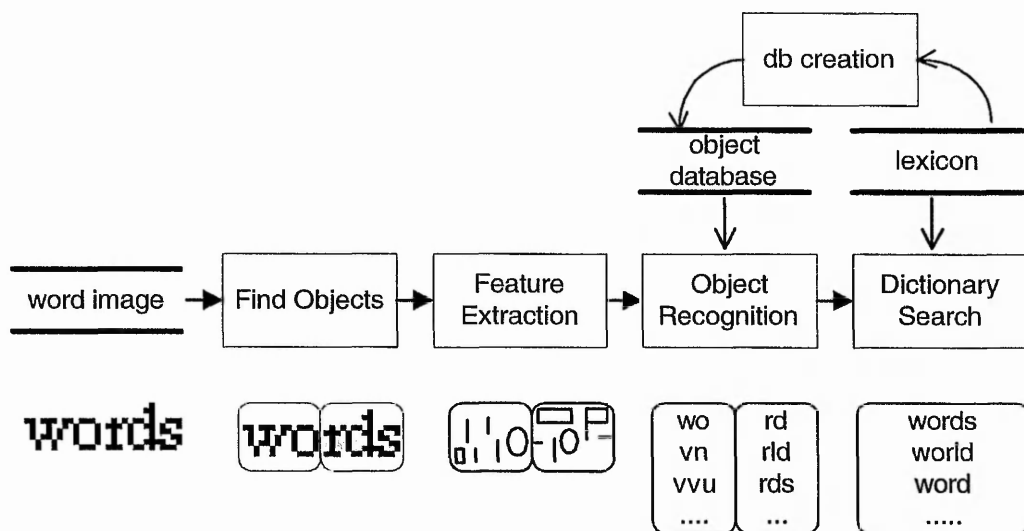


Figure 1.1: Outline of the word recognition approach [Raza et al 97b]

In this approach, the first step is to find different objects of an input word image without attempting to segment the touching characters. Later, each object is passed to the feature extractor, where different independent features are found, which represent the ideal form of the object. Each feature contains important information about the object, such as its position in the object and dimensions. The features extracted are expected to be the same in characters of different fonts and sizes. Hence a database of objects of one point size and font style is enough for the recognition of text of different font styles and point sizes. This is especially true for different point sizes, as the features extracted are normalized. The features considered during the current research are as follows:

- Holes

- Top side open

- Bottom side open

- Left side open

- Right side open

- Top left corner open

- Top right corner open

- Bottom left corner open

- Bottom right corner open

- Vertical bars

- Horizontal bars

- Center of gravity
- Dots of 'i' and 'j'
- Zones (upper, middle, lower, full)

Each object shape is then compared with different object shapes in the database. Objects in the database with a similar shape are selected as different candidates for the object. The database object which has most features in common with the sample object has least cost and is considered as the most likely candidate for that object. After classifying each poor quality object of the word, the alternatives are combined together to form valid words using dictionary lookup. The word(s) with least cost (highest number of features matched) is(are) considered as the most likely candidate(s) for that sample word. The main advantage of this approach is that it bypasses the errors incurred during segmentation of touching characters. If there is more than one word with the same least cost or if the sample word is recognized correctly, but at the lower rank (more cost), then higher level linguistic information such as language syntax and semantics can be used to identify the correct word.

The first chapter of this thesis introduces the problem and states the objective of the present research and outlines the developed word recognition approach.

In the second chapter, a literature review in the area of OCR is described. Historical background of OCR is also presented. Different areas of character recognition are reviewed. Research in the area of poor quality documents is described in detail. Applications of this work and associated problems are also presented.

In the third chapter, object extraction techniques are presented, together with the line and the word finding method. The method for finding a word gap is also described.

The different features extracted for the recognition of poor quality documents along with the extraction method for each feature is described in chapter four. The features extracted, as listed previously, contain certain information about the object such as its position in the object (origin), and length and width (extents), etc.

In chapter five, we describe the implementation of a recognizer system based on the above feature extraction methods. The method for the automatic development of a database, containing both single and touching characters, of any font and point size, is presented. Two methods 'undercut' and 'adding noise' for the artificial creation of touching characters are also described.

In chapter six, results obtained using the recognizer and commercial OCR software are presented. A description regarding collection of the sample documents/data used for achieving the results is also given. It also gives interim conclusions based on the results achieved.

Finally, chapter seven of this thesis gives some suggestions for any further work to be carried out based on the developed method, experiments, results and conclusions achieved during the research.

Chapter 2

Character recognition - a review

2.1 Introduction

There are two approaches that can be applied to isolated word recognition; the Analytic approach or 'Character-based Word Recognition' and the Holistic approach or 'Word Shape Recognition' [Srihari 93].

Considering the analytic approach, word recognition can be performed by a sequential process consisting of the following stages [Kahan et al 87]:

- Character segmentation
- Character recognition, and
- Postprocessing

The word shape approach tries to recognize the entire word [Ho et al 92] and character segmentation is bypassed. Given a word image, its word-shape features are extracted and matched with the word-shape features of the words in the dictionary. The first '*n*' best matched words are considered as candidates for the

word. Using this approach, a sample word image cannot be recognized correctly if its identity is not included in the dictionary.

Character recognition is better known as Optical Character Recognition (OCR) since it deals with the recognition of optically processed characters rather than magnetically processed ones [Govindan and Shivaprasad 90]. It has a long history and is the dominant paradigm in the present research in the text recognition field [Mori et al 92].

Character recognition techniques associate a symbolic identity with the image of a character. This problem of replication of human functions by machines involves the recognition of both machine printed and handprinted/cursive-written characters.

For a given character image, the character recognition system algorithm takes decisions regarding character identifications, which is accomplished by template matching or structural analysis. For recognition of any of the hundreds of fonts in common use, current omni-font OCR systems adopt the structural analysis approach [Baird and Nagy 94], [Bokser 92], [Kahan et al 87]. Many different features including various statistical and structural aspects have been proposed for concise description of the structure of characters in various fonts and font sizes [Kahan et al 87], [Mori et al 92], [Srihari 92]. Also many classification methods like Bayesian classifiers [Duda and Hart 73], nearest neighbour classifier [Dasarathy 91], neural network classifiers [Lippman 87], [Rumelhart and McClelland 82], decision tree classifiers [Moret 82], and syntactic classifiers [Fu 82] have been presented to classify the character images.

Although significantly high accuracies have been observed on good-quality character images the problem of OCR performance on poor-quality character

images remains [Baird 93], [Bokser 92]. Some researchers have proposed new feature representations and new classification methods [Ho and Baird 93], [Bokser 92]. More accurate recognition may be achieved by combining several classifiers [Ho 92], [Xu et al 92]. It has been suggested that the role of training sets for classification rather than classification methodology is an important factor for obtaining an improved recognition [Ho and Baird 94]. OCR efficiency can be enhanced by taking advantage of local typeface homogeneity [Baird and Nagy 94].

The character recognition task is not as simple as it might appear, especially the recognition of poor quality printed documents, facsimile messages and handprinted characters. In fact, even human beings incorrectly identify about 4% of the words in handprints in the absence of context [Suen et al 77].

Character segmentation is a primary step that determines performance of the recognition system. Being an important pre-processing step, character segmentation sections the word image into a number of character images to facilitate application of OCR approaches [Casey 95]. Apart from problems of isolated character classification, improper character segmentation has been seen as one of the key factors for incorrect recognition [Casey and Nagy 82]. For character segmentation the small space available between characters can be regarded as a segmentation point. However, in some situations this technique may not work where there are touching and broken characters resulting usually in degraded text such as multiply generated photocopies or facsimile messages. In such cases, more than one character may be grouped as one character image or one character image may be split into several pieces. A number of methods have been presented showing improvement in segmentation techniques. Some of them are based on connected component analysis, profile analysis, and aspect ratio estimation [Bokser 92], [Elliman and Lancaster 90], [Liang et al 93], [Kahan et al 87]. Also,

there are other techniques which integrate character segmentation with character recognition based on the idea that segmentation decisions are tentative until confirmed by successful recognition of the segmented images [Casey and Nagy 82], [Tsujimoto and Asada 91].

The origin of character recognition can be found as early as 1870, it first appeared as an aid to the visually handicapped [Govindan and Shivaprasad 90]. During the past 30 years, substantial research effort has been devoted to character recognition, which is used to translate human readable characters into machine readable codes.

Several books on optical character recognition [Fisher et al 62], [Kovalevsky 68], [Weaver 72], [Wilson 66], as well as special issues and reports on OCR [Stevens 70], have been published. These publications may provide the interested reader further information on this topic. Special sessions on OCR have repeatedly appeared in the proceedings of the International Joint Conference on Pattern Recognition and of the International Systems, Man, and Cybernetics Conference. Additionally, extensive bibliographies on OCR [Gaillat and Berthod 79], [Harmon 72], [Prather 70], [Shillman et al 74], [Stevens 61], [Suen 78], [Elliman and Lancaster 90], [Govindan and Shivaprasad 90], [Tappert et al 90], [Impedovo et al 91], [Tian et al 91], [Mori et al 92], [Matsui et al 93] have been compiled.

2.2 Historical background

The origin of character recognition can be found in 1870 when Carey invented the retina scanner, that is an image transmission system using a mosaic of photocells, and later in 1890 when Nipkow invented the sequential scanner which was a major breakthrough both for modern television and reading machines [Anderson 69], [Rabinow 69]. However, character recognition first appeared as an

aid to the visually handicapped and the first successful attempt was made by the Russian scientist Tyurin in 1900 [Mantas 86]. [Tauschek 35] obtained a patent on OCR in Germany and [Handel 33] did the same in the United States. At that time certain people dreamed of a machine which could read characters and numerals. This remained a dream until the age of computers arrived, in the 1950's. The principle of Tauschek's work is template matching. This reflects the technology at that time, which used optical and mechanical template matching. Light passed through mechanical masks was captured by photodetectors and scanned mechanically. When an exact matching occurs, light fails to reach the detector and so the machine recognized the characters printed on paper. Other reported attempts are the Fourier d'Albe's Optophone of 1912 and Thomas' tactile "relief" device of 1926.

As mentioned above the modern version of OCR [Anderson 71] appeared in mid 1940s with the development of the digital computer. For the first time, OCR was realized as a data processing approach, with particular application to the business world. From that perspective, David Shepard, founder of the Intelligent Machine Research Company can be considered as a pioneer of the development and building of commercial OCR equipment [Anderson 71].

A sophisticated OCR machine was made combining electronics and optical techniques by [Hannan 62] at Radio Corporation of America (RCA). At that time RCA had the most advanced electron tube technology in the world, which was fully employed in OCR research work. In his paper he concluded that the test results of this programme proved that the RCA optical mask-matching technique can be used to reliably recognize all characters of complete English and Russian fonts (91 channels were necessary). However, no announcement was made for a

commercial RCA OCR based on the techniques. The great experiment ended without a successor.

It is very natural that the advent of computers influenced the design of OCR with respect to hardware and algorithms. A logical template matching method (the simplest one is the peephole method) may be introduced. First of all we assume that an input character is binarized. Binarization is an important preprocess in OCR technology. Ideally an input character has two levels of density, i.e. black and white, commonly represented by 1 and 0 respectively. Imagine a two-dimensional memory plane on which a binarized input character is stored and registered in accordance with some rule, with the character positioned at the top right corner, as shown in Figure 2.1. Then for an ideal character, which has a stroke of constant size and width, black portions are always black and the same is true for white background. The appropriate pixels are chosen for both black and white regions so that the selected pixels can distinguish the input character from characters belonging to other classes. In Figure 2.1, a so-called logical matching scheme is constructed, which is called the peephole method.

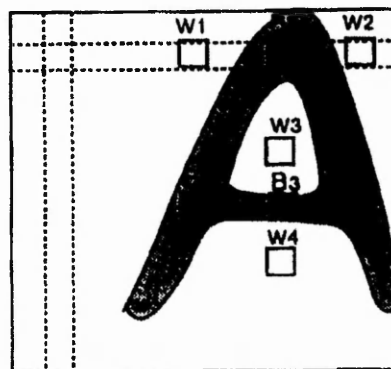


Figure 2.1: Illustration of a peephole method [Mori et al 92]

The first OCR based on the peephole method was announced by the Solatron Electronics Group Ltd. [ERA 57] and called Electron Reading Automation (ERA) in 1957. The characters read were numerals printed by a cash register. The reading speed was 120 characters/second (chs), which was very high due to simple logic operations. The total number of peepholes was 100, which is considerably greater than the ideal number of $\lceil \log_2 10 \rceil = 4$ which would be needed to obtain stable recognition for real data.

[Hodges 83] claims that Alan Turing had devised a scheme for character recognition based on the use of a television camera to map images on the storage tubes used for some of the early computers. Apparently the topic was of great interest to the early pioneers of computing and their discussions foreshadowed some of debate that continues to this day. Norbert Wiener [Wiener 48] mentioned that in 1947 McCulloch and Pitts had designed a machine to read for the blind and thus they had solved the problem of “(making) the pattern of the letters”. On the other hand Turing is quoted as referring to McCulloch in very disparaging terms [Hodges 83].

An alternative approach to the problem of transferring information from the printed to the electronic media would be to print information using magnetic ink, and then read it by non-optical means. As a matter of fact the term ‘Optical Character Recognition’ has been devised in order to make a distinction from such magnetic readers. This technology found applications quite early and is still used widely to read, for example bank cheques. The difference of the scanner technology however obscures a more basic difference between ‘magnetic’ and ‘optical’ character recognition. The number on bank cheques have been designed so that they can be read easily by a machine: the mechanical readers do not look at their shapes but rather at their width in a few places. The form of the character

outside certain areas is irrelevant to the mechanical readers and some effort has been made to use those parts to make the characters readable by people. Thus we have a fundamental split in approaches: making the text easier for the machine to read versus making the machine able to read any text. Leaving aside the pre-history of the field, we refer the reader to the thorough historical review, of OCR, by [Mori et al 92] for a discussion of the substantive work done over the past 35 years.

2.3 Classification of field

A general classification of the character recognition field is shown in Figure 2.2. The two main categories each having its own hardware and recognition algorithms are presented below:

- a) On-line or Dynamic recognition
- b) Off-line or Static recognition

2.3.1 On-line or Dynamic recognition

In on-line character recognition the symbols are recognized as they are drawn [Arakawa et al 78], [Bernstein 68], [Berthod 78], [Brown 64], [Tappert et al 88]. This recognition exploits the temporal information about the movement of the pen. The most common writing surface is the digitizing tablet, which typically has a resolution of 200 points per inch. Different types using various technologies are available and include: electromagnetic /electrostatic [Carau and Tremblay 81], [Pepper 78], [Prugh and Fadden 80], pressure sensitive [Gibson and Talmage 80], [Turner and Ritchie 70], [Lukis and Duhing 85], [Buckle and Strand 81], acoustic sensing in air medium [Bruyne 86], [Romein 81] types and so on. Since in

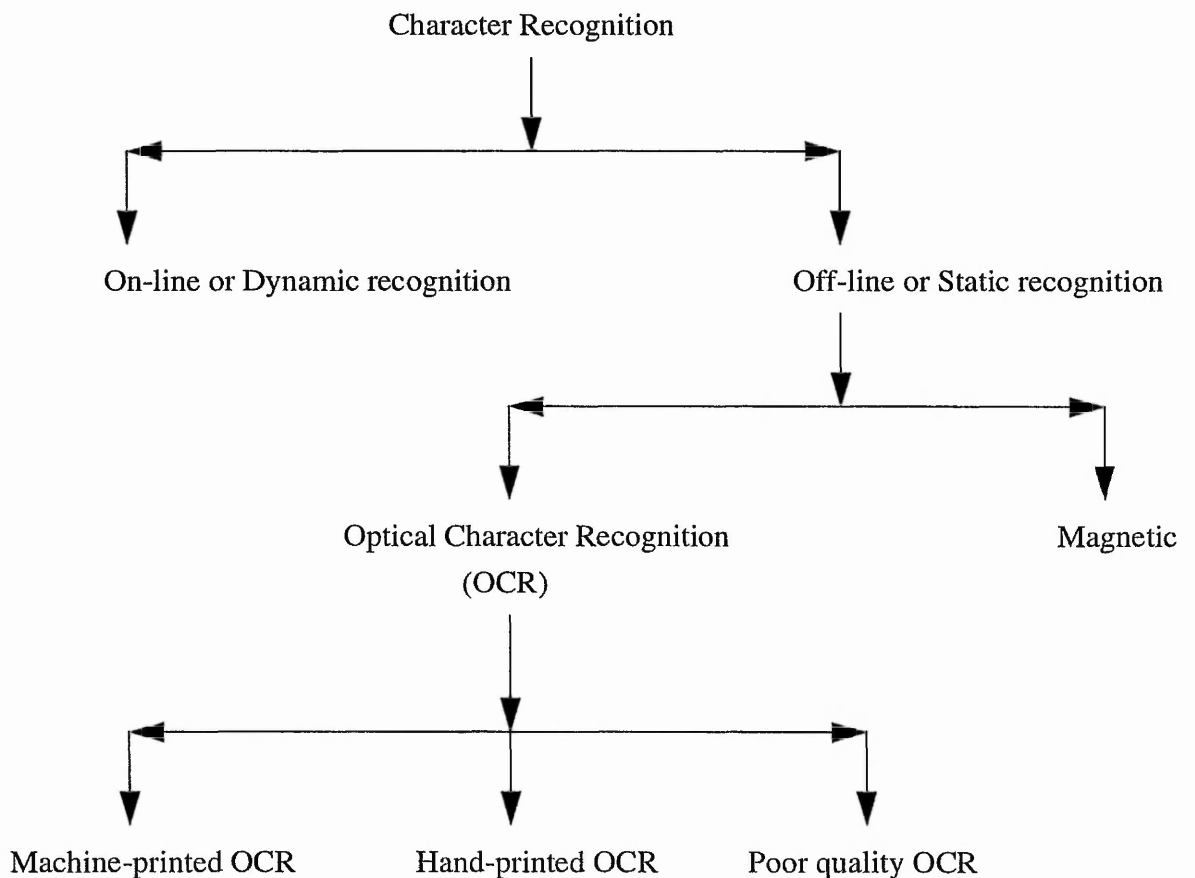


Figure 2.2: The different areas of character recognition

dynamic recognition, characters are represented by line drawings, there is no need for skeletonization, which is relatively costly and an imperfect process. Moreover, several systems make use of the input stroke sequence of a character for recognition rather than the drawing itself.

On-line recognition cannot cope with the document images which have already been produced. On-line recognition requires temporal information, where a transducer is required to capture the writing as it is written. Otherwise, temporal information has to be found by analyzing the shape of the static image [Doermann and Rosenfeld 92], [Boccignone et al 93]. The writing order produced by such

analysis may differ from the original writing order. This may lead to failure of the recognition process. Therefore, it is beneficial for on-line recognition to obtain the temporal information at the time of writing the images. This information can consist of the number of strokes, the order of the strokes, the direction of the writing for each stroke and the speed of the writing within each stroke. A stroke is the writing from pen down to pen up. However, temporal information of on-line systems complicates recognition because of having a lot of variations for a specific letter.

The on-line recognition task can be divided into three main categories:

- Handwriting recognition
- Gesture recognition
- Symbol and drawing recognition

Handwriting recognition may replace the main function of the keyboard. The user writes the data with a pen on a special device instead of typing. The term gesture refers to hand markings, such as circles, brackets and arrows, that function to indicate scope and commands. The more usual menu-oriented operations of pointing and selecting are also normally considered as gestures. Gestures are used as an alternative means of using mouse and auxiliary functions of the keyboard. A typical set of gestures are editing gestures. Symbol and drawing recognition facilitates the entry of arithmetic formulae and geometric shapes, which is usually very difficult and complicated when using the keyboard.

Over the last few decades, a significant amount of work has been done in the area of on-line handwriting recognition including [Wright 88], [Wells 92] and [Powalka 95]. As the present research is not concerned with on-line recognition,

we are not discussing the area in detail. More details about research problems and developments can be found in the above mentioned references.

2.3.2 Off-line or Static recognition

Off-line recognition, which is the field of the present research, is performed after the completion of writing or printing. Off-line recognition processes a scanned image of shapes such as line drawings, prints, handwriting and other graphic images to be recognized. In this recognition, the input is devoid of temporal information - the information about the order in which the shapes were drawn. In other words, in off-line recognition, one does not know how a particular shape is drawn.

In a typical OCR system, the input characters are read and digitized by an optical scanner. The scanned images may contain superfluous or misleading information. The superfluous information may occur because of a poor document quality (e.g. dirt, dust etc.) or may come with the document itself (e.g. guideline on checks, bases on printed form, columns of the tables etc.). The recognition algorithms must be able to identify such superfluous information and cope with them accordingly. The accuracy of the recognition is affected by the quality of the image scanned.

After scanning a particular document to be recognized, each object of the document is then located, segmented and the resulting matrix passed to a preprocessor for smoothing, noise reduction and size normalization. However, preprocessing of the image may lead to loss of some important information and thus the present research avoids a preprocessing stage. After preprocessing a particular object, the preprocessed object is passed to the features extractor. Here distinctive features are extracted from the preprocessed object for classification. Later, these features are compared with predefined features of different characters.

A character or a number of characters with maximum matching of features are considered as the recognized input object.

Off-line recognition can be further divided into two types

- Optical Character Recognition (OCR)
- Magnetic

The present research deals with OCR, and therefore only this technique will be described in the following sections.

2.3.2.1 Optical Character Recognition (OCR)

Optical character recognition can further be divided into the following three different types.

- i) OCR for machine-printed documents
- ii) OCR for hand-printed documents
- iii) OCR for poor quality documents

Since the present research is primarily concerned with OCR for poor quality documents, we only reviewed this type of OCR. Research in the area of machine-printed and hand-printed OCR can be found in Appendix A and Appendix B. However, it has been observed that existing OCR algorithms and systems perform very well on good quality machine printed text which contains minimum fragmented or touching characters. A recognition rate of more than 99% has been achieved using some of these systems. However, these systems do not perform well on poor quality documents (documents containing a lot of touching and broken characters) and recognition rates decrease dramatically as the quality of the

document decreases. Humans can recognize very degraded text. Hence there is a big gap between human and machine capabilities of reading text. The ultimate aim is to obtain a recognition rate as close to the human being as possible.

OCR for poor quality documents

The third generation of OCR systems are related to the recognition of poor quality documents [Mori et al 92]. By the end of 1960's the targets of document readers were for poor print quality characters, and hand printed characters for a large category character set, such as Chinese. These targets have been achieved partially and such commercial OCR systems appeared roughly during the period from 1975 to 1985 [Mori et al 92]. Low cost and high performance are always common objectives for the systems. The dramatic advances of Large Scale Integrated (LSI) technology were of great help to the engineers who were engaged in the development of OCR systems. Although this was common to all electronic systems, in general LSI was especially important for pattern recognition systems.

Poor quality documents appear very common in practical life. They come from different sources such as scanned old documents, low quality prints, photocopies and facsimile messages etc.

[Jagota 90] developed and analyzed a Hopfield-style network model for degraded text recognition. This model is applied towards machine printed word recognition. Words to be recognized are stored as content-addressable memories (CAM). Word images are first processed by an OCR system. The network is then used to postprocess the OCR decisions. It has been reported that the network functions reasonably well as content-addressable memories for dictionaries of up to 500 words and a recognition rate of 85% is achieved. A network trained on larger dictionaries exhibited drastically worsened CAM performance, but they still

make most letter decisions correctly. Hence they can be used well as filters even for larger dictionaries. The noise removal option makes conservative but more reliable decisions. Independent of the size of the dictionaries on which the networks were trained, deriving hypotheses from a confusion matrix of the OCR always gave the best performance.

One major advantage of such networks for this kind of application is that they operate in real-time, even when simulated on a sequential computer. Digital hardware implementations of such a network are also feasible.

Words in the images were machine printed, but of low quality. However, it has not been reported in terms of quality values. Apart from this, font and size of the words is also unknown. Networks trained for larger dictionaries (above 500 words) did not perform well.

Word image matching as a technique for degraded text recognition was presented by [Hull et al 92] and concentrates on the clustering algorithm. The clustering process determines equivalence classes among word images in a passage of text. Initial hypotheses for the identities of words are then generated by matching the word groups to language statistics that predict the frequency at which certain words will occur. This is followed by a recognition step that assigns identifications to the images in the clusters.

The performance of this technique on a running text of 1062 word images is determined. These sample words were printed in an 11 point Times Roman font on plain paper by a laser printer. The resultant pages were then scanned at 200 pixels per inch in 8 bit grayscale on a desktop digitizer and binarized. It is shown that the clustering algorithm can correctly locate groups of short function words with better than a 95 percent correct rate. It is reported that future work will include

investigation of multiple feature sets and the use of other knowledge sources, such as language syntax and semantics to improve the performance of this technique.

The technique described above claims to be tolerant to a wide range of image noise and is thus ideally suited for the recognition of degraded word images. However, by looking at the definition of the sample words, it appears that the technique has not been tested on poor quality documents for example facsimile messages.

[Bose and Kuo 92] applied Hidden Markov Model (HMM) and level-building dynamic programming algorithms to the problem of robust machine recognition of connected and degraded characters forming words in a poorly printed text. A structural analysis algorithm is used to segment a word into sub-character segments irrespective of the character boundaries, and to identify the primitive features in each segment such as strokes and arcs. The states of the HMM for each character are statistically represented by the sub-character segments and the state characteristics are obtained by determining the state probability functions based on training samples. A level building dynamic programming algorithm combines word-segmentation and recognition in one operation and chooses the best probable grouping of characters for recognition of an unknown word.

Approximately 75 words were generated at various degrees of overlap and blur values, using Baird's defect generator [Baird 90]. However, in order to examine the effectiveness of the system, tests were carried out in mostly medium and high noise conditions, i.e. high blur values were used. A SUN-sparc station-1 was used for training and testing the recognition system. The segmentation and feature extraction takes approximately 0.25 second per character. Although the segmentation is automated, the labeling of the segments in the training mode is supervised, that is, labels are manually corrected, as appropriate. The clustering

algorithm takes approximately 40 seconds to grow from 15 to 32 cluster centers for a training feature set of approximately 2000. The recognition phase of the algorithm takes approximately 0.4 seconds per character. The experiments demonstrate the robustness and effectiveness of the new system for recognizing words formed by degraded and connected characters.

[Hong and Hull 94] proposed a relaxation based algorithm for degraded text recognition using word collocation. This algorithm improves the performance of the text recognition technique by propagating the influence of word collocation statistics. Word collocation refers to the likelihood that two words co-occur within a fixed distance of one another. For example, in a story about water transportation, it is highly likely that the word “river” will occur within ten words on either side of the word “boat” [Hong and Hull 94]. The proposed algorithm receives groups of visually similar decisions (called neighborhoods) for words in a running text that are computed by a word recognition algorithm. The positions of decisions within the neighborhoods are modified based on how often they co-occur with decisions in the neighborhoods of other nearby words. This process is iterated a number of times effectively propagating the influence of the collocation statistics across an input text. This improves on a strictly local analysis by allowing for strong collocations to reinforce weak (but related) collocations elsewhere.

The recognition data used in the experiments were generated from the Brown Corpus and Penn Treebank databases. These are large corpora that together contain over four million words of running text. The Brown corpus is divided into 500 samples of approximately 2000 words each [Kucera and Francis 67]. The part of the Penn Treebank database used here is the collection of the articles from the Wall Street Journal that contains three million words. They used the frequency of a word pair to measure its collocation strength. There are 1,200,000 word pairs after

training. No information about the font, point size and quality of the words is given. The neighborhoods were generated for each word by first calculating a feature vector for the word known as the stroke direction feature vector [Ho et al 92]. The number of neighborhood words for each word is also unknown.

In order to evaluate the performance of the algorithm, five articles were randomly selected from the Brown Corpus as the testing samples. There were totally 10,280 words in those testing samples. For each word in those texts, the top10 word candidate lists were generated. A word recognition algorithm, based on different performance models, was stimulated. The performance models used had top1 correct rates of 55%, 65%, 70%, 75%, 80%, 85%, 90% and 95%. It has been reported that the correct rate of first choice is around 95% in all conditions.

The proposed relaxation algorithm currently works as one part of their degraded text recognition system. There are two types of linguistic constraints used in the system. One is local word collocation under statistical language modelling. Another is global structural constraints carried by English grammar. Visual global contextual information available inside a text page is also being considered for integration with the linguistic knowledge sources to further improve the performance of degraded text recognition.

The developed word collocation technique can be used as postprocessing to improve the recognition rate in conjunction with the recognition methods which give a number of alternative words for each sample word.

[Rocha and Pavlidis 95] proposed an approach for segmentation-free recognition specially suited for the processing of touching and broken characters. This is accomplished by recognizing subgraphs even if they have gaps or strokes from other adjacent characters. The method allows the recognition of characters that

overlap or that are underlined. The character recognizer uses a flexible matching between the features and a flexible grouping of the individual features to be matched. Broken characters are recognized by looking for gaps between features that may be interpreted as part of a character. Touching characters are recognized because the matching allows non-matched adjacent strokes. This technique was applied to 5,282 words (over 24,000 numerals) belonging to a United States Postal Services (USPS) database of real printed addresses. The recognition results are: 96.5% of the words were correctly segmented, and 91.8% of the characters were recognized.

[Al-Badr and Haralick 94] reviewed and discussed techniques to handle noisy and connected text. They laid out the strategy of a symbol recognition method that does not require a prior segmentation stage, and hence avoids some of the limitations of the segmentation based techniques. In this system segmentation into symbols is a by-product of the recognition process. The system has three major components: The primitive detector, the matcher and the global control module. They explained the processing done by each component and how the modules interact. This system contributes in three major areas; (i) Robustness: by globally optimizing the process of combining primitives into symbols, it is robust and less sensitive to noise, (ii) Recognition without prior segmentation: does not require segmenting in advance a block into lines, a line into words, nor a word into characters, segmentation is a byproduct of recognition, (iii) Language independence: training determines the symbol set it recognizes. It is mentioned that this paper lays out the overall strategy of a system that implements the recognition and a following paper will report on experimental protocols and results.

[Ricker and Winkler 94] described a system which assists in processing faxed documents. The system is designed to receive order forms via fax, identify the form, extract the appropriate data and present the data to a host computer. The types of data recognized are handprinted characters, machine printed characters, and marksense fields. A model of each form is created from the original image and stored in a form model database. The processing of the form consists of four steps: object extraction, form identification, form registration, data extraction (including character recognition). Once all the data is extracted it is sent for postprocessing where the data is corrected for errors in the recognition phase with respect to contextual dependencies and application specific dictionaries and then to the host computer where the order is confirmed. The system described is presently being used at numerous sites. Results from the sites confirmed the recognition rates obtained by the in house testing. The overall results vary according to how the system is used. Factors such as form complexity and post processing availability affects the overall performance of the system. One site reports that 60% of the images were processed without requiring any operator invention. This represents a 93% reduction in the number of keystrokes that would be required to transcribe a fax manually.

[Fang and Hull 95] presented a modified character-level deciphering algorithm for OCR that has the ability to solve the touching characters problem caused by document degradation, and is tolerant to mistakes in clustering. Visual constraints are systematically combined with language constraints to decipher touching patterns and to detect and reverse clustering errors. The deciphering algorithm was tested on both artificially created and scanned degraded documents with extensive occurrences of touching patterns and clustering mistakes, and achieved satisfactory results in both cases.

[Lam et al 95] proposed an approach that addresses the problem of recognition of touching characters by forming a closed loop system between segmentation and isolated character classification for mutually beneficial feedback. The method uses a variable window sliding throughout the word and results in a tree structure with intermediate nodes representing validated characters. Some experiments were conducted on the touching parts of the word images from journals and newspapers. The method gives very good results on the touching extracts from these word images. The current test images used are from a photocopy of a facsimile document scanned at resolution of 300 ppi. In most of the cases almost all the characters touch each other. The lexicon is made up of the 24469 words contained in a standard Webster dictionary and is represented as a trie structure. The building time for the trie is 2.55 seconds on Sparc II. Various peak finding algorithms (can use geometric constraints along with the slope information) were tried for initial hypothesis generation from segmentation confidence and almost more than 95% of the time the initial hypothesis did not miss the actual segmentation point. It has been reported that problems were due to a few ligatures like 'ff', 'ffi' etc.

Varied recognition rates have been mentioned for different systems outlined above. However, it is difficult to compare the effectiveness of these systems when the documents and methods used vary from system to system. Achieving a higher recognition rate does not necessarily means that the system performs well, as the data might be of good quality. Similarly a system with low recognition rates may not be a poor system, as the data might be of poor quality. In order to compare the effectiveness of the various systems the same test data conditions must be used.

In the present research, we have tried to evaluate the performance of the developed recognizer by comparing its results with the results of an other system (a commercial OCR product) using the same data.

2.4 Character recognition problems

There are many problems related to machine-printed and hand-printed character recognition. The most important ones are as follows.

1. Shape discrimination

A single character has a large number of fonts, e.g. Courier, Helvetica, Times Roman, Elite, Gothic, Orator, some special OCR fonts and a lot of free styles and thousands of handwriting shapes which can represent it. This can be seen in Figure 2.3.



Figure 2.3: Different fonts and standards

2. Similar shapes

Different characters can also have similar shapes. Some examples are U and v, o and 0, s and 5, z and 2 etc.

3. Deformation of the image

Deformation of the image of a character is caused by the following well known factors:

i. Noise

This causes unconnected line segments, holes and breaks in lines, isolated dots, bumps and gaps in lines, filled holes etc.

ii. Translation

This is because of the movement of a whole character or its components.

iii. Rotation

Rotation can be caused by documents being scanned at slightly incorrect angles.

iv. Distortion

Distortion includes local variation, rounding of corners, improper protrusions, dilation and shrinkage, etc.

4. Variation in sizes and pitch

The pitch of 10, 12 or 17 specifies that there are 10, 12 or 17 characters per inch (cpi). 10 pitch characters are usually bigger in both width and height than those in 12 pitch.

A single character has a variety of point sizes e.g. 10, 12, 14 point size etc. As the size of the character increases, the shape of the character becomes bigger both in width and height. A few characters in different sizes can be seen in Figure 2.4.

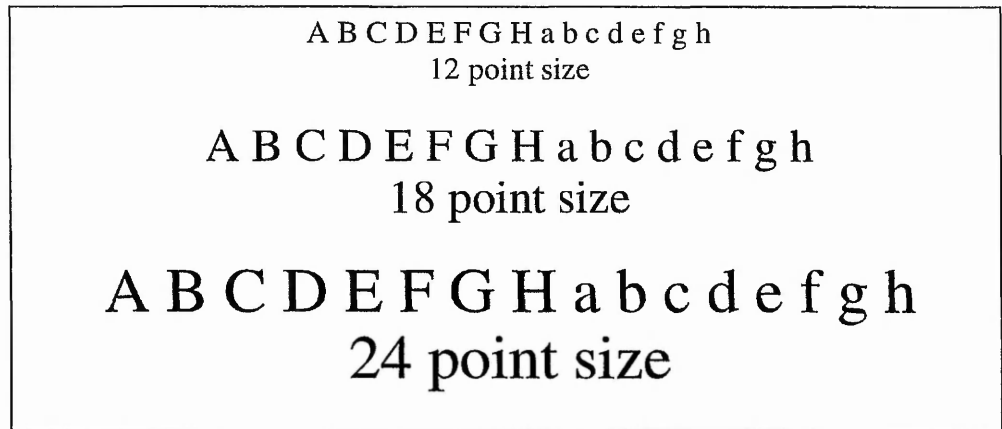


Figure 2.4: Few characters in different point sizes

5. Touching characters

It appears that some characters always touch together when written in a particular font and point size. Apart from this, scanning of documents can also lead to touching of characters, especially when scanned at low resolution.

Poor quality documents, e.g. low quality prints, facsimile messages have a large number of touching characters.

6. Broken characters

Again, poor quality documents, especially facsimile messages have some objects broken into different pieces. The recognition of broken objects is challenging task.

In addition to the above mentioned problems, an optical character reader must be able to distinguish figures from text, recognize touching characters and must be unaffected by proportional spacing and variable line spacing.

2.5 Applications of character recognition technology

Significant effort has been spent on character recognition not only because it is a challenging problem, but also because it provides a solution for processing a large volume of data automatically, e.g. postal code reading [Fukushima et al 91], [Genchi et al 68], [Genchi et al 70], [Mori et al 70], [Neill 69], [Hendrawan and Downton 94], [Downton et al 95]. It also has a large variety of business and scientific applications such as interactive digitization of sounding values on charts [Bolton and Bayle 77], alternatives to hand-printing in the manual entry of data [Devoe 67], automatic recognition of print and script [Harmon 72], recognition of handprinted characters for automated cartography [Lybanon and Gronmeyer 78], CAD recognition [Waite 89], [Shaw 94], [Thomas et al 95], [Poliakoff et al 95], recognition of handprinted text [Munson 68], computer reading of meter reader's handwriting [Webb and Kreutzer 72], etc.

The most important use for optical character recognition stems from the general activities directed towards office automation which dominates information processing [Impedovo et al 91]. Intensive research has made OCR an efficient means of entering data directly into the computer and capturing information from books, sheets and other handprinted and machine printed material.

Optical character recognition technology has many practical applications. Some of the literature covering applications of character recognition technology are in languages other than English, namely, German, Japanese etc. The following are some of the applications for which OCR have been used or suggested by a number of research workers.

- Its use in postal department, for postal address reading and as a reader for handwritten and printed postal codes [Swonger 69], [Genchi et al 68], [Genchi et

al 70], [Notbohm and Hanisch 86], [Focht and Burger 76], [Neill 69], [Lecolinet and Moreau 90], [Downton et al 91], [Hendrawan and Downton 94], [Cohen et al 94], [Strathy and Suen 95], [Downton et al 95] and [Bertille and Gilloux 95].

- Its use by blind people, as reading aid using photosensor and tactile simulators and a sensory aid with sound output [Bliss 69], [Smitch 73], [Badoux 85], [Harness et al 93], [Sherkat et al 93]. It can also be used for reading and reproduction of braille originals [Spronsen and Bruggeman 85].
- Its use in the publishing industry [Skalski 67] and as a reader for data communication terminal [Genchi 69].
- Its use as a telecommunication aid for deaf [Kondraske and Shennib 86].
- This technology can be used for giro services, such as for giro document reading, sorting and ledgering, and for reading giro orders [Haaley 69].
- This technology may have an effective use for character print quality analysis/ measurement [Crawford 72], [Throssell and Fryer 74], in air-line reservation systems [McAbee 67], and in motor vehicle bureau as automatic number plate reader and recorder for road traffic control [Gyarfas 74].
- It may be used in health insurance data acquisition [Timm 73].
- Its use for digital bar code reading [Nassimbene 72], and as a handwriting analyzer for automatic writer recognition and signature verification [Kupriyanov 72], [Sternberg 75].
- It can be used for business applications, such as financial applications like cheque sorting strategy optimization [Murphy and Stohr 75], [Guillevic and Suen 95], [Dodel and Shinghal 95], [Lethelier et al 95].

- Its use for direct processing of documents, as a multipurpose document reader for large-scale data processing, as a microfilm reader data input system, for high speed data entry, for changing text/graphics into a computer readable form, as electronic page reader to handle large volume of mail [Ufer 70], [Kroger 87], [Vossen 86], [Amiri et al 94], [Downton et al 95].
- It may also be used in law enforcement applications [Joshi 74], in educational administration (such as examination assessment and attendance record evaluation) [Hemphill 75], and mark sheet reader for payroll accounting and book-keeping [Christ and Schrag 76].
- Another use of this technology is in customer billing, as in telephone exchange billing system [Yoshida 74], order data logging [Hilgert 70], as an automatic inspection system for I.C. mask inspection and defect detection in microcircuits [Bojman 70] and as a credit card scanner in credit personal identification systems [Herst and Liu 80].
- This can be used in automated cartography [Gronmeyer 79], [Lybanon and Gronmeyer 78], metallurgical industries [Pokluda 77], computer assisted forensic linguistic systems [Perret 80], electronic mail [Polizzano 83], information units and libraries, and for facsimile messages [Smith and Merali 85], [Ricker and Winkler 94], [Raza et al 97b].
- It may also be used for shorthand transcription [Leedham and Downton 86], [Leedham and Downton 87], and in electronic package industries [Berger et al 85], and reading characters stamped on metallic parts [Nakamura et al 86], [Nakamura et al 87].
- Its use for optical census [Ress 75], and for control of outside workers in sales and distributions [Schacht 78].

- This technology may be very suitable for mechanized document reading in textile and clothing manufacture enterprises [Schafer 73], automatic punching of industrial telegraphs [Inoue et al 73], retail data processing applications in food enterprises and for retail product code name and price reading techniques [Eggiman 74].
- One of the most common applications is checking the date and a lot of stamps on drugs and food [Cook 92]. Food and Drug Administration is shifting towards machine-inspected lot numbers and expiration dates for drugs and food. Another increasingly important application is in graphic arts, where the characters themselves are the output. There are other areas where the combination of limited space and the need of human and machine readable information makes OCR attractive. One example is semiconductor packaging. There isn't much space on top of a dual in-line pin (DIP) package, and other packages may have even less. Semiconductor wafers are being labelled for reading by OCR. This is appropriate because semiconductors are helping to spread OCR further into the market place.

2.6 Summary

This chapter describes a literature review in the area of OCR. Historical background of OCR is presented. General classification of the character recognition is presented and described. The research in the area of authors's interested (OCR for poor quality documents) is described in detail. Different problems related to machine-printed and hand-printed character recognition are also mentioned. A number of industrial and commercial applications of this technology are presented.

Chapter 3

Object extraction

3.1 Introduction

In the present research we are dealing with whole word recognition methods in order to avoid segmentation of touching characters. The text stored in a TIFF (Tag Image File Format) files must be separated into individual objects and then words. This chapter details the methods used for this task.

We describe the method for finding text lines in a given image. The edge-following algorithm, used to locate the position and size of the objects in a line is then outlined. Following this, different techniques for finding objects from the line of text are presented. A method for finding word gaps is described, and the chapter concludes with description of a word finding method.

The analysis of automated page layout, that is preprocessing of the page, is a research problem in itself. Much work has been done and published in this area. For example, [Spitz 93] and [Kanai 90] have done work in identifying the columns

of printed text from scanned images, identifying lines within a column and finding word boundaries within a line.

In the present research work, we are dealing with the recognition of poor quality documents, and therefore we use only minimal layout analysis. These are developing basic methods for finding text lines, objects in a line and word boundaries.

3.2 Finding text line

In order to perform OCR effectively, it is necessary to have a scanning resolution of at least 300 dot per inch (dpi). Resolutions lower than this may lead to intolerable increases in the numbers of broken and touching characters. This gives us an image size of 2520 x 3564 pixels, for a full A4 page including margins, thus requiring 1.1 Mb of memory. The current programming environment is UNIX.

After loading a complete TIFF file in memory, the text lines are found using the following simple method. Note that it is assumed that text lines are neither skewed nor overlapping.

A left to right, top to bottom search in a line of TIFF file is made until a black pixel is found. This gives the top edge of the line denoted by T_e , as shown in Figure 3.1. A second search, starting at the left edge of the TIFF file, one pixel below the top line, is made, until a complete row of white pixels is found. A line above this line gives the bottom edge of the line denoted by B_e , as shown in Figure 3.1

The difference between B_e and T_e is calculated using (EQ 3.1). This difference is called line height, denoted by L_h and can be seen in Figure 3.1.

$$L_h = T_e - B_e \quad (\text{EQ 3.1})$$

If L_h is less than 5 pixels, then it is assumed that this is not a text line. It is considered to be noise or some dirt on the page. There is no font size of such height.

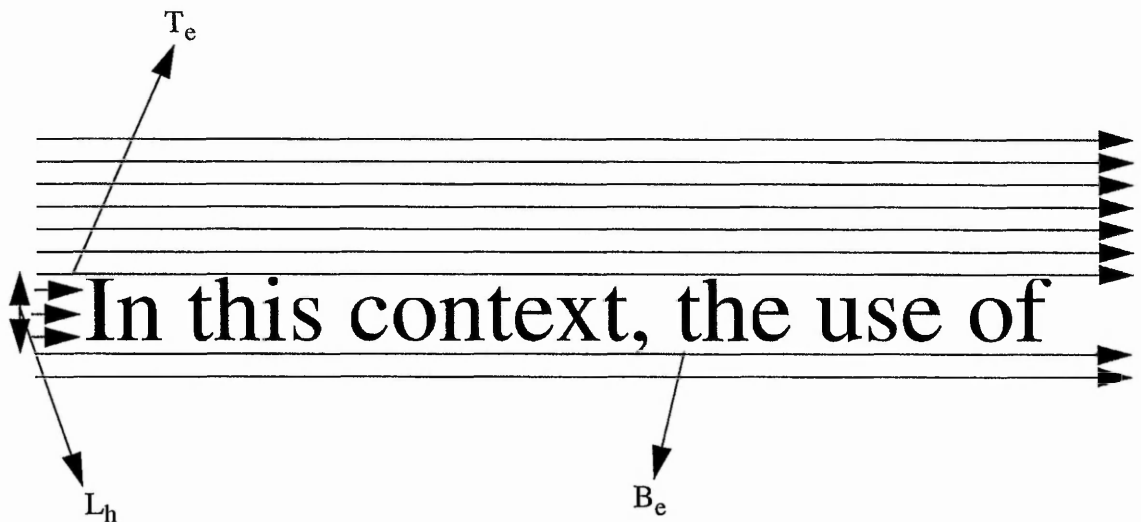


Figure 3.1: Searching for lines of text

Similarly, we may find a very large difference between the bottom and top edge of a line, i.e. we get a line of excessive height say 200 pixels. Again this line is ignored and it is considered to be an improper text line since we assume there is no font size that big. This might be a figure, or it may be the case that the document is of very bad quality and has different lines of text touching together or it is composed of miscellaneous objects, i.e. text, diagrams and figures etc.

The search for the next line in the TIFF file then continues, left to right, downwards, starting on the left, one row below the bottom line of the last line.

The whole process continues until all lines have been found.

A simple method for finding text lines in a given document image is described. The method is capable of extracting text lines provided that lines are well separated from each other and non skewed. If text lines are overlapping, skewed, or touching together, then the developed method cannot extract text lines properly.

3.3 Edge-following algorithm

Outlining is a method of converting any object (in this case a given piece of text) of several pixels width into a single pixel width object (piece of text) by tracing the outside of the object. To get an outline of the object, the following edge-following algorithm is used to trace around the given object. The coordinate rectangle (smallx, smally, largex, largey) enclosing the area where an object is located can be found using this algorithm (Figure 3.2). This rectangle is just large enough to contain the object. Smallx is the left most position of the object and largex is the right most position of the object. Similarly, smally is the bottom most position of the object and largey is the top most position of the object.

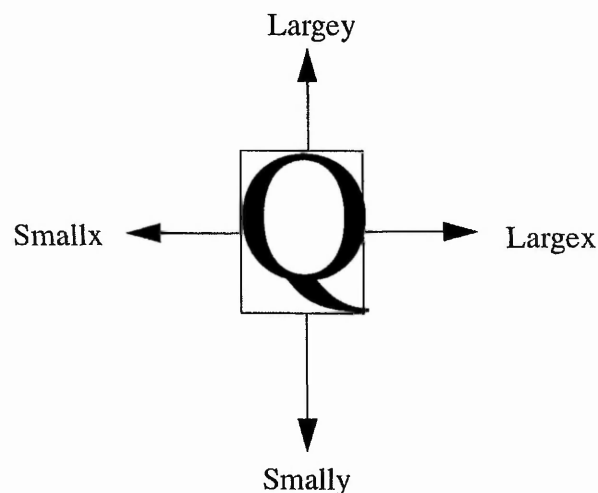


Figure 3.2: Rectangle coordinate of an object

The algorithm must first be given an x, y pixel position adjacent to the edge of an object. That is, before we use this algorithm, it is necessary to locate a white pixel adjacent to an edge pixel of an object. This is done using the object finding methods explained in the next section. It also requires, one of its four movement vectors, as an initial movement vector, i.e. left, right, up or down (1 if an initial movement vector is left of object, 2 if down, 3 if right and 4 if up). The movement vectors specify anti clockwise movement around an object, 1=down, 2=right, 3=up and 4=left. This is explained in Figure 3.3.

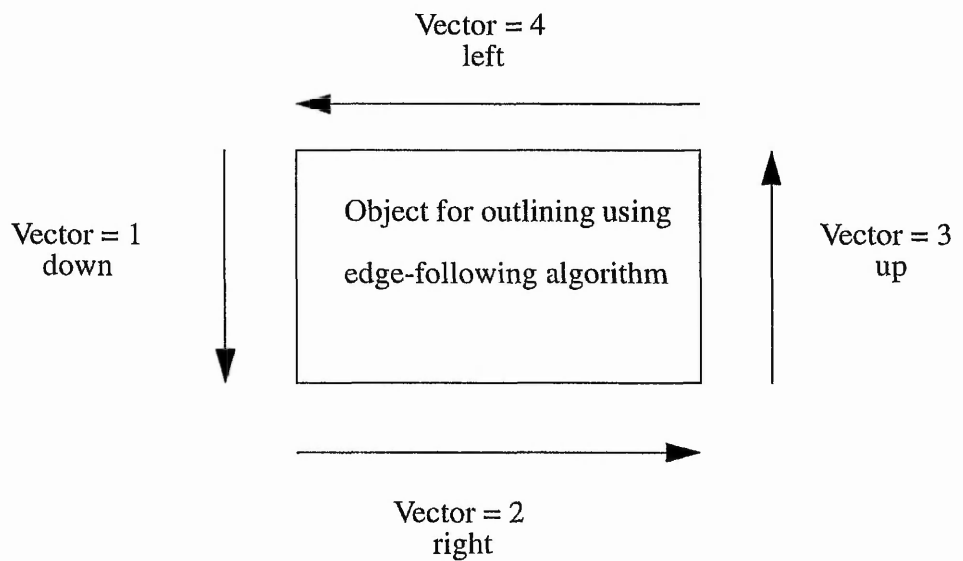


Figure 3.3: The movement vectors describing anticlockwise movement

Given these parameters (x,y coordinate of a white pixel and initial movement vector), the algorithm will then trace around the outside (not on the edge itself) of the object.

The algorithm has three steps:

Step 1. Find the 'next pixel position' to move

Step 2. Adjust the movement vector

Step 3. Accumulate the co-ordinate rectangle

These three steps are executed repeatedly for each position an 'imaginary pixel' takes as it travels around the edge of an object. This process continues until 'pixel' returns to its starting point indicating that the whole object has been outlined. At this point the algorithm terminates. Therefore when the object has been fully followed, the four variables contain the coordinates of a rectangle fully enclosing it (see Figure 3.2).

An edge-following algorithm is described above. The algorithm is very useful and an important step in the recognition of an object, as it gives the size and position of the object. It can also be used to find holes in a given object. The algorithm is capable of outlining objects of different sizes, shapes and kinds.

3.4 Finding objects

3.4.1 Finding objects from bottom to top

The aim of this method is to locate each object on the line. This method involves looking for an object by searching from bottom to top, and then along the line.

A bottom to top, left to right search is made within the line top and bottom edge boundaries, starting at the bottom left corner of the line, as shown in Figure 3.4. When a black pixel is found, the edge-following algorithm is used to find the bounding rectangle of the object of which that pixel is part. The search for another black pixel then recommences starting at the bottom line, one pixel right of the right edge of the last bounding rectangle found.

The process continues until all objects on the line have been found and their bounding rectangles established.

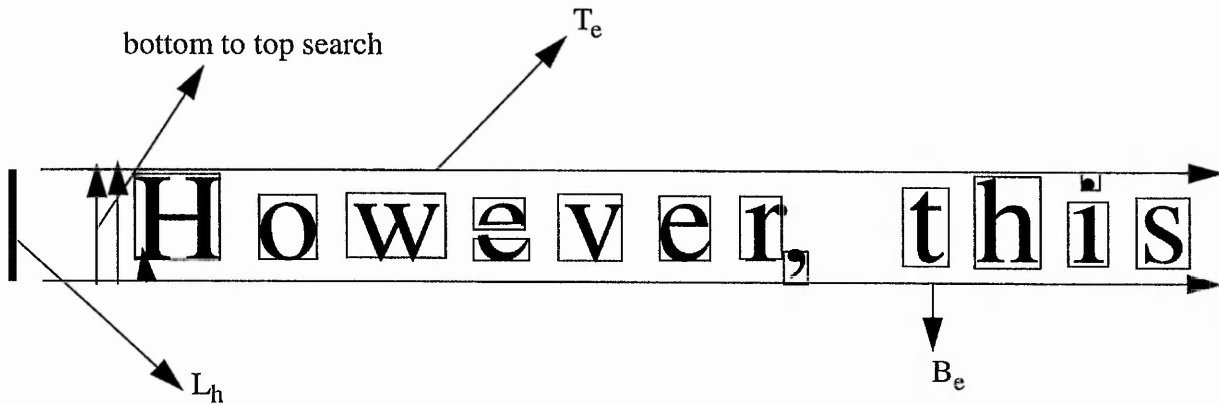


Figure 3.4: Finding objects from bottom to top

Using the above method, we can automatically search for objects and outline them.

However, if we have a document, in which some objects are contained in another object, then the developed method will not be able to extract all objects. The example of such a document is a table containing text inside it (Figure 3.5). In order to find all objects from such a document image another method has been developed, which is described below.

Quantity	Product & Description	Unit Price
1	TDMB436 <ul style="list-style-type: none"> ● Monochrome framegrabber/8 bit RGB display ● Two 1024 x 1024 video stores ● 1024 x 1024 overlay plane ● TMS320C40 processor ● 4MBytes zero wait state local memory and global bus expansion connector ● Programmable capture and display up to 1024 x 1024 pixels ● RGB/composite/S-Video/mono capture ● Trigger input & output for event sync. 	£4550.00

Figure 3.5: An image in which method fails to extract all objects

3.4.2 Finding objects by colouring connected components

The input is the document image, and the aim is to locate each object (an object being a character or multiple touching characters, or some other group of black pixels). A sample document image is shown in Figure 3.6.

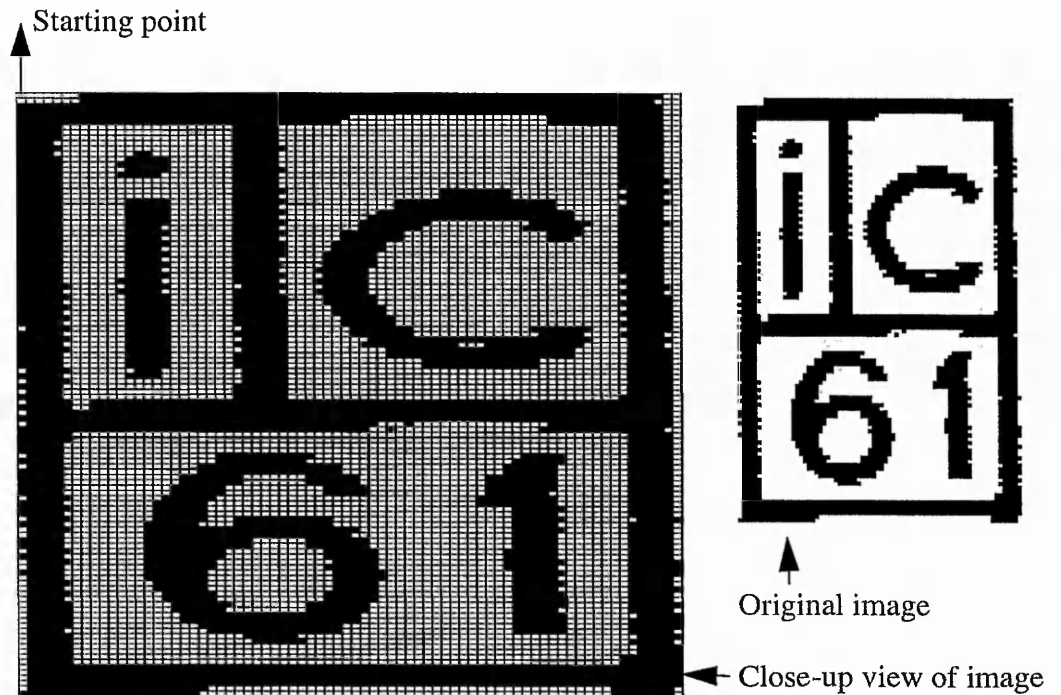


Figure 3.6: A sample document image

A starting corner of the image is located. This is given by the two directions (see Figure 3.7) passed as parameters. From this corner a search is made until a black pixel is found. The order of the parameters determines the direction of this search

For this example, the two directions given are 0 and 6, giving the top left corner as the start point. From here, a left to right, top to bottom search for black pixels is made. When a black pixel is found, its location is noted, and called the seed point (see Figure 3.8(a)). All black pixels adjacent to this in any direction are coloured to

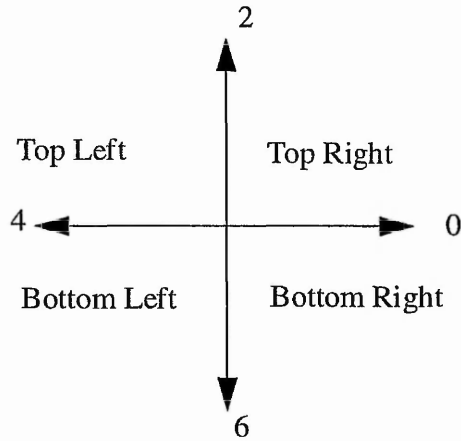


Figure 3.7: Direction diagram

white, as is the seed point pixel until all connected black pixels are coloured (see Figure 3.8(b)).

Once colouring of the object has been taken place, the bounding rectangle of this object is known (see Figure 3.8(c)). It has been calculated within the colouring process. We now have the situation where the bounding rectangle is established, and in effect, that object deleted from the document image. It can no longer interfere with the location of the subsequent objects.

Now, the coordinates of the seed point are recalled for resumption of the search. The same left to right, top to bottom search is made commencing from the last seed point. When another black pixel is found, black to white colouring occurs once again. The process continues until there are no more black pixels in the document image. The sample image after the complete process of colouring and bounding rectangles obtained are shown in Figure 3.9.

We now have bounding rectangles for each object along with their seed points in the order of colouring as shown in Figure 3.10.

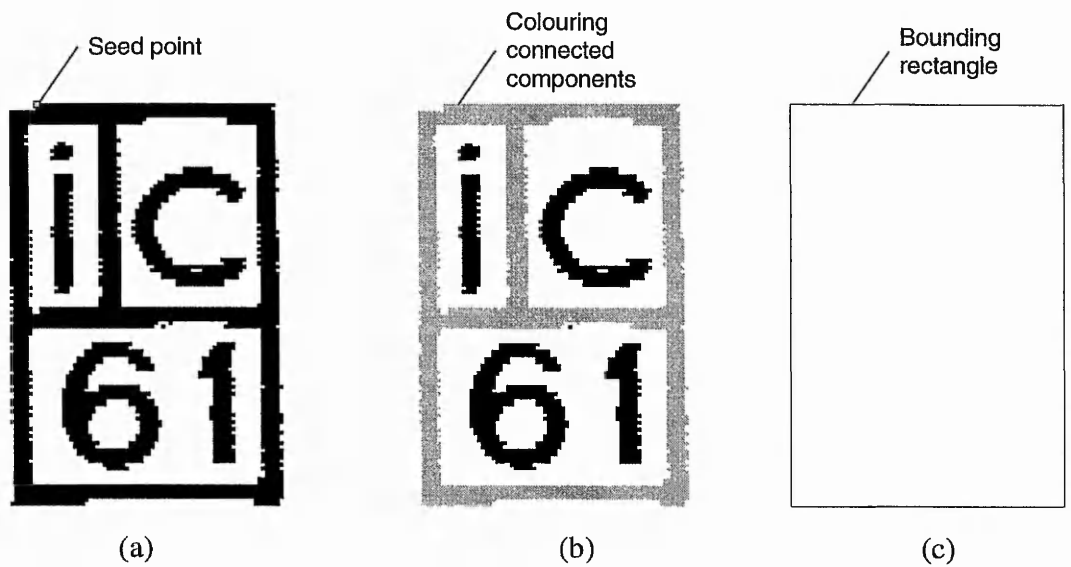


Figure 3.8: Seed point, colouring connected components effect and bounding rectangle

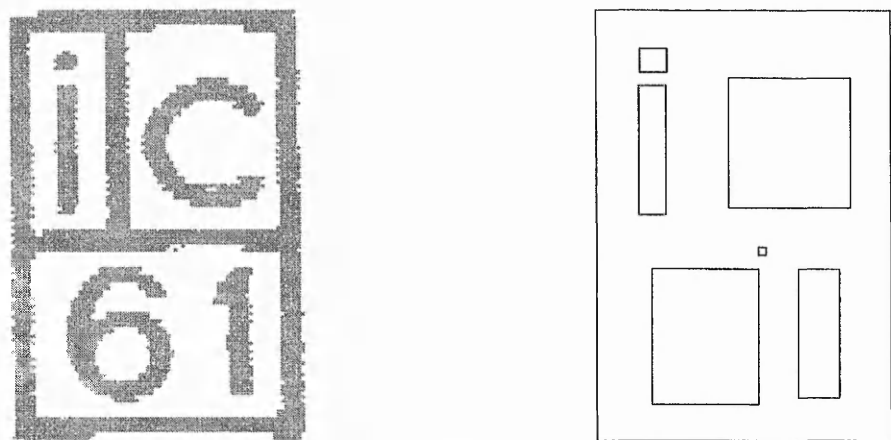


Figure 3.9: Sample image after complete colouring process

Removing huge and tiny objects and obtaining writing order

After finding the bounding rectangles for each object, filtration is carried out. This step removes very small objects (noise etc.) and very large objects (tables, long horizontal and vertical lines etc.), which are not proper text objects. Such objects are considered to be invalid objects, mainly due to their size. The effect of this step is shown in Figure 3.11.

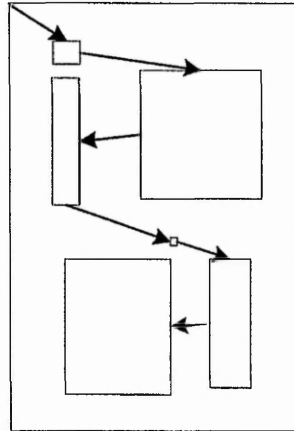


Figure 3.10: Order of the objects after colouring process

The object rectangles obtained from the colouring process are not necessarily in a suitable writing order as seen in Figure 3.10. The object rectangles are first sorted by decreasing y values in order to identify text lines. The rectangles in each text line are then sorted by increasing x values to put the objects into the order of occurrence in the text line. The order of objects after obtaining writing order is shown in Figure 3.11.

The ordered objects are then rejoined together to make words using inter object gaps.

A method for finding objects in a given image is presented. The method is capable of finding all objects in any kind of image. It can locate the objects within another object, and hence solves the problem encountered by the previous method (see Figure 3.5) and described in Section 3.4.1. The colouring connected method gives the object bounding rectangles along with their seed points as required.

The application of this method on the image (see Figure 3.5), is shown in Figure 3.12. Table (shown in light black color) has been removed primarily because of its size. The text inside the table (shown in dark black color) is left as a proper text for

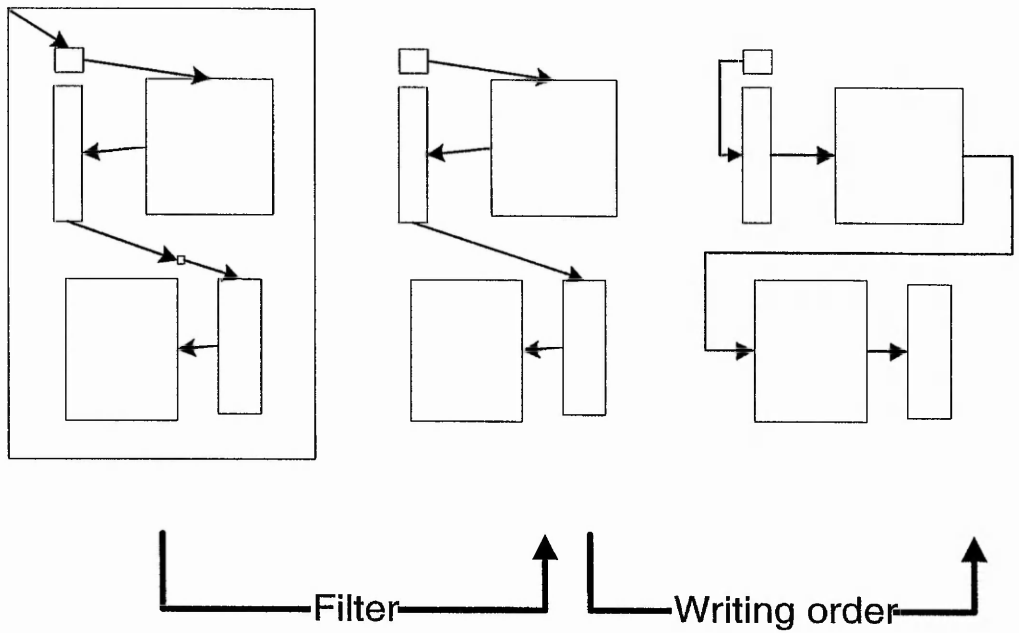


Figure 3.11: Removing huge and tiny objects and obtaining writing sequence

recognition. The method however, also removes all objects connected to the table as shown in the bottom of the table (characters 'gg', 'p', 'p', 'yn' all touch the table border and thus removed).

Quantity	Product & Description	Unit Price
1	TDMB436 <ul style="list-style-type: none"> ● Monochrome framegrabber/8 bit RGB display ● Two 1024 x 1024 video stores ● 1024 x 1024 overlay plane ● TMS320C40 processor ● 4MBytes zero wait state local memory and global bus expansion connector ● Programmable capture and display up to 1024 x 1024 pixels ● RGB/composite/S-Video/mono capture ● Trigger input & output for event sync. 	£4550.00

Figure 3.12: Sample image after removing huge objects

3.5 Finding word gap

The present research deals with whole word rather than single character recognition method. Therefore, it is necessary to be able to detect the position of inter-word gaps. When we encounter such a gap, we must finish reading further characters from the line. This facilitates reading of one word at a time from a given text line.

We have assumed that text is spaced in an orthodox manner with similar sized gaps between words. Therefore we have used a very simple method to separate words from each other. More detailed work on this topic, i.e. finding lines and word boundaries, can be seen in [Kanai 90] and [Spitz 93]. The method developed for finding word gap in the present research is described below:

We start at the bottom left corner of the line, and search upwards until an object is found, or the top of the line encountered. If the latter occurs, we move one pixel to the right and continue from the bottom of the line.

When a black pixel is reached, an object has been found, and so the x and y coordinates of the black pixel are noted. An edge-following algorithm is then used to trace around the outer border of this object, and the extreme left (EL), extreme right (ER), extreme top (ET) and extreme bottom (EB) of the object are noted. The search continues until the next object has been outlined. Now the difference between the adjacent edges (G_o) is calculated using (EQ 3.2), as shown in Figure 3.13. This gap is then recorded.

$$G_o = \text{EL value of current object} - \text{ER value of previous object} \quad (\text{EQ 3.2})$$

The next gap is then found by finding the next object, and comparing its extreme left edge with the extreme right edge of the previous object. Again this value is recorded.

This procedure is repeated until the last object has been found and the gap between it and the previous object calculated and stored.

We now know the total number of objects in a line and the gaps between adjacent objects. The largest gap of this set of gaps is known as the maximum gap, M_g . We then find the word gap denoted by W_g using (EQ 3.3). This word gap calculation allows correct separation of words.

$$W_g = M_g/2 \quad (\text{EQ 3.3})$$

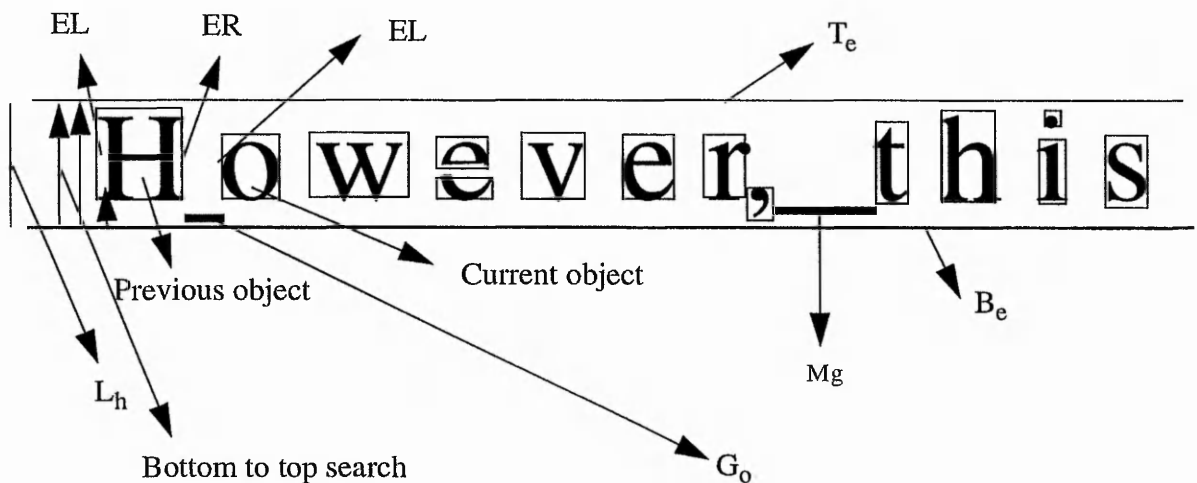


Figure 3.13: Method for finding word gap

A simple method for finding word gap in a given text line is described. The method is capable of finding word gap correctly if text lines are written in an orthodox way with proportional gap between the words and letters.

However, as discussed earlier, this assumed that lines of text will not have variable gaps between words, i.e. some words have a gap of 10 pixels, some have gap of 15 pixels and some have gap of 30 pixels. For such cases, the proposed method will not be applicable, and so layout analysis will be necessary before finding word boundaries.

3.6 Finding words

This section describes how each object of a particular word is found, and how the end point of the word is determined.

Starting from the bottom left corner of a text line, a search is made by moving upwards until an object is found, or the top of the line is encountered. If the latter occurs, we move a pixel to the right and continue from the bottom upwards.

When a black pixel is reached, an object has been found, and so the x and y coordinates of the black pixel are noted. An edge-following algorithm is then used to trace around the outer border of this object, and the extreme left, the extreme right, the extreme top and the extreme bottom positions denoted by EL, ER, ET and EB are noted. This object is considered to be the first object of the word, and its features are extracted using methods described in Chapter 4. At this point we do not know how many characters are in the word.

The search continues for the next object. If the search exceeds the previously calculated word gap, then it is assumed that the next object found is at the start of the next word. If an object is encountered before the word gap, then we have found the next object of the first word. The edge-following algorithm then finds the extreme positions of this object, and its features are extracted.

The procedure continues until each object of a word has been found (Figure 3.14), and passed to the feature extraction module.

When the word gap has been encountered, we know the total number of *objects* in the previous word. (Note that at this stage we do not know whether an object is a single character or two or more touching characters).

The search then continues for the next word and the next, until the end of the line is reached.

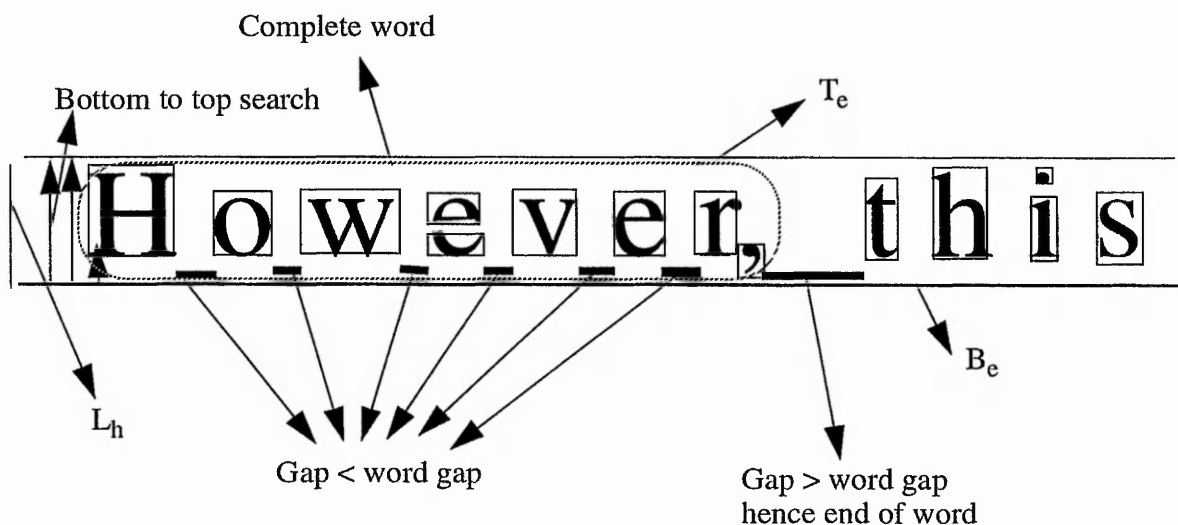


Figure 3.14: Word finding method

The word finding method described above is able to separate words using the pre-calculated word gap value. The method requires the text line to have a proportional gap between the words. In order to find word boundaries with disproportionate gaps, other methods need to be implemented. Development of such methods is not an aim of the present research.

In the case of poor quality documents containing a lot of touching or broken characters, overlapping lines, skewed text, hand written text, tables and figures, the

method cannot successfully find proper word boundaries (see Figure 3.15).

**KOLEJ BANDAR UTAMA, 50 Jalan SS 21/62, Damansara Utama, 47400 Petaling Jaya,
Selangor Darul Ehsan, Malaysia. Tel : 6 03-717 3200 Fax : 6 03-717 2733**

FACSIMILE MESSAGE

To : *Mr John Smith* From : MR LEE KER FOON, Principal
Department of Computing
Fax No : *TNT-U* Date : *15-2-95*
of pages incl. this page : *6* Ref No :

Subject :

as spoken I refer herewith

Figure 3.15: Sample poor quality facsimile message containing hand written text, tables and overlapping lines

Therefore, we have manually marked all machine printed words in all sample facsimile messages in order to obtain proper word boundaries (see Figure 3.16).

**KOLEJ BANDAR UTAMA, 50 Jalan SS 21/62, Damansara Utama, 47400 Petaling Jaya,
Selangor Darul Ehsan, Malaysia. Tel : 6 03-717 3200 Fax : 6 03-717 2733**

FACSIMILE MESSAGE

To : *Mr John Smith* From : **MR LEE KER FOON, Principal**
Department of Computing
Fax No : *TNT-U* Date : *15-2-95*
of pages incl. this page : *6* Ref No :

Subject :

as spoken I refer herewith

Figure 3.16: Manually marking of word boundaries

Objects in one box belong to one word. Recognition of the marked words is then attempted with the developed recognizer.

3.7 Summary

The method for finding text lines in a given image has been presented in this chapter. In order to locate the position and determine the size of an object, an edge-following algorithm has been developed. Different methods for finding objects have also been implemented and are included in this chapter. Finally, the simple methods for finding the gap between words and then the word boundaries themselves are presented.

Chapter 4

Feature extraction

4.1 Introduction

Feature extraction is an important stage in the recognition of characters particularly in the case of poor quality documents. The purpose of pattern recognition is to obtain the image and label it. Any object or pattern which can be recognized and classified possesses a number of relevant and significant features.

In order to recognize a character, different features are extracted, which will exhibit the distinctive characteristics of the character [Suen 82]. Ideally, the features should enable the recognizer to discriminate correctly one class of characters from another.

Feature extraction plays an important role in character recognition. There is no doubt that the central issue in character recognition lies in detection of proper features [Suen et al 68].

In the literature different commonly used feature extraction methods have been described, e.g. global features, distribution of points, geometric and topological features, linguistic descriptions, use of context and fuzzy sets.

There is no general technique for the design of feature extraction which utilizes the designer's a priori knowledge of the recognition problem. Geometrical and topological features are commonly used by human beings in the recognition of patterns because such features can easily be detected by the human eye. For the recognition of alphanumeric characters, human beings usually make decisions on the basis of topological features such as loops, cusps, cross bars, vertical lines, reversal of character strokes etc., and that is why these features are used for majority of the character recognition methods [Chatterji 86].

In the current research, attempts have been made to find different important features of characters in order to achieve the following goals:

- We are primarily dealing with poor quality documents. Therefore it is important that when an object has an extra part or a missing part, as is common in poor quality text, the features chosen are still able to identify the object. This has been achieved by choosing a sufficient quantity of features. A missing section may cause one or two features to be identified incorrectly, but the remaining features still identify the characters. Hence the chosen features are quite robust. The following example illustrates this point.



Figure 4.1: The objects with missing and unwanted extra parts

In Figure 4.1, the first letter has an unwanted extra part in its top half. This extra part leads to converting top side open feature of this object to hole. But this change has not effected the other features of this object, e.g. bottom side open, left side open, right side open and vertical bars etc. Similarly, in Figure 4.1, some part in the lower half of the second object is missing. Therefore, instead of bottom hole of that object, a bottom side open feature may now be extracted. Nevertheless other features of this object can be extracted correctly and are sufficient for its classification.

- We are dealing with multi-size characters i.e. sample text may be of any size from 6 point to 48 point size. Therefore attempts have been made to find features which are generally the same for different sizes of a particular object. For example one of the features we look for is an upper hole in the object. The letter “A” has an upper hole. Now if we consider the letter “A” of any size, this letter should have an upper hole as a feature. Similarly, another feature we extract from the objects is top side open. The letter “H” has a top side open feature. Now this feature is present in the letter “H” of any size. This can be observed in Table 4.1.

Letter	Point size			
	12	18	24	36
H	H	H	H	H
A	A	A	A	A

Table 4.1: Letters with different sizes having same features

- We are also dealing with multi font documents. Therefore, attempts have been made to find features which mostly remain unchanged across different fonts. For

example, consider top side open feature, the letter “U” has a top side open. This letter has top side open in different fonts e.g. Helvetica, Courier, Ariel and Times Roman etc. Similarly, the letter “a” has the lower hole in different fonts mentioned above. This can be seen in Table 4.2. This has the advantage of having one database covering several fonts rather than one database per font, and hence makes the system more efficient. However, to get a better recognition rate one database per font will be more successful - a database in Courier may give 100% recognition rate for a Courier document but only 90% for a Times Roman document, and vice versa.

Letter	Font		
	Courier	Helvetica	Times Roman
a	a	a	a
U	U	U	U

Table 4.2: Letters with different fonts having same features

In the present study, different important features of the objects are found. Each feature has different important information about the object, for example its position in the object (origin), length and width (extents) etc. These features are expected to be invariant with respect to font type and point size. The detailed method for finding each feature is explained below:

4.2 Finding bars

The method described below detects the presence of the following features in the input object image.

- (i) Vertical bars, e.g. H, L, etc.
- (ii) Horizontal bars, e.g. L, E, etc.
- (iii) Forward diagonal bars, e.g. Z, etc.
- (iv) Backward diagonal bars, e.g. X, etc.

In order to illustrate the method, an example input image is used, as shown in Figure 4.2.

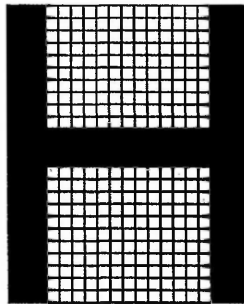


Figure 4.2: A sample input image

Assume that we wish to find the presence and 'extents' of vertical bars in the image. We therefore have to pass the function two direction parameters, based upon the direction indicator shown in Figure 4.3.

For vertical bars, we need to look vertically down the image from left to right. The first 'premier' direction given is therefore direction '6' or south. The second is '0' or 'east'.

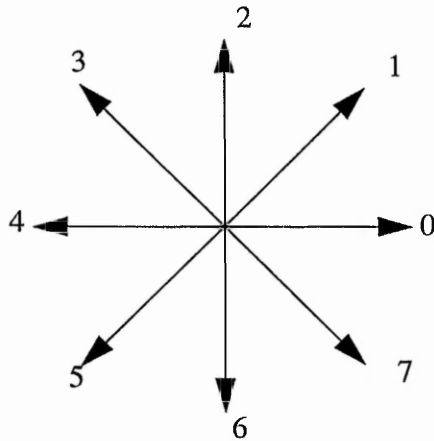


Figure 4.3: Direction indicator

Note that for horizontal bars, we would need to look horizontally, and then from top to bottom. In this case our premier direction is 'east', the second 'south'. Note also that for diagonal bars, the odd directions 1, 3, 5 and 7 are used.

Referring back to the example, our given directions, south and east, form a top-left corner (see Figure 4.3), which is the starting point for the search employed by this method.

The first step is to find a black pixel. The direction of search is detected by the two directions given. In the case of our example, a top to bottom (premier direction south), left to right (second direction east) search is made.

When a black pixel is found, its coordinates are noted. Attempts are now made to find a line of black pixels in the premier direction commencing from that point. This results in the line shaded light black in Figure 4.4 being found for our example image.

The black line will be terminated by either a white pixel or, as in the example case, by the end of the image in the premier direction. The coordinates of the last black pixel in the line are noted, and from this, and the coordinates of the starting

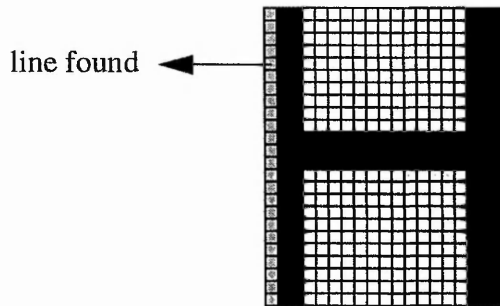


Figure 4.4: Finding a line of black pixels

point, the length of the line can be calculated. This length is then compared to the minimum bar length (B_{\min}) defined below.

Minimum bar length

A line of pixels must be of a certain length in order for the line to be classified as part of a vertical, horizontal or diagonal bar feature. Therefore we define a quantity B_{\min} against which lines are compared. Care must be taken in order to set this quantity such that only genuine bars are classified as such. For example, if B_{\min} is set too small, we may find bars in objects which clearly should have none. Similarly, if it is set to high, genuine bars will not be detected.

B_{\min} is defined as a percentage of the relevant image dimensions. For example, a line is considered to be part of a vertical bar if it is greater than X percent of the image height.

Experiments yield suitable values for X. Currently, X=40% gives appropriate results (Figure 4.6(c)).

Whilst this appears low, it has been found to detect bars suitably. Consider the images shown in Figure 4.5.

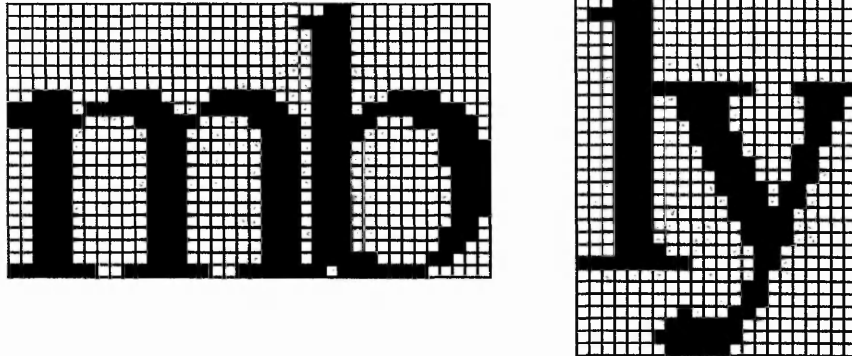


Figure 4.5: Sample images for setting minimum bar length

In this case two touching characters are present. A minimum bar length of 80%, say, would not capture the vertical bars of either 'm' or 'l' (Figure 4.6(a)). 30% would, but would find too many bars, for example, it would incorrectly find the vertical bars of 'y' (Figure 4.6(b)). A value between these will find only true vertical bars.

In the example (see Figure 4.4), we now have a line of black pixels of known length. If this length is less than the B_{\min} , then the line is not considered to be part of a bar. In this case, a search is made for the next black line, as follows.

The search for a black pixel recommences at either of two points. Consider a situation as shown in Figure 4.7.

Here, the line of black pixels found in the dot of the 'i' will not be of sufficient length to be classified as a vertical bar. However, there may yet be a part of a vertical bar in the same column that must be searched for. Therefore, if the distance between the terminal pixel of a black line and the edge of the image in the current premier direction is still greater than the B_{\min} , the search continues below that terminal point.

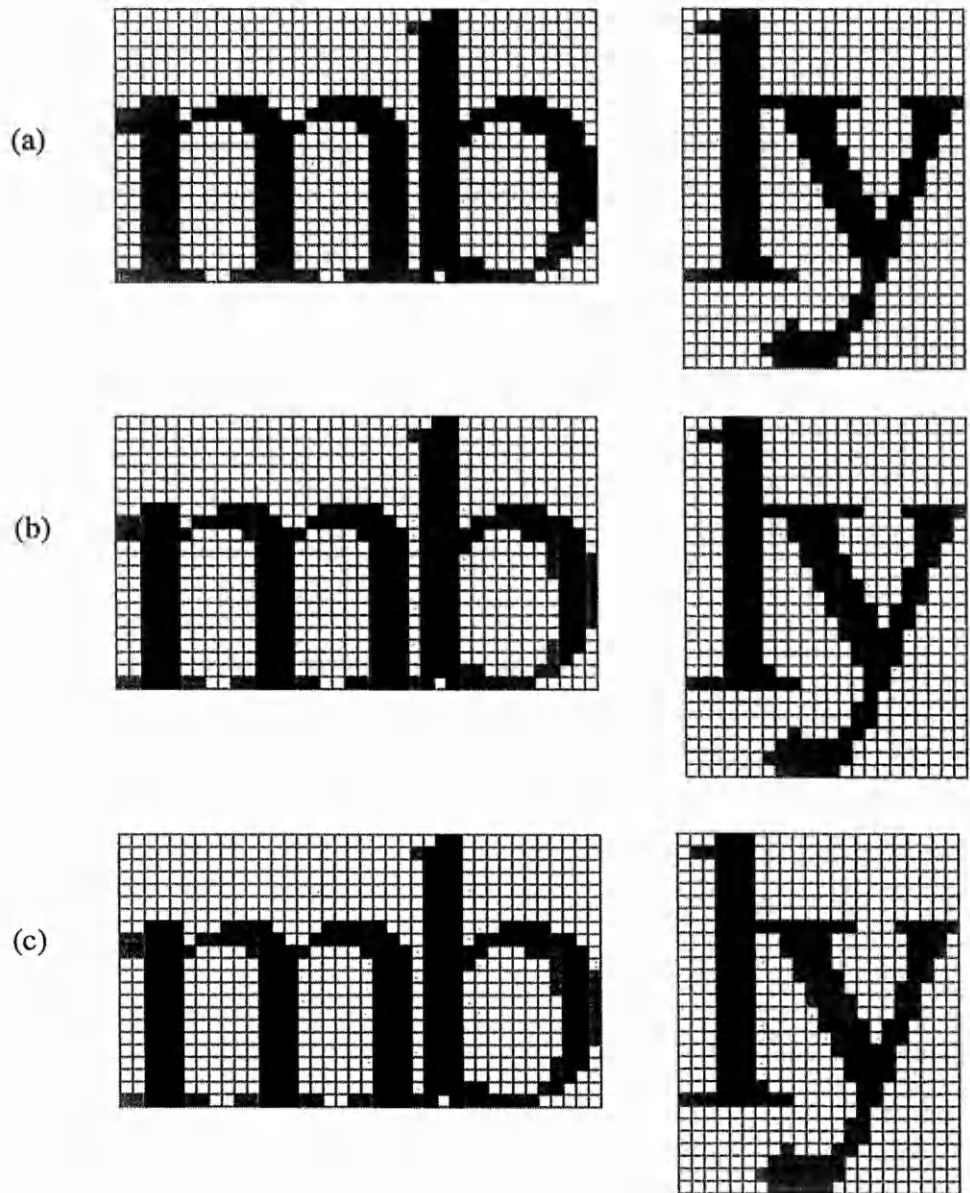


Figure 4.6: Different settings for minimum bar length (a) 80%, (b) 30% and (c) 40%

Otherwise, the search continues from the point at the top of the image, one pixel to the side of the previous column in the direction passed as the second parameter.

If the length of a line of black pixels is larger than the specified B_{\min} , the aim is to find the other lines of pixels which form part of the same bar (generally

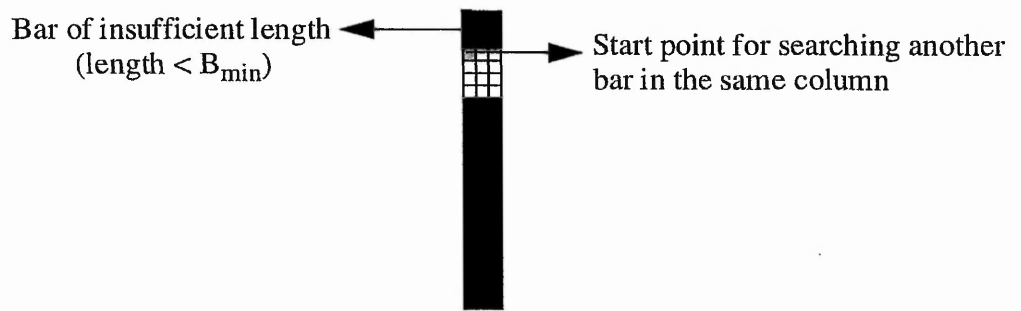


Figure 4.7: Continuing the search for a vertical bar

characters will be formed from 'lines' thicker than one pixel wide). Once this has been done, more data concerning the bar can be calculated, not just its presence.

The search for an adjacent black line begins at a point at the top of the image, one pixel to the side of the previous black line in the second direction given. If no black pixel is found, then we already have our bar. If a black pixel is found, a line originating from this point is sought, as described above. If the length of this line is smaller than the B_{\min} , then again we already have our bar. But if the line is longer than the B_{\min} , it is a part of the bar. In this case its length is compared to the previous line length. The larger is stored as the maximum bar length.

This process of finding adjacent lines continues until the bar is complete. We now have a situation as illustrated for the example in Figure 4.8.

The dark black line shaded in Figure 4.8 is the first line of 'black' pixels that does not form part of the vertical bar, mainly due to its length.

The above procedures search from the column to the right of this for further lines in an attempt to locate further bars. However, once a bar has been found, (as shown shaded light black in Figure 4.8) its origin and dimensions are found. Figure 4.8

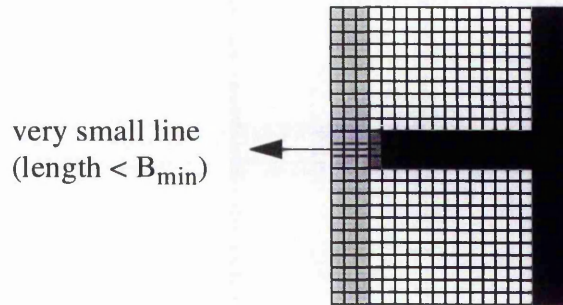


Figure 4.8: A vertical bar

shows this bar as a simple illustration. Figure 4.9 shows the bar alone in more detail.

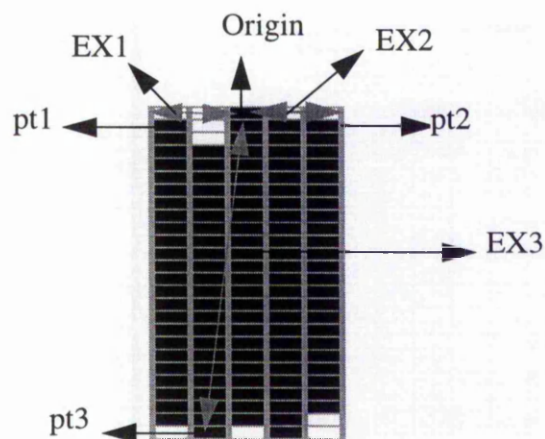


Figure 4.9: A vertical bar in close up

For each bar, we calculate 3 extents (EX1, EX2, EX3), which basically give extra data concerning the dimensions of the bar. Each results from a calculation using the 'Origin' of the bar.

To find the origin, we use the coordinates of the two points which are at the start of the first and last lines in the bar, marked as pt1 and pt2 in Figure 4.9.

i.e. $Width = pt2 - pt1$ (their x-coordinates in the case of vertical bars)

$MidWidth = Width/2$

$Origin(X,Y) = x$ coordinate of $pt1 + MidWidth$, y coordinate of uppermost pixel in bar

The Origin is shown in Figure 4.9.

Now consider one further point, the 'terminal point' of the bar. This is the bottom-most point of the bar, shown as $pt3$ in Figure 4.9. Now we can define the three extents and can be seen in Figure 4.9.

$EX1(X,Y) = pt1(X,Y) - Origin(X,Y)$

$EX2(X,Y) = pt2(X,Y) - Origin(X,Y)$

$EX3(X,Y) = pt3(X,Y) - Origin(X,Y)$

These extents are calculated for each bar of the current type (vertical, horizontal, etc.) in the image.

4.3 Finding corners open

One method is used for detecting the presence of either of the following features:

- (i) Top left corner open e.g. d
- (ii) Top right corner open e.g. L
- (iii) Bottom left corner open e.g. q
- (iv) Bottom right corner open e.g. p, r

The method is given two directions depending upon which of these features is being searched for. The directions are based upon the diagram shown in Figure 4.10.

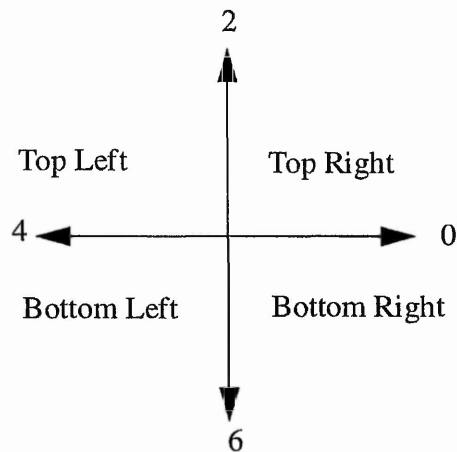


Figure 4.10: Direction diagram

If we wish to search for the presence of the top right open feature, we look at the two directions whose apex forms a top right corner, that is, 4 and 6 in this case.

The method is the same for each feature, but takes a different pair of parameters. In order to illustrate the method, let us consider the search for top right corner open. Figure 4.11 shows a typical input image.

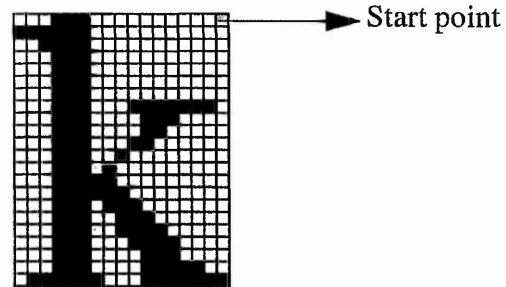


Figure 4.11: A sample input image with initial starting point

A starting point is chosen depending upon the two directions passed as described below. It is the pixel in the corner formed by the meeting of the two directions.

For top right corner open, the directions passed are, 4 and 6, imply a start point in the top right corner, shown in Figure 4.11 as 'Start point'. If this point is a black foreground pixel, it is assumed that the feature is not present and the method ends. If it is white however, the method continues.

The first direction passed is used first (it does not matter which is used first, hence the arbitrary order allowed when passing directions initially).

The next pixel in the first direction is considered. If it is also white, then a rectangle is formed, as illustrated in Figure 4.12. If it is black, then no action can be taken in that direction.

The second direction is then considered. The next pixel in this direction, starting from the starting point, is checked. If it is white, then the method attempts to move

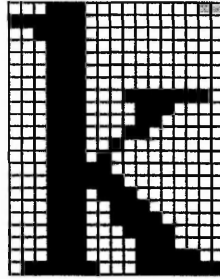


Figure 4.12: Forming a rectangle

from that point, in the first direction, until the boundary of the rectangle already formed.

If such a line can be created (the presence of a black pixel in this line will prevent such an outcome), then the line becomes part of this rectangle, as shown in Figure 4.13.

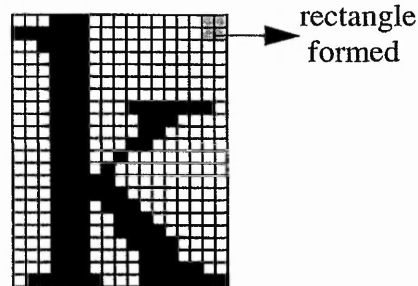


Figure 4.13: Adding a line and expanding the rectangle

The aim is to increase the rectangle size until no more lines can be added in either of the two directions. Once no further lines are possible, the size of the rectangle, and its shape indicate the extent to which the feature is present.

So we continue to alternate between directions, adding lines until no more are possible. In the example, we look in the first direction, left again. The next pixel to

check is to the left of the starting point, one pixel left of the last pixel in the rectangle on that row, shown as a 'Start point' pixel in Figure 4.14.

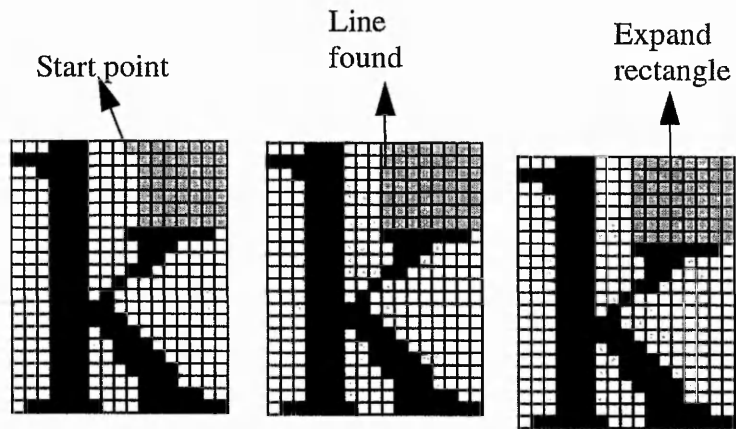


Figure 4.14: Starting point of the next line, finding line and expanding rectangle

Again we look in the second direction, from this point, until the end of the existing rectangle in the second direction. If no black pixels are encountered, we add the line to our rectangle, and continue to try to add lines, next by looking from the next point down in the second direction ('Start point' pixel in Figure 4.15). Figure 4.15 illustrates this:

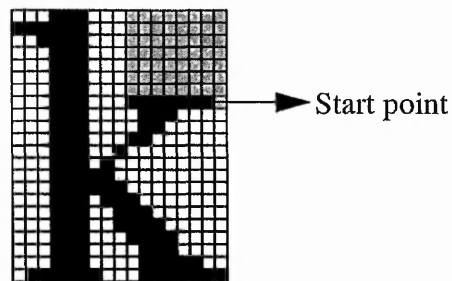


Figure 4.15: The latest situation

Now we look in the first direction (left) from the 'Start point' pixel. In this example we have a situation where the line cannot be completed. The next pixel

left is black (see Figure 4.16). Therefore nothing is added to the rectangle, the first direction is marked as having been completed, and second direction used for looking for further lines.

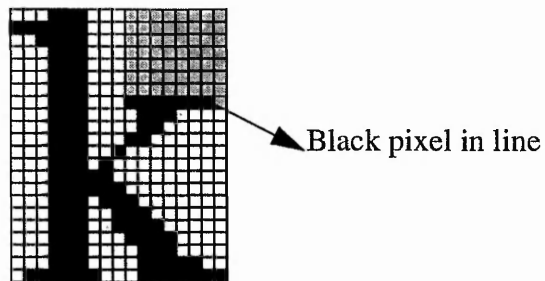


Figure 4.16: Direction exhausted as black pixel is found after start point

Note that in this situation, alternating stops, and the remaining uncompleted direction searched until that too is exhausted. The method stops at that point.

Figure 4.17 shows the next starting point ('Start point' pixel).

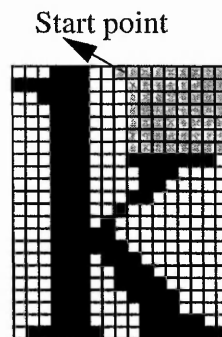


Figure 4.17: Next pixel

A line in the second direction is then found, and the line added to the rectangle, as shown in Figure 4.18.

As the second direction has already been exhausted, the first direction is used again until that direction is also exhausted. The next pixel to look at is one to the

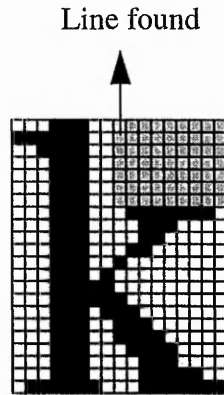


Figure 4.18: Another line added to the rectangle

left of the rectangle, in Figure 4.18. If this pixel is black, then no line can be attempted. If the line vertically downwards from that point is not of the same length as the height of the rectangle, no line is found. Otherwise the line is added to the rectangle. This will give us the following situation (see Figure 4.19).

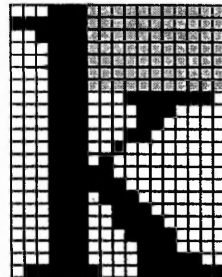


Figure 4.19: Maximum successful growth of the rectangle

The next pixel to look at is one to the left of the rectangle, in Figure 4.19 and is shown in Figure 4.20. As this is black, then no line can be attempted. This direction is marked as being exhausted.

Now, as both directions have been exhausted, the method ends. The rectangle such as that illustrated in Figure 4.19 is the result.

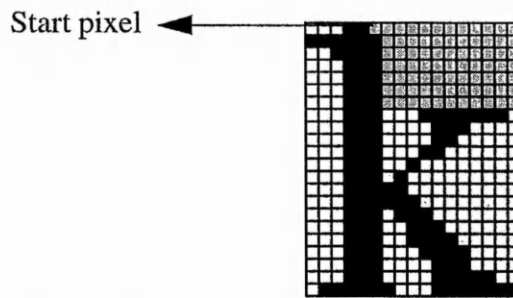


Figure 4.20: Direction exhausted as start pixel is black

Once a feature has been found, we store two things, the origin and one extent (EX1). The origin is the first starting point as described previously, i.e.

Origin(X,Y)=First start point

The EX1 is found by calculating, the length of the diagonal running from the origin to the opposite corner of the rectangle.

$EX1(X,Y) = \text{Origin}(X,Y) - \text{diagonally opposite}(X,Y)$

Figure 4.21 illustrates this.

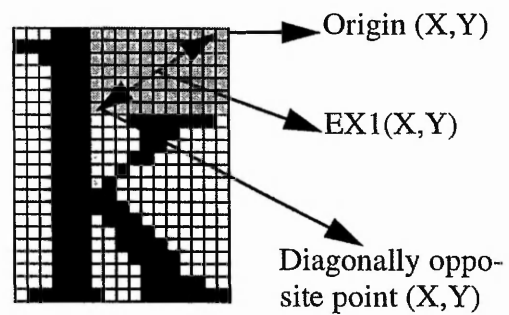


Figure 4.21: Storing feature details

4.4 Finding centre of gravity

A method has been developed to extract the following:

- (i) Centre of gravity using whole object (Centre of gravity)
- (ii) Centre of gravity using boundary of the object (Edge centre of gravity)

Each of these measures is stored as an 'extent'. The method is described as follows:

4.4.1 Centre of gravity using whole object

The aim of this feature is to determine where most of the ink of the object is located. As an example, consider the object images shown in Figure 4.22.

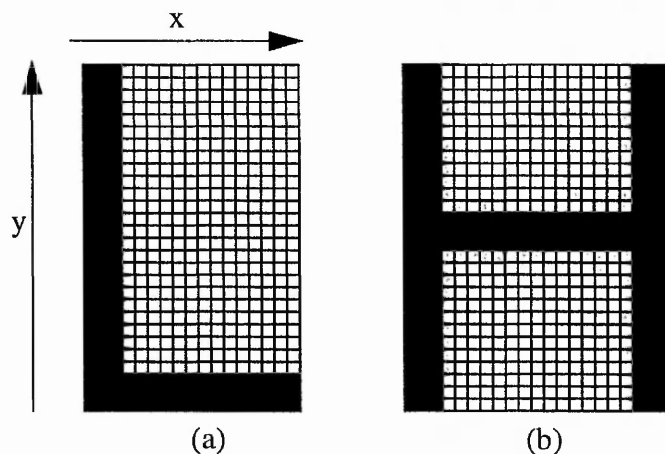


Figure 4.22: Sample input object images

In Figure 4.22(a), the centre of gravity would intuitively be near the bottom left corner of the image, as most of the ink is on the left or bottom edge of the image. In Figure 4.22(b), the character is symmetrical both horizontally and vertically, and well balanced, the centre of gravity is therefore expected to be near the centre of the image.

The centre of gravity $C(Xg, Yg)$ of an object is obtained from (EQ 4.1) and (EQ 4.2).

$$Xg = \frac{\sum_{x, y \in image} C(x, y) \times x}{\sum_{x, y \in image} C(x, y)} \quad (\text{EQ 4.1})$$

$$Yg = \frac{\sum_{x, y \in image} C(x, y) \times y}{\sum_{x, y \in image} C(x, y)} \quad (\text{EQ 4.2})$$

where $C(x,y)$ is the colour of the pixel at (x,y) , either 1 if black or 0 if white.

The Centre of gravity of our example images is calculated as $(609/126, 1302/126)$ and $(2343/213, 3111/213)$, as shown in Figure 4.23.

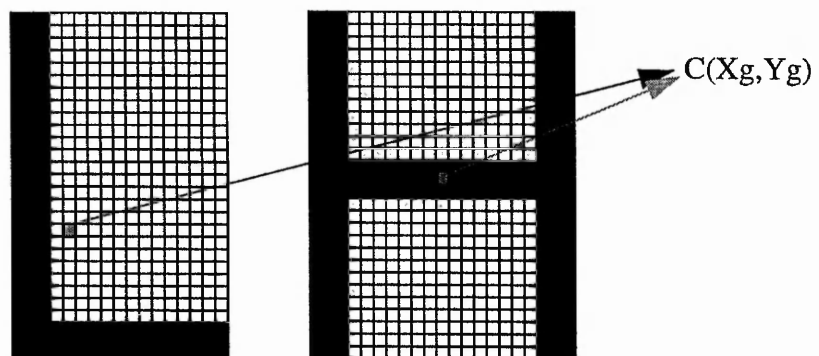


Figure 4.23: Centre of gravity using whole object

4.4.2 Centre of gravity using boundary of an object

The aim of this feature is also to determine where most of the ink of the object is located, but using the boundary of the object. This allows comparison with

previous whole object centre of gravity method. To obtain the object's boundary, the black pixels that are part of the colour transitions in either horizontal or vertical directions are marked, as shown in Figure 4.24.

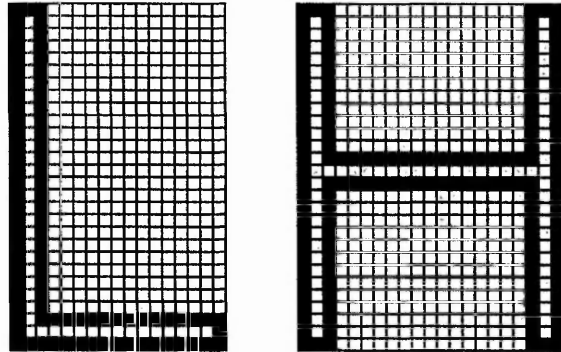


Figure 4.24: Sample images obtained from the boundary of the real objects

The edge centre of gravity ($EC(X_g, Y_g)$), is then obtained using the above equations ((EQ 4.1) and (EQ 4.2)). In this case $EC(X_g, Y_g)$ is the same as the $C(X_g, Y_g)$, but could be different for touching characters.

Once $C(X_g, Y_g)$ and $EG(X_g, Y_g)$ have been found, we now calculate the origin and extents of the feature and store them. The origin is simply the centre pixel of the input object image (see Figure 4.25).

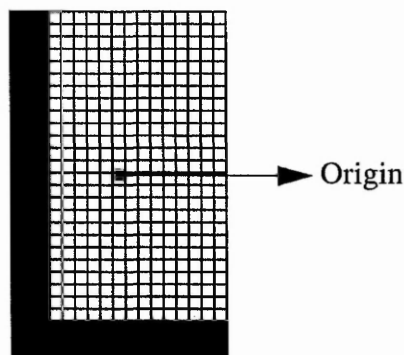


Figure 4.25: Origin

Two extents 'EX1', 'EX2'. are to be calculated. These are the distances between origin and centre of gravity and edge centre of gravity respectively. i.e.

$$EX1(X,Y) = \text{Origin } (X,Y) - C(X_g, Y_g)$$

$$EX2(X,Y) = \text{Origin } (X,Y) - EC(X_g, Y_g)$$

4.5 Finding open sides using multi-colouring method

A method has been developed for finding the presence of any of the following features:

- (i) Left side open, e.g. S, Z etc.
- (ii) Right side open, e.g. E, C etc.
- (iii) Top side open, e.g. U, V etc.
- (iv) Bottom side open e.g. M, N etc.
- (v) Holes, e.g. O, B, g etc.

The input for this method is a binary image consisting of white background and black foreground as shown in Figure 4.26(a).

This binary image is then converted into a colour image. This step also includes adding a border consisting of white pixels, 2 pixels deep to the image. This is necessary for the “Multi-colouring” part of the process explained later. Such a colour image with a border is illustrated in Figure 4.26(b).

A number of steps are now taken using the colour image. These accept a parameter according to which of the listed features is being searched for. Before describing these steps, a number of directions are defined as shown in Figure 4.3.

Now, one direction is used for each feature. To search for right side open, direction ‘0’ is used, while to search for bottom side open ‘6’ is used. If no direction is chosen, the method will look for holes.

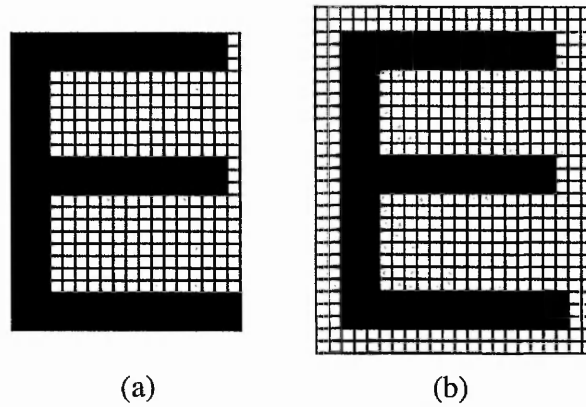


Figure 4.26: (a) A binary input image, (b) Colour image including border

4.5.1 Finding the seed point

The seed point is a pixel located on the white pixel border opposite to the chosen direction. It is set to a different colour to the background or foreground colour, for example green. Figure 4.27 shows the seed point when right side open is being searched for.

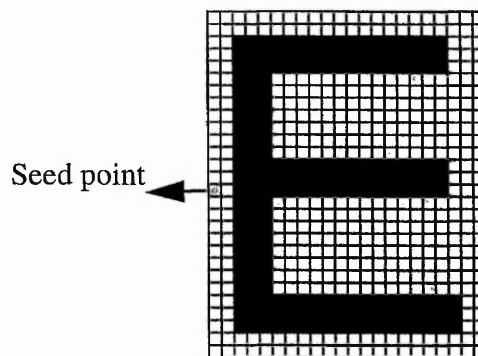


Figure 4.27: Seed point

Note that if no direction is passed (i.e method looks for a hole), any pixel on the border can be used as a seed point and set to colour green.

4.5.2 Multi-colouring to green

This is a vital step leaving the image suitable for further processing. The function takes an image such as that shown in Figure 4.27 and produces a coloured image as shown in Figure 4.28. This is done by using the directions shown in Figure 4.3.

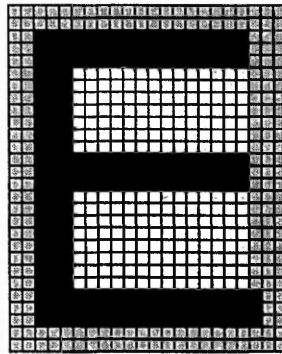


Figure 4.28: Coloured image

Firstly, the direction opposite to the passed direction is excluded. Colouring cannot occur in this direction. NB: for hole searching, no direction needs to be excluded. For right side open, direction '0' is passed, so direction '4' is excluded. This leaves three premier directions, in this example 0, 2 and 6.

Now a search direction is calculated for each giving direction pairs. The formula is simply

$$\text{second direction (sndDir)} = \text{premier direction} + 2.$$

Now in the example, we have (0,2), (2,4) and (6,0) pairs.

Each pair forms a corner when viewed with respect to the direction diagram (Figure 4.3), as shown in Figure 4.29.

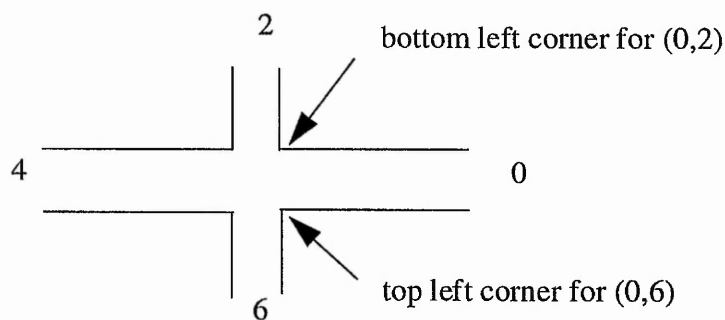


Figure 4.29: Corners

The image corner represented by the first direction pair is the starting point.

Colouring takes place in the direction of the premier direction. If there is nothing is to be done, a step of one pixel is made in the second direction. Colouring is a process of turning a white pixel to green provided that white pixel is next to a green pixel. That is why the seed point (green pixel) is necessary.

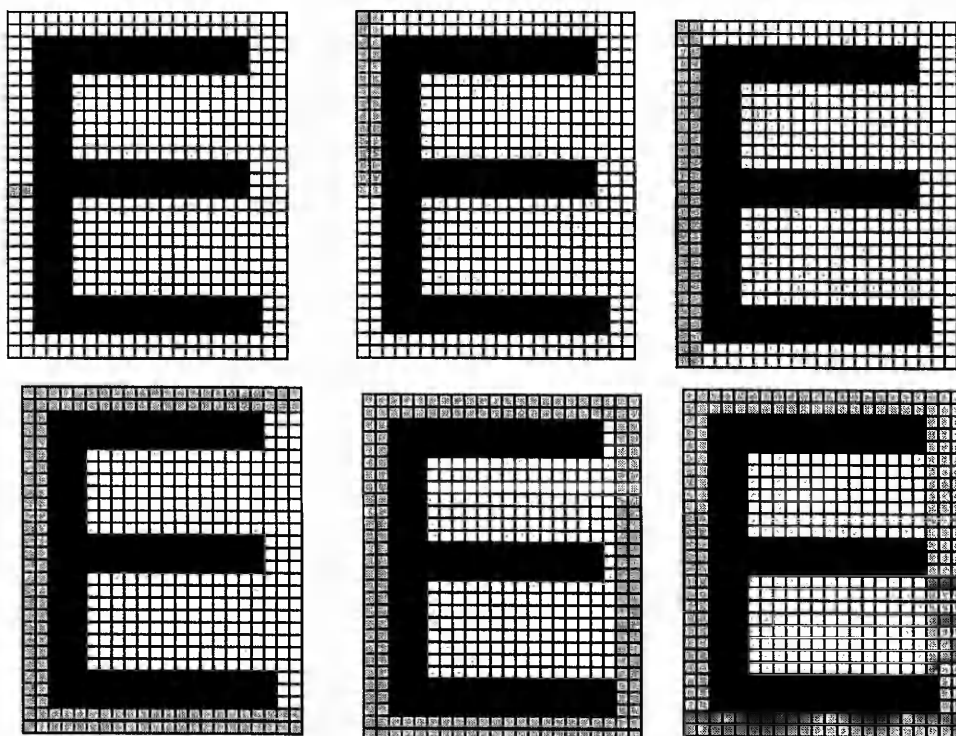


Figure 4.30: The colouring process

Each direction pair is used until no further colouring can occur in that direction. Once the last pair has been used, the first is used again. This process continues until the image is completely coloured, as illustrated in Figure 4.30. Some example coloured images are shown in Figure 4.31.

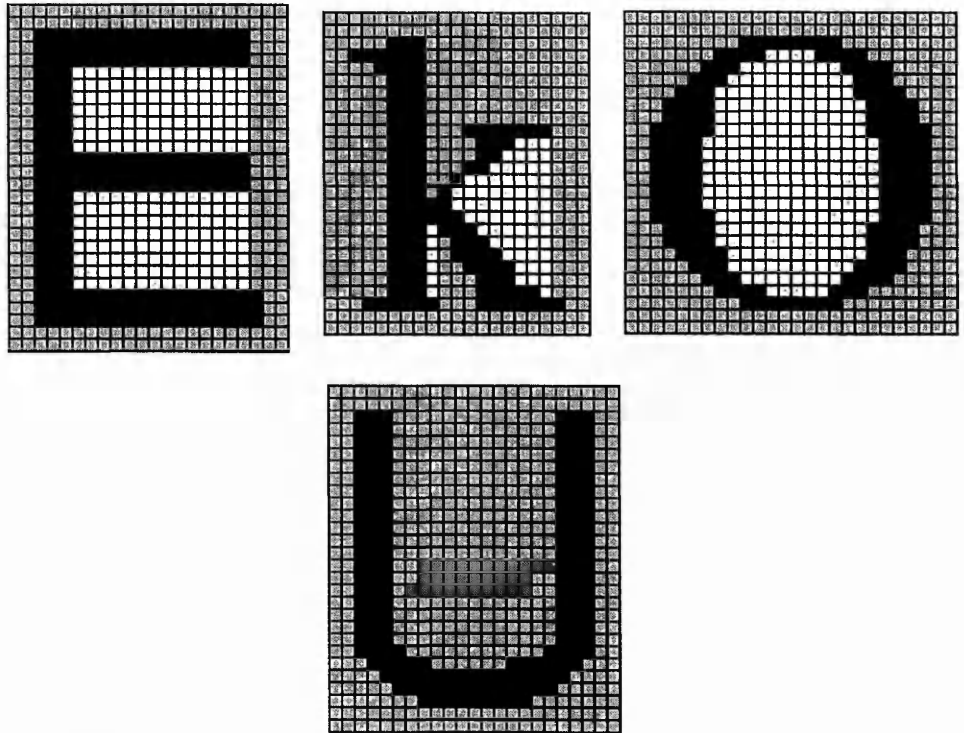


Figure 4.31: The effect of colouring after using all necessary pair directions

4.5.3 Finding a green/white pixel pair

This part of the method finds corners based on direction pairs. It takes the coloured image as shown in Figure 4.31, and creates a new seed point, this time colouring it differently, say, red.

In order to find the green/white pixel pair, a corner of the image is needed from which to start looking. Again this corner is selected by using the direction passed initially. Two further directions are calculated using the following formulas:

Dir 1: Initial direction + 4;

Dir 2: Initial direction + 2;

For right side open

Dir 1 = $0+4 = 4$;

Dir 2 = $0+2 = 2$;

NB: for no parameters, any corner can be used as the starting point.

Again, these directions reveal a corner based on the direction diagram (Figure 4.3). For direction pair (4,2), this gives bottom right corner as shown in Figure 4.32.

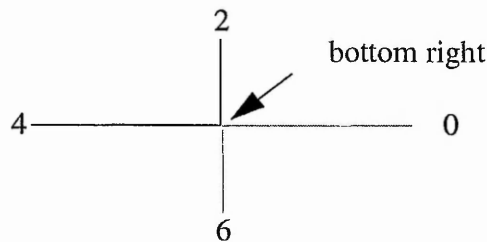


Figure 4.32: Corner obtained from the direction pair

Each pixel in the image is checked until it satisfies the condition for a pixel pair. In the example, pixels are checked from right to left (premier direction: 4) and from bottom to top (second direction: 2) until a green pixel is found adjacent to a

white pixel in that direction. The white pixel is coloured red, and designated as the seed point, for example, see Figure 4.33.

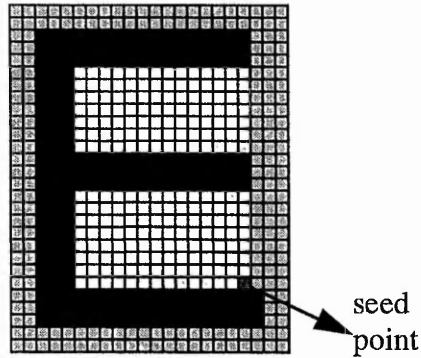


Figure 4.33: New inside seed point

4.5.4 Multi-colouring to red

Multi-colouring to red will give us our open area. Starting at the new red seed point, pixels are coloured to red using the above method for colouring. This stops when no more pixels can be coloured in any of the allowed directions. For the example, this leaves the image looking as shown in Figure 4.34.

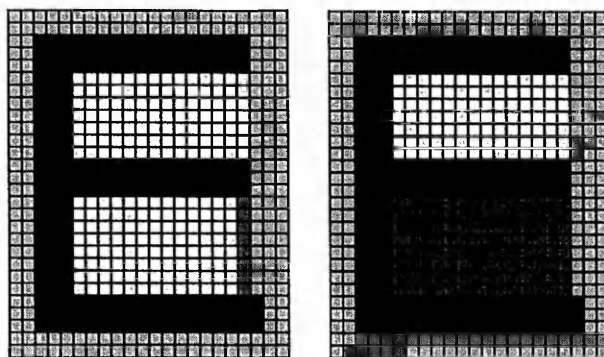


Figure 4.34: The effect of colouring by using new seed point

Now, it is known whether the feature searched for exists or not. In the example, there is a right side open as this colouring occurred. Moreover, this is not the complete method - more information can be gained.

4.5.5 Multi-colouring from red to green

If a red area of image exists, it can be considered as having been processed. By colouring red to green, further green/white pixel pairs may be looked for.

Therefore, this step takes all red pixels and sets them to green colour. For the example, this leaves a situation as shown in Figure 4.35.

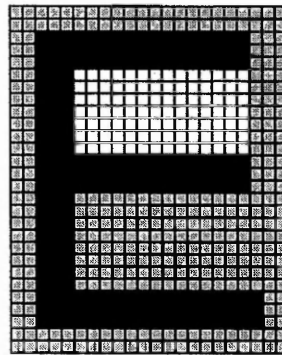


Figure 4.35: Image after setting all red pixels to green

4.5.6 Further steps

Steps (Section 4.5.3 to Section 4.5.5) are now repeated, stopping when there are no further white/green pixel pairs, or, in other words, all white pixels have been coloured. Figure 4.36 shows the image after a further colouring to red.

Once the feature has been found, we store the origin and three extents (EX1, EX2, EX3). The origin is defined as the first red/green transition (P0), as shown Figure 4.37, i.e. $\text{Origin}(X,Y) = P0(X,Y)$

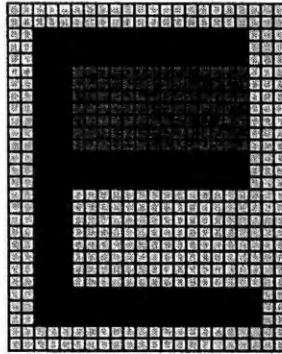


Figure 4.36: Another open side found by colouring white to red

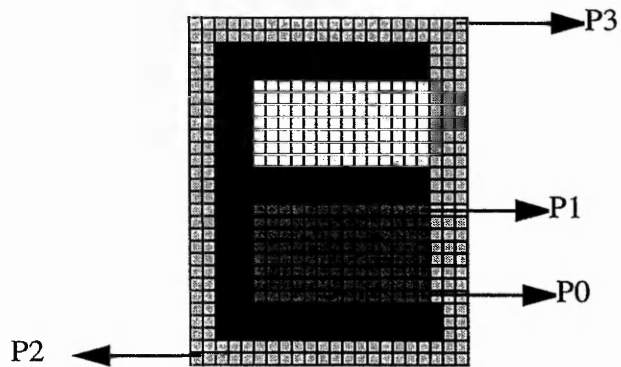


Figure 4.37: Finding the origin and extents

Three extents calculated using the following equations.

$$EX1(X,Y) = P1(X,Y) - P0(X,Y)$$

$$EX2(X,Y) = P2(X,Y) - P0(X,Y)$$

$$EX3(X,Y) = P3(X,Y) - P0(X,Y)$$

$P1(X,Y)$ is the last red/green transition. $P2(X,Y)$ is the 'object origin' as opposed to the origin of the side open feature, and $P3(X,Y)$ is the north east point of the image.

4.6 Finding zones

A method has been developed for finding the zone of each object within a word.

It is assumed that characters could be of any one of the following zone types:

Middle zone (m)

The middle zone type contains the letters or objects with no ascenders or descenders, e.g. a, c, e, i, m, n etc.

Upper zone (u)

This is the zone type, in which characters with ascenders lie, e.g. b, d, h, k, l etc.

Lower zone (l)

This zone type contains the letters, which have descenders, e.g. g, j, p, q, etc.

Full zone (f)

This is a special zone type. This zone type is contained by the objects, which have ascenders as well as descenders. From the definition of this zone type, it is clear that no single letter has this zone, as no single letter in printed text has both ascenders and descenders. However this can be found in handwritten text, e.g. 'f' etc.

In printed text objects formed by merging upper and lower zone letters fall into this zone type. For example, when letter 'l' and 'y' are joined together, then the zone type of joined letters 'ly' is full.

At line level we do not always have information of what is the zone of text in a line or zone of different words in a line or zone of different characters in a word. This is because of the following main reasons:

- Sometimes it happens that a line has a mixed text of different font and sizes. This may be explained in Figure 4.38.



John stood **first** in the examination.

Figure 4.38: Text with different fonts and sizes

In the above example, it is very difficult to determine the zoning information of the text or of different words in a line at the line level.

- It may also happen that the text line is bigger in beginning of the line and smaller later on or vice versa. This can well be seen in Figure 4.39.



I am pleased to learn from you that your department is able to admit more than 20 students

Figure 4.39: Text line bigger in beginning and smaller at the end

Again it is very difficult to find out the zoning information of each character of the word at line level.

Due to the above problems, it was decided to find all characters of a word from the line first and then the zoning information of each character will be calculated. It is assumed that each word is usually written using the same font and size. This is not always the case - it occasionally happens that the first letter of the first word in the start of a paragraph is bigger than the following letters of the word and also the letters of the preceding words. This is shown in Figure 4.40.

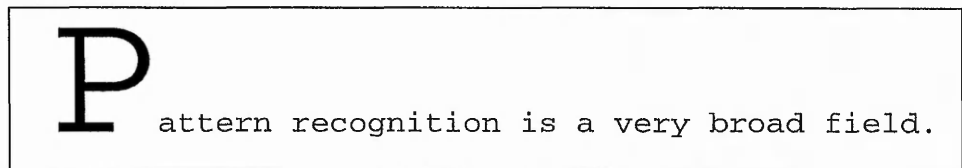


Figure 4.40: Line of text with first letter much bigger than others

The input is a word image, an example of which is shown in Figure 4.41, and will be used to illustrate the method.

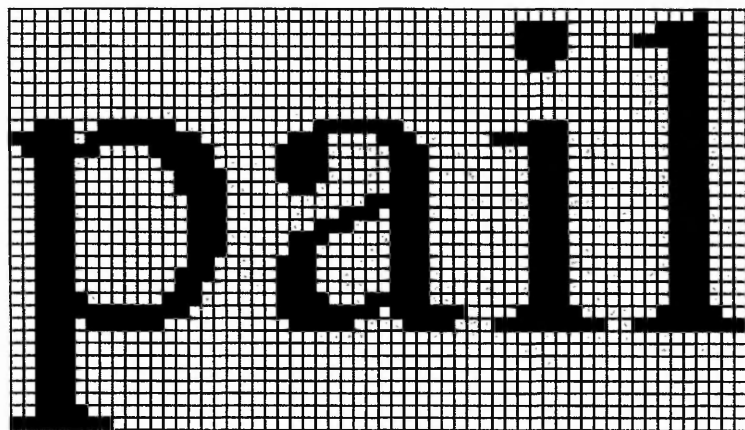


Figure 4.41: A sample word image

The first step is to create a horizontal projection histogram. This can either be determined from the number of black pixels in each scan line of the word or from the number of transitions in the colour along the line as shown in Figure 4.42(a).

The histogram for the example word will have an appearance similar to that shown in Figure 4.42(b).

Before the histogram is used to make the appropriate calculations, a smoothing process is applied as described below. This reduces the peaks and evens out the gaps, making subsequent calculations more likely to be accurate (Figure 4.42(c)).

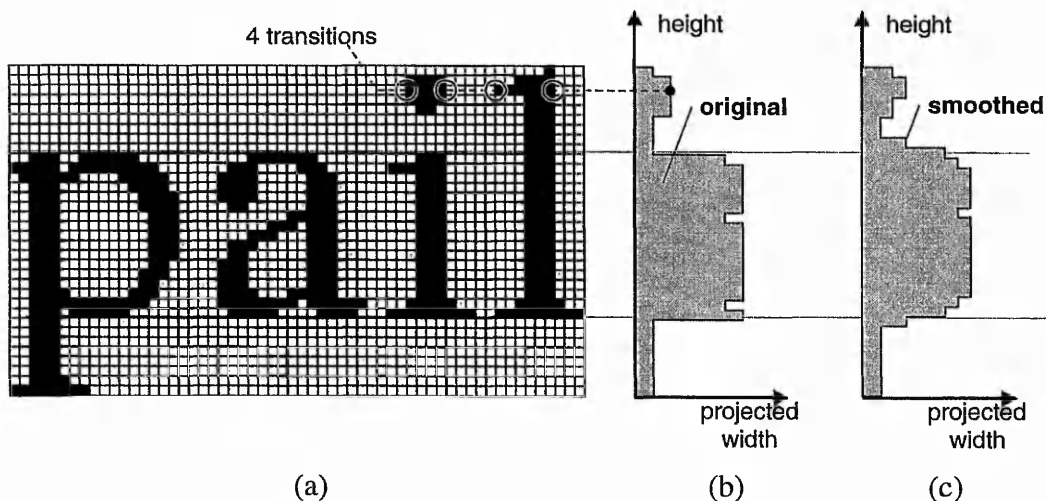


Figure 4.42: A horizontal projection histogram derived from colour transitions for the example word (a) Example word image (b) Original histogram (c) Smoothed histogram

Each element (h_j) in the histogram is smoothed into h'_i with respect to its ' d ' neighbours using (EQ 4.3).

$$h'_i = \frac{\sum_{j=i-d}^{i+d} \frac{d-|j-i|}{d} \times h_j}{\sum_{j=i-d}^{i+d} \frac{d-|j-i|}{d}} ; \begin{matrix} i=0,1,\dots,H \\ h_j=h_0 \text{ if } j<0 \\ h_j=h_H \text{ if } j>H \end{matrix} \quad (\text{EQ 4.3})$$

Where H is the height of the bounding rectangle of the word. The value for d has been experimentally determined to be 3.

To detect the shoulder of the middle zone plateau in the histogram a cut off value is calculated. This value is used to find the location of the middle upper and middle lower lines of the word.

The cut off value is derived from the maximum value of the histogram, that is, the row of the image with the most transitions after the smoothing process. Starting from the maximum cell in the histogram, the first minimum below the cutoff value is obtained in both upwards and downwards directions. These cells are regarded as being outside the middle zone plateau. Searching the first cell that exceeds the cutoff in inwards direction yields the desired zone line positions.

If this maximum value is lower than an experimentally derived threshold value, the histogram is considered as 'flat'. A typical threshold used is 4 transitions. In this case, the cut off value is defined as 0.3 the maximum.

If the maximum value indicates that the histogram is not 'flat', the cutoff value is derived from the average histogram value modified towards the maximum value using (EQ 4.4). This reduces the impact of potentially, exceptionally large maxima. The modifier has experimentally determined to be 0.09 [Powalka 95].

$$Cutoff = Average + (Maximum - Average) \times Modifier \quad (EQ 4.4)$$

The resulting middle upper and middle lower lines are depicted in Figure 4.43.

Once the middle lines have been found, the top and bottom lines are determined. If the ascenders and descenders are present in the word, then the outer limits of the histogram represent the upper and lower lines, as shown in Figure 4.43.

If, for example no ascenders are present (Figure 4.44), the upper shoulder coincides with the upper limit of the histogram. The upper line is then derived from the size and position of the middle zone, using an experimentally obtained average upper zone width. The upper zone is on average 50% of the middle zone, the lower zone is 47%. In Helvetica font however, these values have been observed

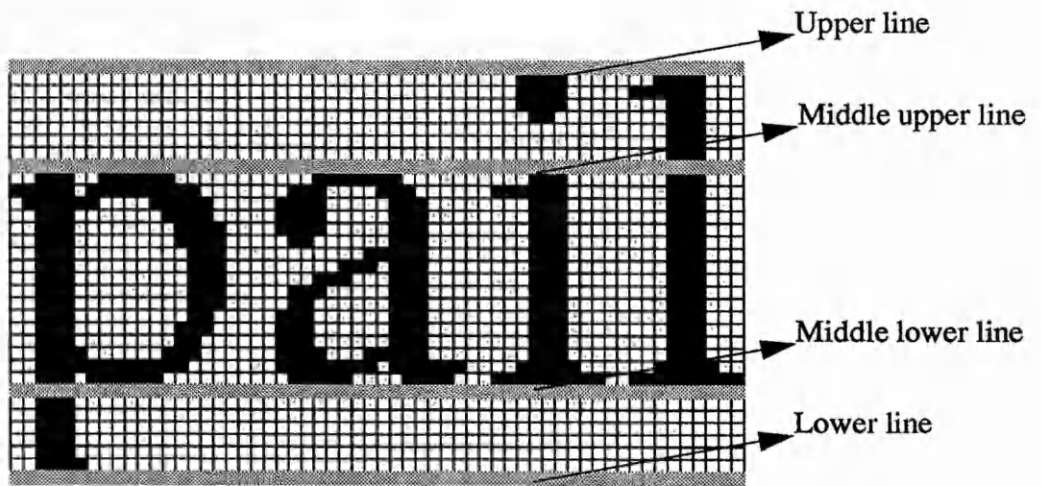


Figure 4.43: Calculation of upper, middle upper, middle lower and lower lines to be significantly smaller (30% and 30%). The upper and lower lines are therefore derived from the (EQ 4.5) and (EQ 4.6) respectively, and are shown in Figure 4.44.

$$Y_{upper} = \max(\text{histogram top}, Y_{middleUpper} + 50\% (Y_{middleUpper} - Y_{middleLower})) \text{ (EQ 4.5)}$$

$$Y_{lower} = \min(\text{histogram bottom}, Y_{middleLower} - 47\% (Y_{middleUpper} - Y_{middleLower})) \text{ (EQ 4.6)}$$

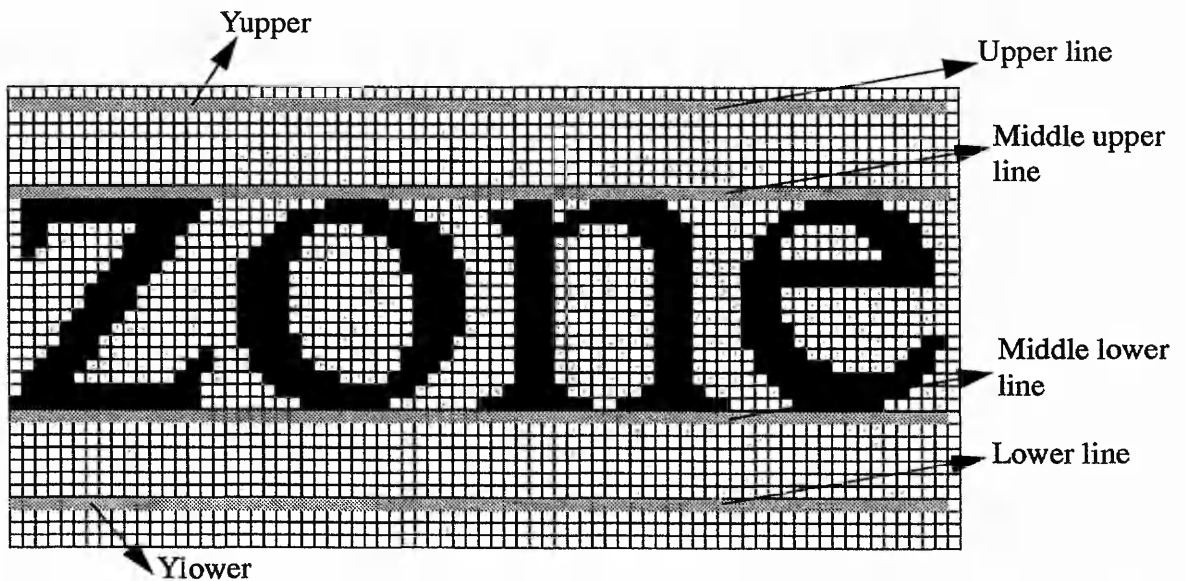


Figure 4.44: Estimation of upper and lower lines in a word with no ascenders and descenders

4.7 Finding dots of 'i' and 'j'

A method has been developed to find dot-shaped objects in the given image using a number of constraints [Hennig et al 97]. The found dots were then used to detect whether a text document has been scanned upside down or not. Here the constraints are used to find the dots of i and j. The method is described below.

The input for this method is a binary image containing one or more objects. As an example, consider the images shown in Figure 4.45.

```
~!"£$%^&*()_+~`{|[]@':;<>? , . / | \ 0123456789  
abcdefghijklmnopqrstuvwxyz  
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

Figure 4.45: Sample image for finding dots

The first step is to find different regions, i.e. objects in the image using the colouring connected components method as described in Section 3.4.2. A set of different constraints is then used to identify the objects which are most likely to be dots namely: aspect ratio, region area to bounding rectangle ratio, circumference to area ratio, concavity to convex hull ratio and no inside white pixel condition (see Figure 4.50 and Table 4.3). These constraints are explained below:

1. Aspect ratio

A dot is ideally a filled circle, which has an aspect ratio of the circle is 1. Therefore the aspect ratio of the bounding rectangle of the dots is also expected to be close to 1. The effect of this constraint on the example image is shown in Figure 4.46. This excluded very tall objects (e.g. 'l' and 'j' etc.) as well as very flat objects (e.g. '_' etc.).

```

! " $ % ^ & * ( ) _ + = { } | [ ] \ ' : ; < > ? , . / \ 0 1 2 3 4 5 6 7 8 9
abcdefghijklmnopqrstuvwxyz
ABCDEFGHIJKLMNOPQRSTUVWXYZ

```

Figure 4.46: The effect of aspect ratio constraint

2. Object area to bounding box area ratio

The ratio of the area of a circle and the area of its bounding rectangle is approximated by the number of black pixels in the object and the area of its bounding rectangle. The found value should be close to 0.79 as the ideal value for the circle is $\frac{\pi}{4}$. Figure 4.47 shows the effect of this constraint on the sample image excluding less dense objects (e.g. 'O', 'H' and 'L' etc.).

```

! " $ % & * ( ) _ + = { } | [ ] \ ' : ; < > ? , . / \ 0 1 2 3 4 5 6 7 8 9
abcdefghijklmnopqrstuvwxyz
ABCDEFGHIJKLMNOPQRSTUVWXYZ

```

Figure 4.47: The effect of area to bounding box ratio constraint

3. Circumference to area ratio (Compactness)

For an object to be a dot, the ratio between the square root of the objects's area and the length of the object's outer edge boundary (i.e. circumference of an ideal circle) should be close to 0.28, as the ideal value for a circle is $\frac{1}{2\sqrt{\pi}}$. Edge-following algorithm is used to obtain the objects's outer boundary as shown in Figure 4.50(a). The object's area is then expressed as the number of black pixels in it. The effect of this constraint is shown in Figure 4.48 excluding non-compact objects.

```

~!"$%&*()_+~`{|}[]^'::;<>? ,./|\0123456789
abcdefghijklmnopqrstuvwxyz
ABCDEFGHIJKLMNOPQRSTUVWXYZ

```

Figure 4.48: The effect of circumference to area ratio constraint

4. Concavity area

As an ideal dot is convex, the total area of the concavities found in the object should be zero. The total area of the concavities is expressed as the total of all areas that have to be added to the polygonal object in order to form a convex hull as shown in Figure 4.50(b). The effect of this constraint can be seen in Figure 4.49. This constraint excludes objects with larger concavities (e.g. ‘C’, and ‘G’ etc.).

```

~!"$%&*()_+~`{|}[]^'::;<>? ,./|\0123456789
abcdefghijklmnopqrstuvwxyz
ABCDEFGHIJKLMNOPQRSTUVWXYZ

```

Figure 4.49: The effect of concavity area constraint

5. No inside white pixel

The constraints defined above may accept objects as dots which contain a white area inside, such as simple circles (e.g. ‘0 or o’ etc.). Similarly the letters ‘e’ or ‘a’ in poor quality printing when touch from the sides making a convex shape. Ideally the entire objects would have to be verified, but at present only few pixels are verified to be black. The centre of the bounding rectangle should be the centre of the dot and therefore must be a black pixel. The eight points halfway between the centre and the edges of the bounding rectangle are also tested to check if they are black (see Figure 4.50(c)).

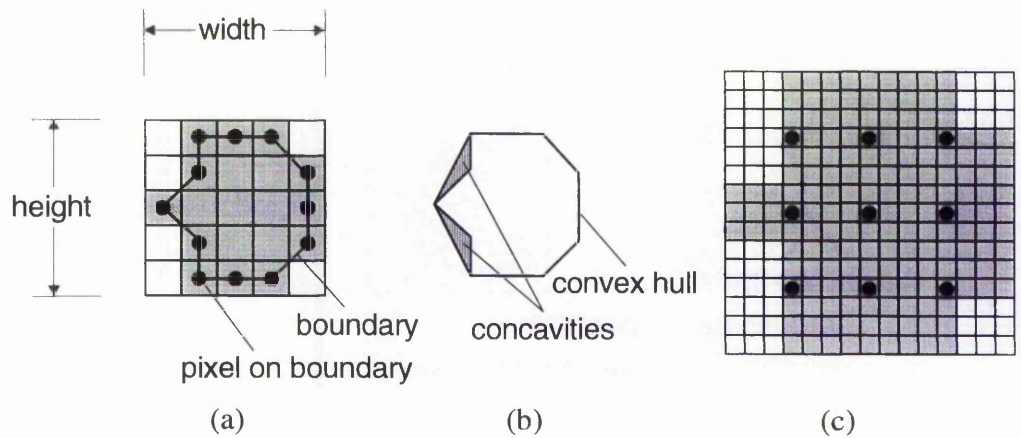


Figure 4.50: Identifying dot-shaped regions: (a) height/width ratio; black pixels/area of bounding rectangle ratio; boundary length/region area ratio; (b) area of concavities/convex hull area ratio (c) no white pixels are allowed inside the area [Hennig et al 97]

Criterion	acceptancy range	ideal value (circle)	used value	value observed in the above Figure
aspect ratio of bounding box	$\frac{height}{width} \in [\gamma_{hw}^-, \gamma_{hw}^+]$	$\gamma_{hw}^- = \gamma_{hw}^+ = 1$	$\gamma_{hw}^- = 0.6$ $\gamma_{hw}^+ = 1.7$	$\frac{5}{5} = 1.0$
black pixels to bounding box area	$\frac{blackPixels}{boundingArea} \in [\gamma_{bb}^-, \gamma_{bb}^+]$	$\gamma_{bb}^- = \gamma_{bb}^+$ $= \frac{\pi}{4} \approx 0.79$	$\gamma_{bb}^- = 0.6$ $\gamma_{bb}^+ = 1.0$	$\frac{19}{25} \approx 0.76$
circumference to area ratio	$\frac{\sqrt{blackPixels}}{circumPixels} \in [\gamma_{cc}^-, \gamma_{cc}^+]$	$\gamma_{cc}^- = \gamma_{cc}^+$ $= \frac{1}{2\sqrt{\pi}} \approx 0.28$	$\gamma_{cc}^- = 0.2$ $\gamma_{cc}^+ = 1.0$	$\frac{\sqrt{19}}{12} \approx 0.36$
area of concavities	$\frac{concavityArea}{convexHullArea} \in [\gamma_{cv}^-, \gamma_{cv}^+]$	$\gamma_{cv}^- = \gamma_{cv}^+ = 0$	$\gamma_{cv}^- = 0.0$ $\gamma_{cv}^+ = 0.3$	$\frac{1}{13} \approx 0.077$
white pixels inside region	no white pixels allowed	test all pixels in the circle	test selected pixels only	all 9 pixels are black

Table 4.3: Criteria to identify dot-shaped objects [Hennig et al 97]

Each of the above constraints accepts some objects as dots that actually are not dots. Therefore all constraints are applied together and only objects which satisfy

all of the constraints are considered to be dots. The effect of all constraints on the sample image is shown in Figure 4.51, successfully identifying only true dots.

```

-!"£$%^&*()_+~`{|}[]@'::;<>? ,./|\0123456789
abcdefghijklmnopqrstuvwxyz
ABCDEFGHIJKLMNOPQRSTUVWXYZ

```

Figure 4.51: The effect of all constraints

Once a dot has been detected in an image, its position (origin) and two extents (EX1 and EX2) are calculated and stored as follows:

The origin is the centre pixel of the dot's bounding rectangle (see Figure 4.52).

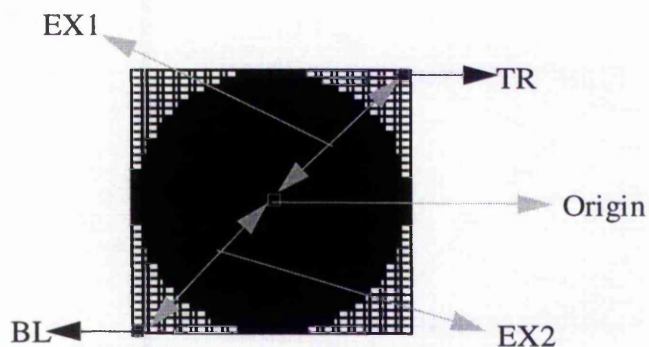


Figure 4.52: Storing feature details

Two extents 'EX1', 'EX2' are the distances between origin and top right corner (TR(X,Y)) and bottom left corner (BL(X,Y)) of the bounding rectangle respectively (see Figure 4.52).

$$EX1(X,Y) = \text{Origin}(X,Y) - \text{TR}(X,Y)$$

$$EX2(X,Y) = \text{Origin}(X,Y) - \text{BL}(X,Y)$$

4.8 Summary

This chapter describes different methods for extracting different important features in a given object, which represent the ideal form of the object. The features extracted are vertical bars, horizontal bars, left side open, right side open, top side open, bottom side open, top left corner open, top right corner open, bottom left corner open, bottom right corner open, centre of gravity, dots of i and j , zone. Each feature contains different information such as its position in the object (origin) and length and width (extents). The developed methods are able to extract these features successfully in different objects.

Chapter 5

Implementation

5.1 Introduction

In this chapter, implementation of a recognizer system is described. The method for automatic development of databases both of single and touching letters of any font and point size is presented. A method for finding different letter combinations in the words of the dictionary used is described and is used for creating touching characters. Two methods for creating touching letters artificially namely 'Undercut' and 'Adding noise' have been developed. The Undercut method tries to move letters closer physically by reducing space between them and hence forcing them to touch. The Adding noise method tries to grow letters from their different parts in order to make them touch.

The first step for the recognizer is to find alternatives of each object of the sample word. For this, a separate table of alternatives is used for each object, which can store a maximum of 100 along with their total cost. It has been observed through experiments that a table of 100 alternatives for a particular object is sufficient to store correct alternatives of that object. A table with less alternatives

may not store the correct alternative in the case of a poor quality object. Storing a large number of alternatives (more than 100) for each object is good, but needs more computational time for the second step of the recognizer (making words using alternatives and dictionary lookup). The second step involves the searching of dictionary words in the tables of alternatives using a recursive search.

5.2 Finding existing combinations of letters in a dictionary

5.2.1 Introduction

In the present work a 4k word dictionary is used. This dictionary contains words used frequently in everyday life. A letter combination can consist of 1, 2, 3 or more letters, up to the maximum word length in the dictionary (typically sixteen). This is done so that whenever we get an object formed by touching letters in our sample word, we may be able to recognize it, using the database of touching letters without trying to segment into single letters. By forcing maximum combinations of letters to touch artificially, we can generate their database. In this way, we bypass errors occurring during a segmentation stage. It has been observed that half of the errors in recognition are due to incorrect segmentation of touching objects [Chen and DeCurtins 93].

The used dictionary can have words of three types, which are explained below:

Type1: This type includes words in which each character is lower case, for example, ball, character, etc.

Type2: Each character in a word is upper case, for example, BALL, CHARACTER, etc.

Type3: In this type, the first character of the words in upper case and the rest are always lower case, for example, Ball, Character, etc.

Hence there are a total of 12000 words in the dictionary used.

The method for finding all possible combinations of letters in the dictionary is described below:

5.2.2 Method

Consider a dictionary with just two words, 'cake' and card'. When the three types of word are considered, we then have six words, as shown:

cake, CAKE, Cake, card, CARD, Card

Now we want to find all 1, 2, 3 and 4 letter combinations existing in this set of words, 4 being the maximum word length in this small dictionary.

Our list of 1-letter 'objects' therefore becomes

c, a, k, e, C, A, K, E, C, a, k, e, etc.

We then sort this list alphabetically and remove duplicate entries, giving the following list.

A, C, D, E, K, R, a, c, d, e, k, r

Two character combinations are initially listed as:

ca, ak, ke, CA, AK, KE, Ca, ak, ke, etc.

Again we sort this list alphabetically and remove duplicate entries, giving all possible combinations of 2 letter in this dictionary as follows:

AK, AR, CA, Ca, KE, RD, ar, ca, ke, rd

We repeat this for three and four letter combinations.

The method remains the same however many words are in the dictionary.

5.2.3 Analysis

Table 5.1 describes the number of letters combined, total existing combinations in different type of dictionaries and net total existing combinations.

Letters combined	Total existing combinations			Net total existing combinations
	Type1	Type2	Type3	
2	402	402	195	999
3	2541	2541	1038	6120

Table 5.1: Total number of existing combinations for 2 and 3 letters in each type of dictionary and their net total

Compare total *existing* combinations with total *possible* combinations as shown in Table 5.2. Clearly there are fewer existing combinations (for example, letter pairs such as 'zz' and 'qx' don't exist in any words in the dictionary). It is unnecessary and inefficient to store letter combinations which do not exist.

Letters combined	Total possible combinations			Net total possible combinations
	Type1	Type2	Type3	
2	676	676	676	2028
3	17576	17576	17576	52728

Table 5.2: Total number of possible combinations for 2 and 3 letters in each type of dictionary and their net total

5.3 Automatic database development

5.3.1 Introduction

The aim of this piece of work is to use the feature extractor described in Chapter 4 to find the features of known objects, thus creating a database for use in identifying unknown objects.

The input is a 4k dictionary of words. From this, a list of all possible objects (from single characters up to a maximum of, say 16 touching characters) occurring in the dictionary are created. A copy of this list is then converted into a format readable by the feature extractor, which then finds the features of each object. The features are matched with the known objects from the original list to give an automatically created database.

The database is automatically generated using a single parameterized program. It can be generated for single and touching letters of any font and point size. As described previously (Section 4.1), each feature has a number of attributes or dimensions giving comprehensive information, and this is stored in the database.

5.3.2 Overview of method

There are a number of stages in the development of the database. The first is simply to list each possible combination of one or more (up to a pre-defined maximum) letters in the dictionary. Then for each set of combinations of each number of letters the following steps will be taken.

- Convert the list of combinations to a postscript file (the most suitable format for adding required information)
- Convert postscript file to a TIFF file (feature extractor cannot read postscript file)

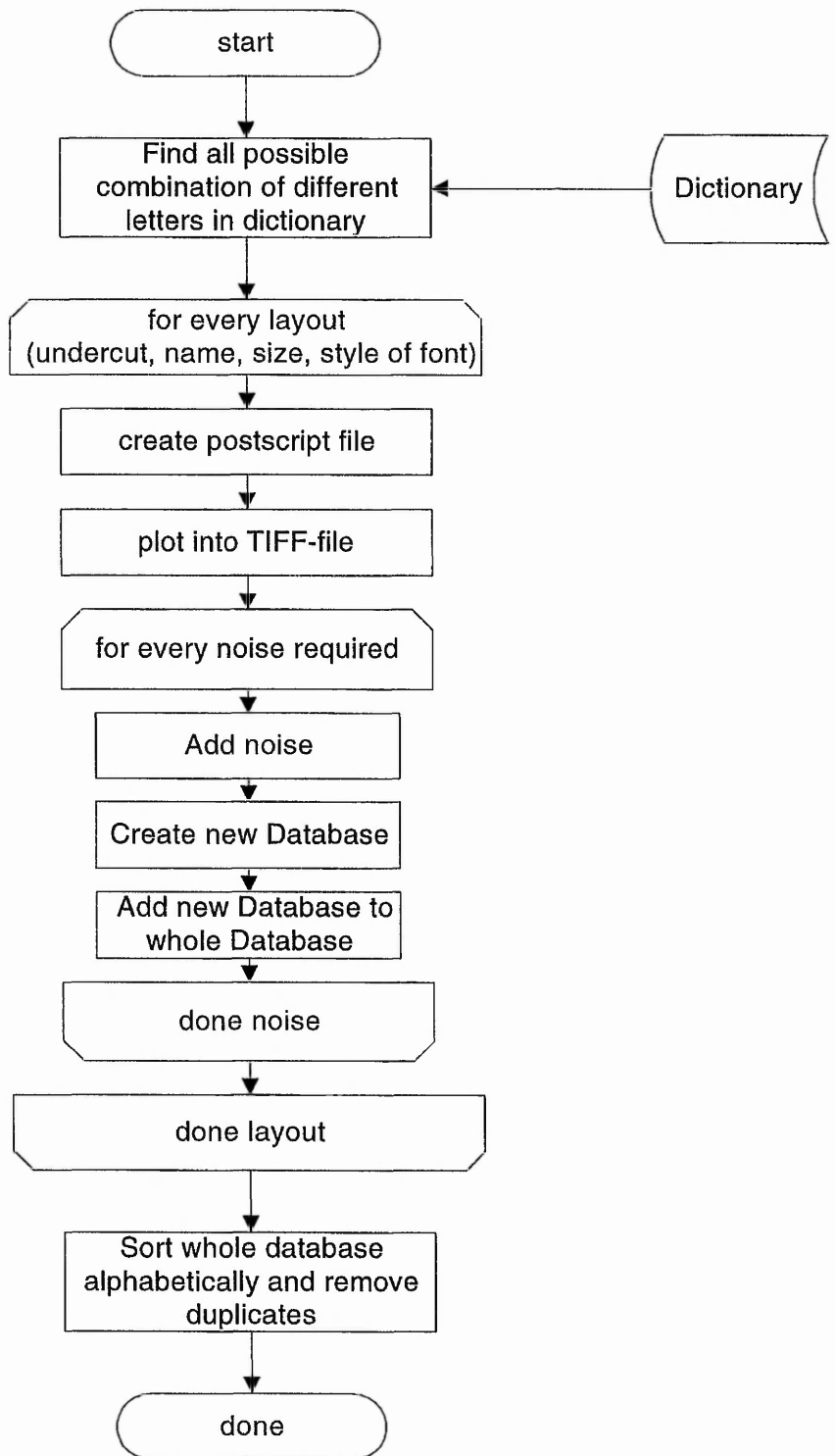


Figure 5.1: Overview of the method

- Use feature extractor on TIFF file to create database

The reasons for these steps are described later. See Figure 5.1 for a diagrammatic overview of the method.

Before describing these steps, the result of finding all possible objects of x letters (see Section 5.2) is shown here.

Consider a very small 3-word sample dictionary shown in Figure 5.2 (the actual method uses a 4000 word dictionary).

apple
hat
loaf

Figure 5.2: A sample dictionary

This dictionary is used to produce a set of list of ‘objects’ of length 1,2,3 etc. For this sample dictionary this results in a set of lists as illustrated in Figure 5.3.

1	2	3	4	5
A, E, F, H, L, O, P, T, a, e, f, h, l, o, p, t	AF, AP, AT, Ap, HA, Ha, Le, LO, Lo, OA, PL, PP, af, ap, at, ha, le, lo, oa, pl, pp,	APP, App, HAT, Hat, LOA, Loa, OAF, PLE, PPL, app, hat, loa, oaf, ple, ppl	APPL, Appl, LOAF, Loaf, PPLE, appl, loaf, pple	APPLE, Apple, apple

Figure 5.3: Possible objects in the sample dictionary

Later in the method, objects with more than one letter can be artificially joined in a process known as ‘undercutting’ and ‘adding noise’ (Section 5.4). This allows features of touching letters to be extracted and added to the database.

The reason for these sets and artificially joining and creating a database of touching letters is as follows. In a word, one or more characters may touch, creating a single object. This is especially likely in poor quality documents. Rather than segmenting such objects, the entire object is recognized by matching features with the features of different objects in the database. Hence we avoid errors during the segmentation stage of typical OCR systems.

The next part of the method begins the process of putting these lists of objects into a format which the feature extractor can read (TIFF).

5.3.3 Converting to postscript format

A program has been written for converting ASCII text into postscript. The format of the postscript file is important, mainly in preparing the data for use by the feature extractor.

The program takes as input the text, plus its font name, font size and font style, and automatically converts this data into the postscript format. An undercut value is also given to the program, the need for which is described later (Section 5.4). The output postscript file has four sections, as illustrated in Figure 5.4.

Section 1 is an encoding of font name and style, section 2 is an encoding of the object separation. Section 3 contains user information, section 4 contains the objects themselves in an appropriate format. These sections are described below.

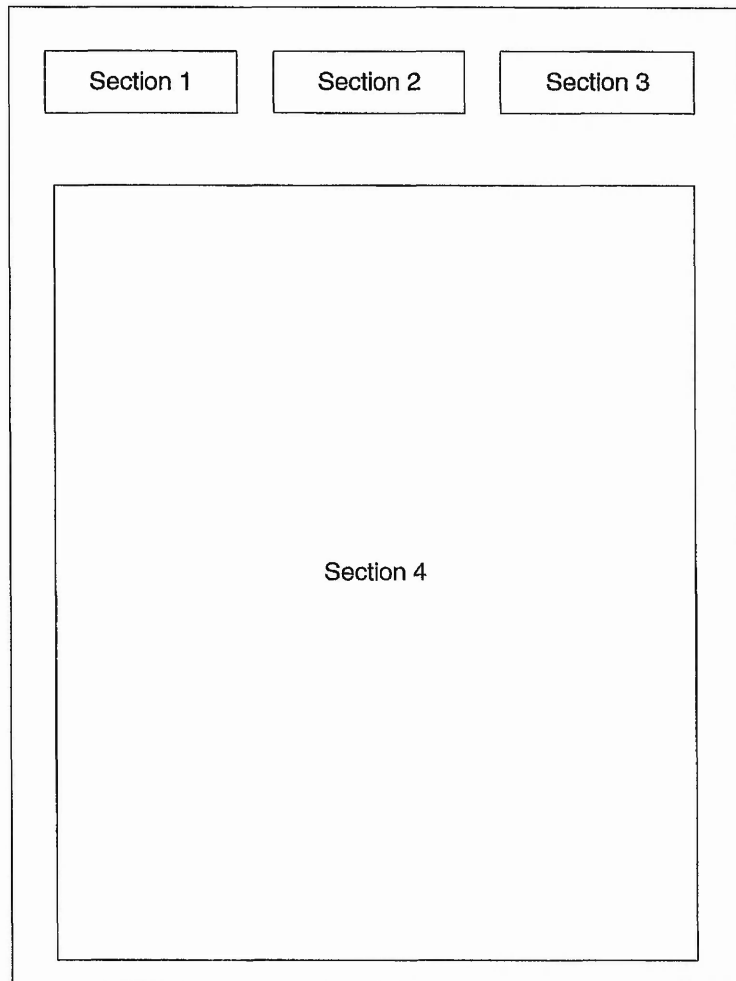


Figure 5.4: postscript file format

5.3.3.1 Section 1

This is an encoding consisting of pipe (|) and dash (-) characters. A pipe indicates a binary '1' or 'set', a dash a binary '0' or 'not-set'. The reason for the encoding is that it is later used by the feature extractor. So this encoding tells the system the font details (compare with section 3 in Figure 5.4, which tells the *user* this information).

It consists of nine characters; for an illustration of an example encoding, see Figure 5.5.

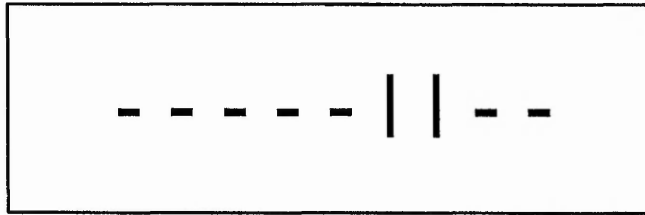


Figure 5.5: A sample encoding

The first character is always a dash. This simply indicates that the next eight are the code characters.

The next four characters are currently unused, but will be available for further information needed to be encoded here. Therefore they are all dashes at present.

The final four characters are used to represent font name and its style, which, for the limited number of fonts being considered is sufficient. The setting for these characters are as follows (see Table 5.3).

Setting	Pipe ()	Dash(-)
Spacing	variable	fixed
Serif	serif	no-serif
boldness	bold	non-bold (plain)
italicism	italic	non-italic

Table 5.3: Setting of the code characters

Some example settings for these last four code characters are as follows:

- -|| Helvetica, bold, italic (fixed spacing, no serif, bold, italic)
- | - - Courier, plain (fixed spacing, serif, non-bold, non-italic)
- ||| - Times Roman, bold (variable spacing, serif, bold, non-italic)

5.3.3.2 Section 2

This consists of two pipes, with their separation indicating the gap between two objects in the main part (Section 4) of the postscript page. This is useful for when the lines of objects are being used to find each object. By insisting on a fixed spacing, and informing the feature extractor of this at the beginning of the file, objects will be found easily, accurately, and, of course, quickly.

5.3.3.3 Section 3

This follows the two separation pipes described above, and simply provides user information. Typically the ASCII text equivalent of the first encoding is placed here; font name and its style, and perhaps the undercut value, etc.

The remaining section contains the objects themselves. It should be noted that the ASCII text is only converted to postscript using the one font size, as each feature is size independent. We do not need to create a database for each point size of a particular font. Therefore size is not needed in the encoding line. Note though the following important point.

Generally, the features which have been chosen are found to be font independent. Therefore it might appear unnecessary to create a database for more than one font. However, it has been observed that some letters of one font have some features that differ from the features of the same letter in a different font. As an example, consider Figure 5.6.

Here, the Courier 'I' will have left and right side open features. The Helvetica 'I' will have none of these features present.

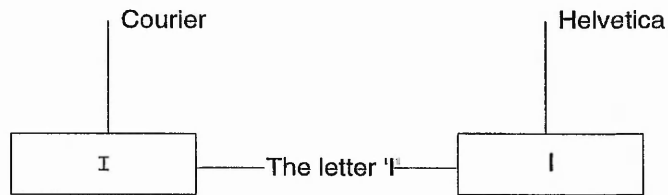


Figure 5.6: Font dependent features example

Therefore, we use the font encoding to create a database for each of three fonts Helvetica, Courier and Times Roman. However, note that during current research only single character databases are created for different fonts. For objects of two or more characters, only Times Roman font database is created and used. This is done both for efficiency and also to demonstrate that a single font database can be used for the recognition of text of different font styles and point sizes. Furthermore, this font has maximum touching characters among all used fonts.

5.3.3.4 Section 4

The remainder of the postscript file consists of a number of lines of objects. Each line has the following format.

- (i) Start pipe
- (ii) Objects separated by a specified length gap
- (iii) End pipe

The start and end pipes have an important function. They serve to indicate the start and end limits of the line, which is useful to the feature extractor for extracting text line correctly, in the similar way to separating objects with a fixed and known gap size. Once a line has been located, the objects in that line can be found (see Section 3.4 for a description of this process).

5.3.4 Converting postscript to TIFF file

The feature extractor cannot read postscript format file, so this conversion is made. The Ghostscript program is used, at a resolution of 300 dpi. The resulting TIFF file contains the same image as the postscript file, but can now be input to the feature extractor. Figure 5.7 shows the TIFF file image for the two letter combinations in the sample dictionary.

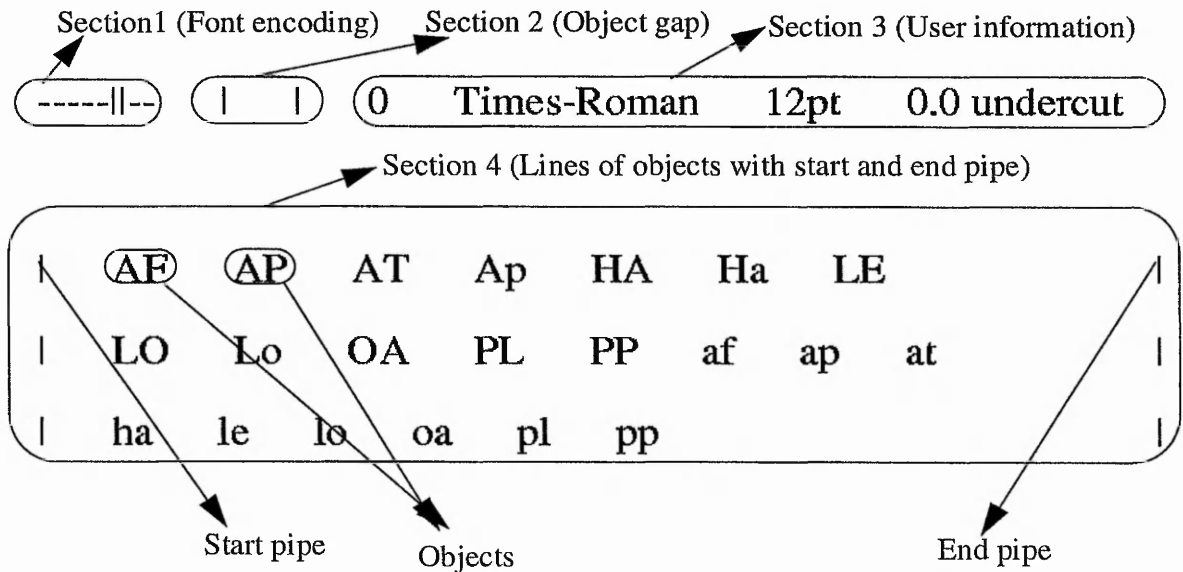


Figure 5.7: TIFF file image for the two letter combinations

5.3.5 Creating and updating the database

This process requires two files as input. The first is the TIFF file containing object images. The second is the ASCII file used to create the postscript file which then became the TIFF file. Using these two files, and the feature extractor, a database file is constructed.

The method involves a number of steps:

5.3.5.1 Find boxes in TIFF file

Each box in the TIFF file is found using the colouring connected method as described in Section 3.4.

5.3.5.2 Find text page

The text page is denoted by the four extremes of the objects on the page, as indicated in Figure 5.8.

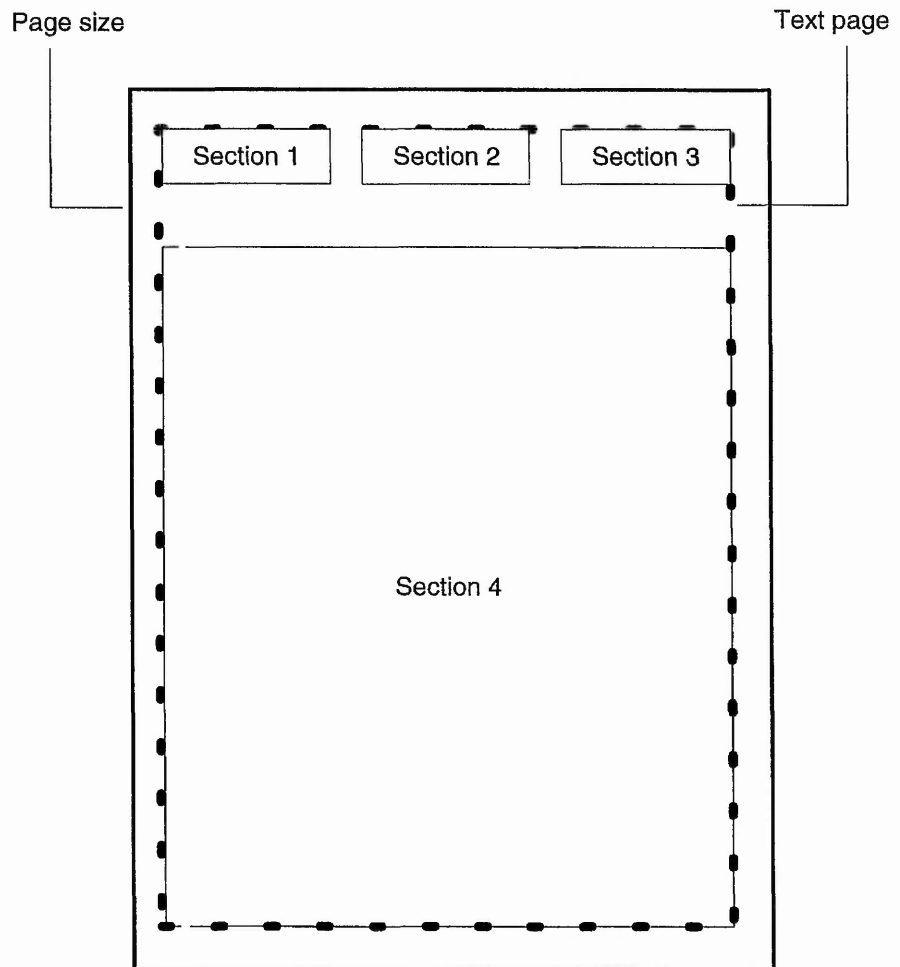


Figure 5.8: Text page

The find boxes step will have resulted in a set of box coordinates being stored. The extreme left, top, right and bottom values in this store will give the coordinates of the text page.

5.3.5.3 Find and decode encoding

The font name and style encoding, described earlier in this section, must be found and converted to a numeric value that represents the font/style.

In this step, the importance of using the dash and pipe becomes clear; two formulas determine that an object is either character:

$$\text{Dash}(-) = \text{ObjectWidth} > \text{ObjectHeight} \times \text{DashRatio} \quad (\text{EQ 5.1})$$

$$\text{Pipe}(|) = \text{ObjectHeight} > \text{ObjectWidth} \times \text{PipeRatio} \quad (\text{EQ 5.2})$$

Pipe ratio and dash ratio have been experimentally proved to give accurate results, typically, dash ratio = 1, and pipe ratio = 3.

Once the first dash has been located, the following eight objects form the encoding. By determining using the above equations, which are pipes and which are dashes, a number is calculated as follows:

Let dash (-) = 0 and pipe (|) = 1

We convert the resulting 8 character binary code to decimal. For example, consider the encoding for the plain Times roman seen in a TIFF file (Figure 5.7) as

----||--

In binary, this becomes

00001100

In decimal, this becomes 12, and represents the font name and style and will be placed in the database alongside each object.

5.3.5.4 Find separation

The next two objects will be pipes (Figure 5.7), and their separation will give the gap found between objects in the TIFF file. This step simply confirms the existence of the two pipes, and uses their coordinates to calculate and then store their separation.

5.3.5.5 Find start line pipe

The method now looks for next pipe (Figure 5.7), which will be the start of the line pipe. This causes the user information to be skipped over, as required. The start line pipe is confirmed by checking its width and height satisfying (EQ 5.2), and that its left coordinate is sufficiently close to the left of the text page. If these conditions are met, the coordinate of the start pipe are noted.

5.3.5.6 Find end line pipe

The next step is to find the next pipe, which should be the pipe at the end of the line (Figure 5.7). In order to be so, the object must satisfy the two similar conditions:

- (a) it is a pipe (satisfies (EQ 5.2))
- (b) it is closer to the right edge of the text page

If this happens, we now know the coordinates of a text line, as illustrated in Figure 5.9.

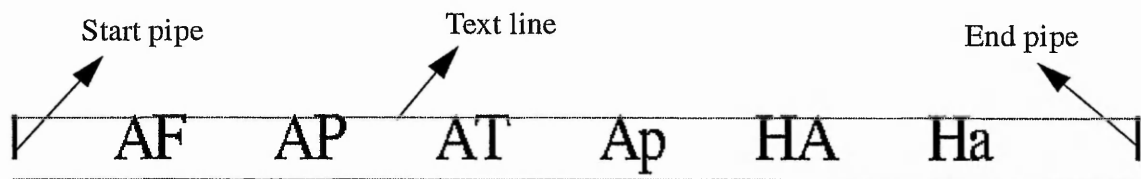


Figure 5.9: A text line

5.3.5.7 Read text line

We know the coordinates of each object box on each line of the TIFF file, and now we must find which of these objects are to be passed to the feature extractor. Figure 5.10 shows the object boxes on a typical line in a TIFF file.

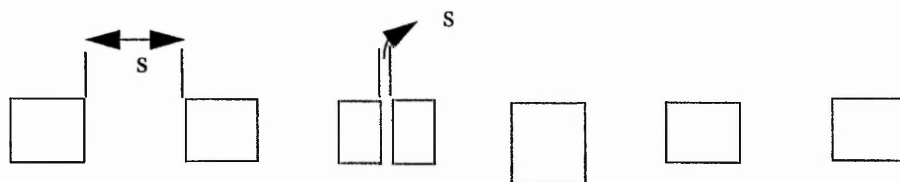


Figure 5.10: Object boxes on a line

We want to discard those objects which are not formed by touching characters - in other words, where the undercutting and the noise adding process failed to force two or more objects to touch when this was the intent.

To achieve this, we use the object separation value that was calculated from the encoding line (see Section 5.3.5.4).

Firstly, we look at the coordinates of the first object and the next object along the line, and calculate their separation. This is marked as 's' in Figure 5.10.

If this separation, 's' is greater than half of the calculated object separation value, then we deduce the two objects are not related to each other. In other words, they are not objects which failed to be made to touch. Therefore the first object is a complete object either consisting of touching characters, or a single character. We can therefore pass this object to the feature extractor.

The second case to be considered is between the third and fourth object in Figure 5.10. Because this separation is small, less than our threshold value, we can assume that the two objects are related - they should be touching but the initial gap was sufficient to survive undercutting and the adding of noise. We therefore do not want to pass either object to the feature extractor, so both objects in the TIFF file will be ignored, and the corresponding entry in the ASCII file discarded.

Objects are not discarded immediately. In some cases there may be more than two objects close together, in which case we must not discard any until a gap larger than the threshold has been found. Then the previous objects encountered since the last large gap can be discarded. This ensures that only single objects are passed to the feature extractor.

Each line of the TIFF file is processed in this way.

5.3.5.8 Add to the database

Once we have found an object suitable for addition to the database, then the object is passed to the feature extractor. Note that at this stage, we also know from the ASCII file what the object is meant to be. The link between ASCII file and TIFF file entries is maintained throughout the method, such that when the feature extractor returns a set of feature extents, these can be stored in the database file with the known object or characters which they represent, and the font.

5.3.6 Database format

As mentioned previously, it is the position (origin) and the dimensions (extents) of a feature of an object that is recorded, rather than its existence. The feature extractor therefore takes an object and extracts different features, as described in Chapter 4. For every feature, its position in the object and extents (EX1, EX2, EX3 etc.) are calculated. These are placed in the database file. Preceding these are two information; the object taken from the ASCII file and its font code; the decimal value calculated from the TIFF file encoding (font information) This gives a database entry the following format (see Figure 5.11).

Object	Font Code	Features Table (Feature Name, Origin, EX1, EX2, EX3)
--------	-----------	--

Figure 5.11: Database entry format

An example entry is shown in Figure 5.12.

IL	12	Features Table {HBar, 0 1, 0 -106, 942 0, 0 6, 0 -6}, {HBar, 0 1, 0 193, 657 0, 0 6 0 -6}, {VBar, 6 1, 171 200, 0 -300, -57 0, 57 0}, {Hole, 10 1, 228 175, 200 0, 0 -262, 200 12}, {VBar, 6 1, 514 200, 0 -300, -57 0, 57 0},...
----	----	---

Figure 5.12: An example entry database

This shows the origin and extents for some features for the object 'IL' in Times Roman plain font.

5.3.7 Completion of database creation method

Once each object on the line has been considered, and if necessary (Section 5.3.5.7), passed to the feature extractor, the next line is processed in the same way.

Once again the start and end pipes are found and the line between these limits used. The process is repeated until there are no more lines in the TIFF file, and no more TIFF files.

The database is then sorted by number of characters and then alphabetically within these. Duplicate entries are then removed. This completes the construction of our automatic database as required.

5.4 Artificially joining characters and objects

5.4.1 Introduction

Consider the two letter object consisting of 'T' and 'L'. Placing these in the TIFF file would result in the following image, (Figure 5.13).

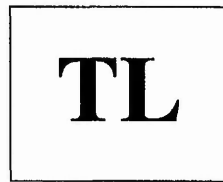


Figure 5.13: A two letter object

Passing this to the feature extractor will result in a set of feature extents being calculated. However, these would not match those found were the same two characters to be touched in a poor quality document, as shown in Figure 5.14:

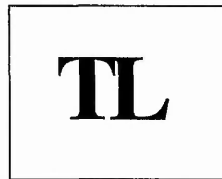


Figure 5.14: Two touching characters

Here, the bottom open feature would be found, unlike in the first case. Clearly such discrepancies will have an adverse effect in its recognition if the database of single letters is used.

The other approach is to segment touching objects into single letters and then recognize single letters using a database of single letters. Segmentation, however, is difficult in the case of poor quality documents, which contain a large number of

touching objects. As a result of erroneous segmentation, two or more objects may be grouped into one character (e.g. 'rn' into 'm' or 'cl' and 'ol' into 'd'), or one character may be segmented into two characters (e.g. 'U' into 'll' or 'h' into 'li').

Looking at the difficulties faced by segmentation based approach, we have tried to avoid it. We attempt to recognize each touching object as a whole by comparing its features with the features of known touching objects in the database, and therefore avoided segmentation errors. We therefore need single as well as touching objects in our database.

In order to obtain touching objects, the following two methods of artificially joining objects have been developed. The artificially created touching objects are expected to be similar to the unknown touching objects found in the poor quality documents. Joining has been simulated by:

- (i) Undercut
- (ii) Adding noise

5.4.2 Undercut

This method modifies the character kerning to move letters closer to each other until they touch [Raza et al 97b]. We call the degree of kerning as 'Undercut' (as used for PostScript command 'ashow'). Negative undercut moves letters closer together, whereas positive undercut will move them away from each other by a given undercut value. No undercut will only allow a few combinations to touch each other depending upon font and point size. A few different undercut values are applied to a given prototype document obtained from the letter combinations found in the dictionary used, hence forcing them to touch. The effect of different undercut values on a document obtained from some two letter combinations is

shown in Figure 5.15. For each undercut value, the features of touching combinations are extracted and stored in the database along with their known characters, later duplicate entries are removed.

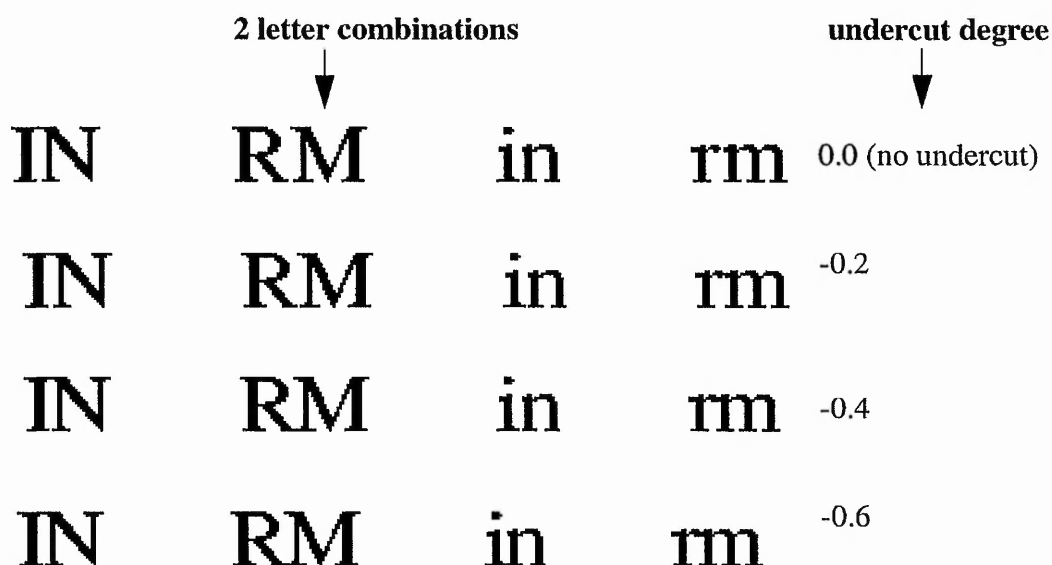


Figure 5.15: The effect of different negative undercut values on some two letter combinations

5.4.3 Adding noise

This method tries to grow letters randomly towards each other from their different parts until they touch. The touching objects obtained using this method are more similar to the touching objects found in poor quality documents than obtained by undercut. It has been observed that in poor quality documents, touching objects are usually formed by the growth of two or more single letters.

Figure 5.16 shows the outline of the method. In the diagram, letters in brackets indicate a parameter that is given by the user to that function to give best and desired results.

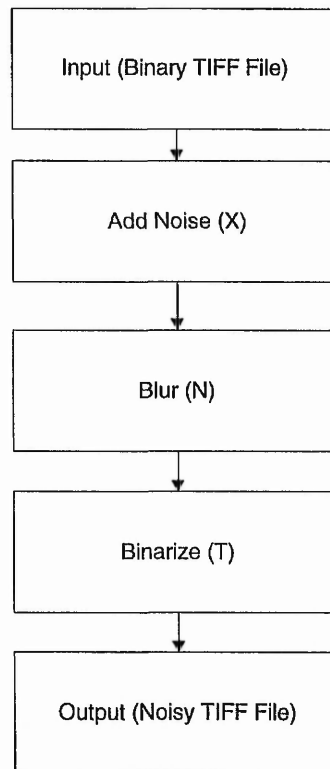


Figure 5.16: Steps in creating noisy TIFF file

In this method, the randomly selected pixels from the original input TIFF image are blackened if they are horizontally adjacent to another black pixel (i.e. randomized horizontal growth). In other words, the noise is only added closer to the letters rather than every where in the image. The noisy image is then blurred using a blur distance (N) and binarize threshold (T). That is, if more than T percent of pixels in an $N \times N$ neighbourhood of the considered pixel are black, then that pixel is also set to black (for example 20% of pixels in a 3×3 neighbourhood). This results in objects growing towards each other until they bridge gaps between combined parts of single objects or between different objects. The add noise step can modify the features of a characters (e.g. the lower open an 'h' might become a hole), whereas the blurring step can cause previously separated letters to form a single object of touching letters [Raza et al 97b]. These effects are similar to those observed in poor quality documents.

In order to obtain touching objects different kinds of noise are added and then blurred and binarized as well using different values. The effect of different noise and blur values on a document obtained from some two letter combinations is shown in Figure 5.17.

2 letter combinations									noise blur	
al	cl	ff	fl	ft	lo	m	rt	um	—	—
al	cl	ff	fl	ft	lo	m	rt	um	10%	20%
al	cl	ff	fl	ft	lo	m	rt	um	20%	17%
al	cl	ff	fl	ft	lo	m	rt	um	30%	10%
al	cl	ff	fl	ft	lo	m	rt	um	—	20%
al	cl	ff	fl	ft	lo	m	rt	um	—	10%

Figure 5.17: The effect of different noise and blur values on some two letter combinations

5.4.4 Summary

Two methods for artificially joining of single letters into touching objects have been modelled and presented. These methods are: (a) Undercut that moves characters closer to each other to a certain value, thus forcing them to touch; (b) Adding noise that grows letters towards each other until they touch. The effect of adding noise has been seen similar to that observed in poor quality documents. The touching objects obtained during these two methods have stored in the database along with their features, which are used for the recognition of unknown touching objects without their segmentation.

5.5 Recognizer

5.5.1 Introduction

The outline of the developed Word Recognition (WR) approach based on object recognition and dictionary lookup is shown in Figure 5.18. In this approach, the first step is to find different objects of an input word image without attempting to segment the touching characters. Later, each object is passed to the feature extractor, where different independent features are found, which represent the ideal form of the object.

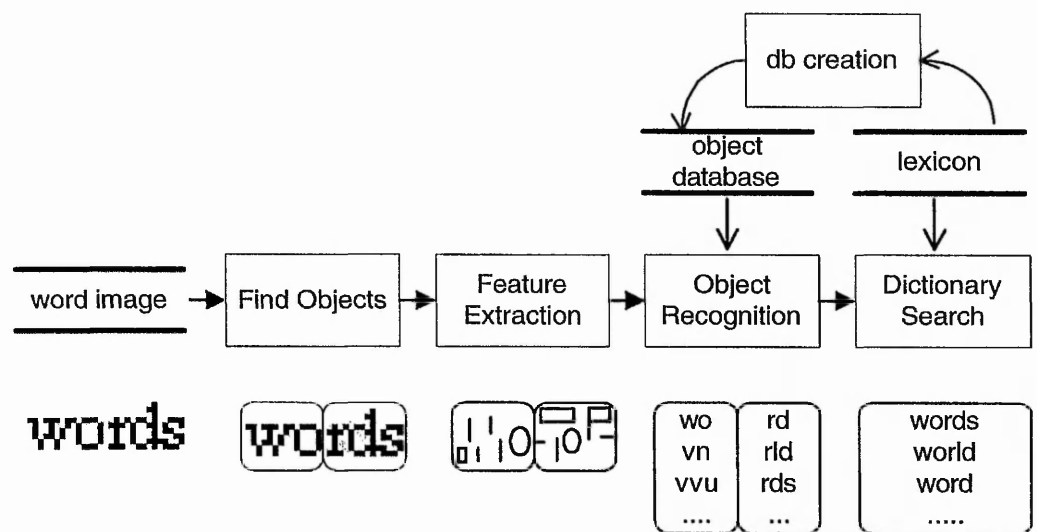


Figure 5.18: Outline of the word recognition approach [Raza et al 97b]

The recognizer developed during the current research then tries to recognize the word using object recognition dictionary lookup. The recognizer has two main steps, which are as follows:

- 1) Finding object alternatives
- 2) Building words using alternatives

5.5.2 Finding object alternatives

Once we have calculated the feature description of an object in a sample word to be recognized, we now find the possible alternatives for that object in that word. Each object may have a maximum of 100 alternatives and are stored in a separate table of fixed size along with their corresponding costs as described later in this section. The table of alternatives is kept sorted in ascending order of cost. This helps us to store the candidates with best matches (least cost) at the top of table and worst at the end of the table.

In this step, the approximate number of characters in the object (N_{app}) are estimated using its height and width with the help of (EQ 5.3) [Raza et al 97b].

$$N_{app} = 1 + \frac{width}{height} \quad (\text{EQ 5.3})$$

Only database entries that describe objects of N_{app} characters are considered for recognition of the object. (EQ 5.3) has been found to be fairly accurate for most objects but may fail for certain objects. For example, 'm' might be estimated to be two characters long while 'fi' might yield an N_{app} of 1. Therefore we also consider database entries whose lengths fulfil the range (EQ 5.4).

$$nr \text{ of letters} \in [N_{app} - 1, N_{app} + 1] \quad (\text{EQ 5.4})$$

The similarity between a database entry and the object under observation is expressed as the total cost of transforming one into the other using a modified form of the edit-distance. Insertion or deletion of one feature has been defined to have a cost of 1. The difference in the number of features of any one type (e.g. one versus three holes) is totalled over all feature types using (EQ 5.5). However if the counts are identical (2 holes versus 2 holes), the cost for transforming one feature into

another of the same type is obtained and totalled for every pair of features (EQ 5.6). The transformation cost ‘*Dist*’ between two given features can be derived from the distance of their respective origins and the differences in their extents. Its value has to be less than 2, the cost for deletion and re-insertion of the feature. The cost function ‘*Dist*’ is detailed in the following section for the various features. The sum of the transformation cost of every feature type and the cost of insertions or deletions gives the overall cost, describing the closeness of match [Raza et al 97b]. This is obtained using (EQ 5.7).

$$Cost_{(ins)/(del)} = \sum_{f \in featureTypes} |nrFeatures_f^{database} - nrFeatures_f^{sample}| \quad (EQ 5.5)$$

$$Cost_{trans,f} = \sum_{f \in features \text{ of Type } f} Dist(feature_i^{database}, feature_i^{sample}); (Dist \in [0, 2]) \quad (EQ 5.6)$$

$$Cost_{total} = Cost_{(ins)/(del)} + \sum_{f \in featureTypes} Cost_{trans,f} \quad (EQ 5.7)$$

The cost of transforming the sample feature string into a database feature string is obtained for each of the database feature strings fulfilling (EQ 5.4). The best matches (i.e. those with the lowest cost) are maintained in a sorted table, keeping the overall winner at the top. These tables of object alternatives are obtained for every object in the word.

5.5.2.1 Transformation cost of various features

The transformation cost of a sample feature and a database feature is directly derived from the origin and the extents (E_1, E_2, E_3) of the feature.

The difference between database and sample features are normalized and accumulated. Normalization can be derived from a chosen value of the three extents, using the length d of the vector E_i , as given in (EQ 5.8).

$$norm = (d(E_{f,i}) + d(E_{t,i})) / 2 \quad (EQ 5.8)$$

The four differences between the origin (O) and extents (E) of the sample feature(f) and database feature (t) are weighted by w_i using (EQ 5.9).

$$Dist = \frac{w_0 d(O_{f,p}, O_{t,i}) + \sum_{i=1}^3 w_i d(E_{f,p}, E_{t,i})}{\left(w_0 + \sum_{i=1}^3 w_i \right) norm} \quad (EQ 5.9)$$

Where $w_0=3$, $w_1=2$, and $w_2=w_3=1$. The main criteria for choosing these weights is that the origin of a feature is given more importance than the extents, E_1 is given more importance than E_2 and E_3 .

5.5.3 Building words

For each object in the sample word, there is a table containing up to a maximum of 20 alternatives. The alternative may be a single character or a combination of characters. For each table, we find the minimum and maximum number of characters amongst the alternatives.

We use these figures to calculate the longest and shortest possible words, thus giving us the range of length of words that may be formed using these tables of alternatives and dictionary lookup, i.e.

$$L_{min} = \sum_1^{no\ of\ tables} \text{minimum number of characters in table} \quad (\text{EQ 5.10})$$

where L_{min} is the minimum word length.

$$L_{max} = \sum_1^{no\ of\ tables} \text{maximum number of characters in table} \quad (\text{EQ 5.11})$$

where L_{max} is the maximum word length.

The recognized sample word may have any number of characters between L_{min} and L_{max} inclusive. Therefore only those words from the dictionary that fall within this range are selected and attempted to be constructed from the tables of object alternatives. The construction involves a recursive search across different tables.

Constructing a word from the tables corresponds to the matching of two sequences of elements (i.e. constituent characters of the word and entries of one of the tables). For this, the recursive search attempts to match the respective first element and then - if successful - the remaining elements. The entries in the first table (i.e. one possible interpretation of the first object in the word's image) determine the number of characters from the word hypothesis used for the matching. The search yields the accumulated cost of the matches involved and can be formulated below:

For every match between an entry in the currently first table and the beginning of the current part of the word hypothesis, the cost of that match (as stored in the table) is added to the overall cost of matching the remaining tables/characters. If no more tables are available to match the remaining characters against, the word

$\text{cost}(\text{characters}, \text{tables})$

$$= \begin{cases} \min_{\substack{\text{entries matching} \\ \text{start of characters}}} \left[\text{cost}(\text{first character}(s), \text{entry of first table}) + \right. \\ \left. \text{cost}(\text{remaining characters}, \text{remaining tables}) \right] & ; \text{ if end of } \text{characters} \text{ and } \text{tables} \text{ is reached} \\ 0 & ; \\ \infty & ; \text{ otherwise (i.e. matching failed)} \end{cases}$$

total cost of word hypothesis

$$= W_{\text{cost}} = \text{cost}(\text{hypothesis}, \text{tables}_{0-\text{Nb. tables}})$$

hypothesis is too long to be constructed from the objects found in the image. Alternatively, if (non-empty) tables remain after reaching the end of the hypothesis characters, the hypothesis is too short in relation to the image. In both cases, the cost is assigned infinity, i.e. the construction of that hypothesis (or latter part of that hypothesis) fails. Only if the ends of both lists (characters and tables) are reached simultaneously, the word has been constructed successfully. The costs of all the object alternatives involved are accumulated and the minimal overall cost is maintained.

A dictionary word that can be constructed from the tables of alternatives represents a possible solution. The word along with its W_{cost} is stored in a separate table called a word table containing the different word alternatives. This table can store a maximum of twenty constructed words. It has been observed that the word table of this size is enough to contain the correct solution for the sample word. After storing each constructed word along with its W_{cost} , the word table is sorted in ascending order with respect to W_{cost} . This is done, so that the word(s) with lowest W_{cost} is(are) on the top of the table and are our recognized sample word(s). If there is more than one word at the top rank with the same lowest cost or if the

correct word is found but at the lower rank, then higher level linguistic information may be used to identify the correct word [Jobbins et al 96].

5.6 Summary

This chapter describes the implementation of a recognizer system. The method for obtaining existing letter combination of different lengths from the words of the dictionary used is presented. An automatic database development method of single as well as touching letters of different fonts and sizes is also described. Using this database each object of a sample word, whether single or touching, can be recognized without segmenting touching objects.

Two methods namely 'Undercut' and 'Adding noise' for obtaining touching combinations of letters artificially has been implemented and described. The methods try to produce touching objects similar to those found in the poor quality document, which in turn are used for the recognition of unknown touching objects without segmentation. This avoids errors committed during segmentation.

The developed recognizer for the recognition of poor quality words has also been described. It has two steps: The first step attempts to find possible candidates of each object of the sample word. The found candidates of each object are stored in a separate table in ascending order according to their total cost. This helps us to store the candidates with best matches (lowest cost) on the top of table. The second step involves the building of words in the tables of alternatives using a recursive search and the dictionary. If a dictionary word can be constructed in the tables, then that word along with its W_{cost} is stored in the word table. Hence after searching all required dictionary words in the tables we get a word table in ascending order according to their W_{cost} . The word(s) with lowest W_{cost} (highest number of features matched) in the table is(are) the recognized sample word(s).

Chapter 6

Results and Conclusions

6.1 Introduction

This chapter presents the results obtained from two series of tests (real facsimile messages and artificially created documents) using the developed recognizer. The recognizer involves multiple independent features and dictionary look-up. Sample data was also tested using an OCR-based commercial software package 'READIRIS' and the results obtained are presented and discussed. READIRIS is a "high performance but easy to use" OCR package. It is a learning system but it appears to be automatic - this enables it to recognize text from a range of font styles and point sizes automatically and also to learn new fonts. It also uses linguistic context to find solutions. READIRIS can deal with most kinds of documents, however it has been reported that READIRIS cannot read text which is too dense, i.e. where nearly all letters in each word are touching.

A brief description of the collection of sample documents used for achieving these results is also given in this chapter. Finally, conclusions based on the results achieved are presented.

6.2 Experiment 1

This experiment deals with the recognition of real facsimile messages.

6.2.1 Sample data collection

From a collection of 95 sample facsimile messages, 50 were selected in an attempt to evaluate the recognizer performance under known conditions. They include letters, tables, advertisements and forms with both machine printed and hand printed text. The current research deals with the recognition of machine printed poor quality words. The machine printed text in these facsimile messages appeared to be written in different fonts, for example, Times Roman, Helvetica and Courier; the most common font appeared to be Times Roman. This font was intentionally selected for testing as it is widely used, and considered to be the most difficult for recognition, as it tends to produce more touching characters compared to other commonly used fonts. The point sizes varied from 8 point to 36 point. The 12 point size is very common in general printed text and facsimiles messages. The total number of words found in all messages was 6029. The number of words in a facsimile message were between 18 and 366 inclusive. The style of the text found was mainly 'plain', however, in some facsimile messages, italic, bold and underline style can also be seen. The number of letters in each word varied between 1 and 18.

Touching characters were observed in many documents. Some words had broken letters and some had unwanted extra-part letters (see Figure 6.1). The sample documents were scanned using the 'Hewlett Packard Scanjet Plus' scanner at a resolution of 300x300 dots per inch (dpi) into bi-level TIFF images. The software involved in scanning these documents was 'Deskscan'. All facsimile messages are shown in Appendix C.

After scanning the facsimile messages, each word in the facsimile message was manually marked. Automatic and correct extraction of word boundaries was not the aim of the present research. However, the method developed for extracting and finding possible objects works accurately with the images considered.

Since the present research deals with poor quality word recognition using dictionary lookup, numbers, integers and punctuation marks were not marked for recognition. Words including nouns were marked, even if they were not present in the dictionary. In order to make recognition possible, such words were added to the original dictionary as required.

Looking forward	Department	wishes	Burton Street	will
I am pleased to learn from	lecturers	more than 20 students		THE
January	February	sincerely	Grand	February
mentioned	Thursday	Charles	information	unless otherwise

Figure 6.1: Some poor quality words from different facsimile messages

6.2.2 Results

The facsimile messages were initially processed using an OCR-based commercial software package 'READIRIS'. This was done in order to obtain a comparative study of the developed recognizer and commercial software, and thus to show the effectiveness of the developed approach. The results obtained using the READIRIS software are given in Table 6.1.

Facsimile reference number	Total words	Recognized words	Not recognized	Percent recog (%)	Percent not recog (%)
01	244	177	67	72.5	27.5
04	249	166	83	66.7	33.3
06	366	253	113	69.1	30.9
07	267	211	56	79.0	21.0
08	221	205	16	92.8	7.2
09	174	94	80	54.0	46.0
10	146	111	35	76.0	24.0
11	126	87	39	69.0	31.0
12	183	102	81	55.7	44.3
13	152	119	33	78.3	21.7
14	157	79	78	50.3	49.7
15	69	41	28	59.4	40.6
17	21	0	21	0.0	100.0
19	36	18	18	50.0	50.0
21	107	40	67	37.4	62.6
24	115	77	38	67.0	33.0
25	46	13	33	28.3	71.7
27	46	28	18	60.9	39.1
29	36	20	16	55.6	44.4
33	52	31	21	59.6	40.4
34	102	34	68	33.3	66.7
35	69	42	27	60.9	39.1
37	221	196	25	88.7	11.3
38	108	85	23	78.7	21.3
39	56	31	25	55.4	44.6
40	40	20	20	50.0	50.0

Table 6.1: Recognition results for facsimile messages obtained using commercial software (READIRIS)

Facsimile reference number	Total words	Recognized words	Not recognized	Percent recog (%)	Percent not recog (%)
41	166	101	65	60.8	39.2
42	265	192	73	72.5	27.5
44	18	11	7	61.1	38.9
46	32	11	21	34.4	65.6
47	30	18	12	60.0	40.0
48	36	15	21	41.7	58.3
53	42	1	41	2.4	97.6
55	32	3	29	9.4	90.6
57	125	61	64	48.8	51.2
58	117	87	30	74.4	25.6
61	156	83	73	53.2	46.8
62	239	159	80	66.5	33.5
66	124	76	48	61.3	38.7
69	172	155	17	90.1	9.9
74	150	73	77	48.7	51.3
80	189	42	147	22.2	77.8
81	114	40	74	35.1	64.9
82	44	24	20	54.5	45.5
88	111	59	52	53.2	46.8
89	97	64	33	66.0	34.0
90	25	0	25	0.0	100.0
92	196	177	19	90.3	9.7
94	38	24	14	63.2	36.8
95	102	56	46	54.9	45.1
overall results	6029	3812	2217	55.5	44.5

Table 6.1: Recognition results for facsimile messages obtained using commercial software (READIRIS)

After processing these facsimile messages using the commercial software and obtaining the recognition rate, these were then processed using the developed

Facsimile reference number	Total words	Recognized words	Not recognized words	Top1 (%)	Top2 (%)	Top5 (%)	Top10 (%)	Top15 (%)	Top20 (%)	Not recognized (%)
01	244	224	20	68.9	86.5	90.2	91.0	91.4	91.8	8.2
04	249	200	49	61.8	76.3	79.1	79.5	80.3	80.3	19.7
06	366	336	30	75.1	86.9	91.3	91.5	91.8	91.8	8.2
07	267	225	42	71.2	82.8	84.3	84.3	84.3	84.3	15.7
08	221	217	4	73.3	92.3	98.2	98.2	98.2	98.2	1.8
09	174	147	27	64.9	76.4	83.3	83.3	83.9	84.5	15.5
10	146	121	25	60.3	75.3	79.5	81.5	82.2	82.9	17.1
11	126	109	17	73.0	81.0	84.1	85.7	85.7	86.5	13.5
12	183	157	26	68.3	83.1	84.2	84.2	85.8	85.8	14.2
13	152	141	11	67.1	86.8	90.8	90.8	92.1	92.8	7.2
14	157	118	39	58.0	68.8	73.9	73.9	75.2	75.2	24.8
15	69	62	7	78.3	84.1	87.0	89.9	89.9	89.9	10.1
17	21	21	0	81.0	90.5	95.2	95.2	100.0	100.0	0.0
19	36	33	3	58.3	69.4	80.6	86.1	91.7	91.7	8.3
21	107	82	25	56.1	67.3	72.0	74.8	76.6	76.6	23.4
24	115	102	13	73.0	86.1	87.8	88.7	88.7	88.7	11.3
25	46	37	9	65.2	73.9	78.3	80.4	80.4	80.4	19.6
27	46	33	13	54.3	67.4	67.4	69.6	69.6	71.7	28.3
29	36	28	8	66.7	77.8	77.8	77.8	77.8	77.8	22.2
33	52	48	4	59.6	86.5	90.4	90.4	92.3	92.3	7.7
34	102	52	50	34.3	43.1	47.1	48.0	50.0	51.0	49.0
35	69	62	7	69.6	79.7	88.4	89.9	89.9	89.9	10.1
37	221	209	12	74.7	90.0	94.1	94.6	94.6	94.6	5.4
38	108	93	15	63.0	84.3	84.3	85.2	85.2	86.1	13.9
39	56	34	22	55.4	58.9	58.9	60.7	60.7	60.7	39.3

Table 6.2: Recognition results for facsimile messages obtained using developed recognizer

Facsimile reference number	Total words	Recognized words	Not recognized words	Top1 (%)	Top2 (%)	Top5 (%)	Top10 (%)	Top15 (%)	Top20 (%)	Not recognized (%)
40	40	31	9	50.0	72.5	75.0	75.0	77.5	77.5	22.5
41	166	138	28	53.6	71.7	78.9	81.9	83.1	83.1	16.9
42	265	251	14	67.5	89.1	93.6	94.0	94.3	94.7	5.3
44	18	13	5	55.6	66.7	72.2	72.2	72.2	72.2	27.8
46	32	26	6	62.5	71.9	78.1	78.1	78.1	81.2	18.8
47	30	29	1	76.7	90.0	96.7	96.7	96.7	96.7	3.3
48	36	29	7	50.0	69.4	72.2	75.0	80.6	80.6	19.4
53	42	19	23	35.7	38.1	42.9	42.9	42.9	45.2	54.8
55	32	28	4	71.9	81.2	84.4	87.5	87.5	87.5	12.5
57	125	90	35	49.6	64.0	68.8	70.4	72.0	72.0	28.0
58	117	111	6	78.6	92.3	94.9	94.9	94.9	94.9	5.1
61	156	110	46	58.3	68.6	69.9	69.9	70.5	70.5	29.5
62	239	181	58	52.3	70.7	73.6	74.9	75.7	75.7	24.3
66	124	95	29	64.5	71.8	75.8	75.8	75.8	76.6	23.4
69	172	170	2	76.2	95.3	98.3	98.8	98.8	98.8	1.2
74	150	103	47	46.0	61.3	64.7	68.0	68.7	68.7	31.3
80	189	98	91	28.6	37.0	42.9	47.6	49.7	51.9	48.1
81	114	80	34	58.8	66.7	68.4	70.2	70.2	70.2	29.8
82	44	43	1	86.4	93.2	95.5	97.7	97.7	97.7	2.3
88	111	78	33	49.5	62.2	65.8	69.4	70.3	70.3	29.7
89	97	91	6	75.3	87.6	89.7	92.8	92.8	93.8	6.2
90	25	4	21	4.0	4.0	12.0	12.0	16.0	16.0	84.0
92	196	190	6	81.6	92.9	96.4	96.9	96.9	96.9	3.1
94	38	35	3	68.4	86.8	92.1	92.1	92.1	92.1	7.9
95	102	81	21	58.8	71.6	77.5	78.4	79.4	79.4	20.6
Overall	6029	5015	1014	61.8	74.6	78.5	79.8	80.6	81.0	19.0

Table 6.2: Recognition results for facsimile messages obtained using developed recognizer

recognizer. This recognizer is based on the extraction of multiple independent features and dictionary lookup without segmenting touching characters. The recognition results derived from the developed recognizer are presented in Table 6.2.

Performance of the developed recognizer and commercial software on different facsimile messages is shown in Figure 6.2.

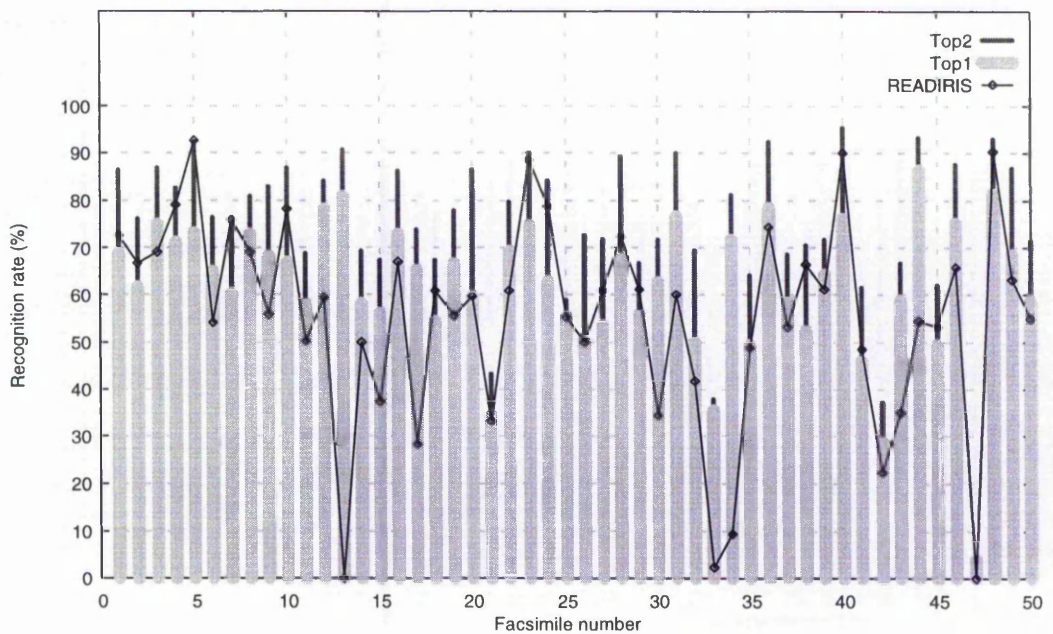


Figure 6.2: Performance of the commercial software and developed recognizer (considering Top1 and Top2 choices)

From Figure 6.2 it can be seen that READIRIS and the developed recognizer gave varied recognition rates for all facsimile messages ranging from 0% to 92.8% and 4.0% to 86.4% (considering the top1 alternatives) respectively. The main reasons for these variations are the number of touching characters, underlined words and broken characters. Therefore a low recognition rate is observed for the facsimile messages having more words containing more touching characters and underlined words, broken objects and vice versa. Overall, an 61.8% recognition

rate is obtained using the developed recognizer by considering top1 alternatives (74.6% for top2, 78.6% for top5, 79.8% for top10, 80.6% for top15 and 81.0% for top20 alternatives) and 55.5% for using READIRIS (Figure 6.3), which demonstrates the effectiveness and capabilities of the developed non-segmentation based experimental recognizer.

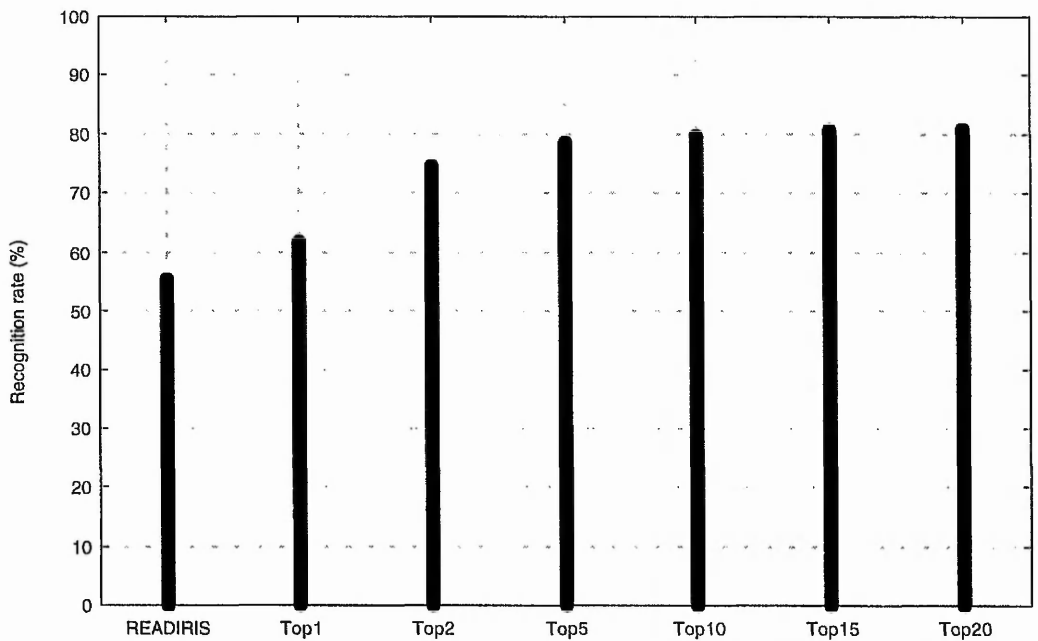


Figure 6.3: Average recognition rate for all facsimile messages using the READIRIS software and the developed recognizer by considering different choices

It has been observed that READIRIS can generally recognize words containing up to at most three touching characters. Correct recognition beyond this limit is not generally achieved. Even within this limit it fails for difficult cases. Since the software relies on segmenting touching objects, segmenting of touching objects which look similar to single letters is not feasible, for example, 'ol' and 'd', 'lo' and 'b', 'rn' and 'm' etc. Similarly, segmenting 'IN' into 'I' and 'N' or 'UI' into 'U' and 'I' is not easily achieved. Also, it has been observed that this software has difficulty in recognizing words having parts missing and broken objects.

The developed recognizer tries to recognize the whole object within a word without segmenting it. It attempts to find a number of alternatives for each object by matching its features with the features of different single and touching object in the database. Therefore for a touching object which is similar to a single object (for example 'ol' and 'd'), both of these candidates are considered, along with others. The correct candidate is then selected by looking at the candidates for the other objects within a word. That is why a higher recognition rate is obtained using the developed recognizer than with READIRIS.

The developed software can also recognize part-missing objects as well as objects containing extra parts. It will attempt to recognize objects vertically broken into two objects (by joining them to form a single object). However, it fails to recognize horizontally broken characters (e.g. 'U' broken into 'I' and 'I' etc.). In such cases, each broken object is recognized individually with the assumption that it is not broken. This is the major problem in obtaining a low recognition rate for facsimile messages containing broken objects. The developed software also faces difficulty in recognizing words which are underlined, hence making all objects in a word to touch. This happens because the features of such words are generally different from its features when touching but without the underline. The database only stores the features of non underlined touching objects.

Although the first choice for the sample words of some facsimile messages using the developed recognizer is lower than with READIRIS (see Figure 6.2), the overall recognition rate of all facsimile messages is still higher than READIRIS (61.8% for the developed recognizer and 55.5% for the READIRIS, see Figure 6.3). This may be because READIRIS takes into account a lot of information, for example, general format (layout) of the whole page, where as the current developed software works only at word level and does not know the general layout

of the document, that is, the start of the paragraph, end of the paragraph, start and end of the sentence, heading and sub-headings, etc. A further reason is that the developed method for zone extraction is not totally reliable. Also, correct zone extraction in poor quality documents containing noise is difficult, therefore zone information was not used in the recognition of these facsimile messages. These are the main reasons for getting a low recognition rate for some of the facsimile messages at top rank.

In many cases, 'of' was recognized as 'Of' as a top choice and 'of' was recognized as second choice. Similarly for single letters, for example, 'c' was recognized as 'C', 's' as 'S' and 'o' as 'O'. A zone extraction method in conjunction with layout information could address these problems. However, the top2 rank recognition rate of these facsimile messages is significantly higher than both the top1 rank for all facsimile messages and the READIRIS rates for most of facsimile messages (see Figure 6.2). A further improvement is obtained by considering the top five choices (see Figure 6.3 and Figure 6.4).

Although we have presented a top10, top15 and top20 choices obtained for sample words, these do not give a significant improvement (see Figure 6.3 and Figure 6.5). For sample words which are recognized correctly, but are at a lower rank, then the higher level linguistic information such as language syntax and semantics can be used to identify the correct choice [Jobbins et al 96]. The READIRIS software gives only one choice for each word.

Although we acknowledge the fact that the developed recognizer uses a dictionary, it has been observed that the READIRIS software does not provide sufficient output for most of the unrecognized words (see Figure 6.6). Therefore a dictionary search to improve READIRIS results does not appear possible.

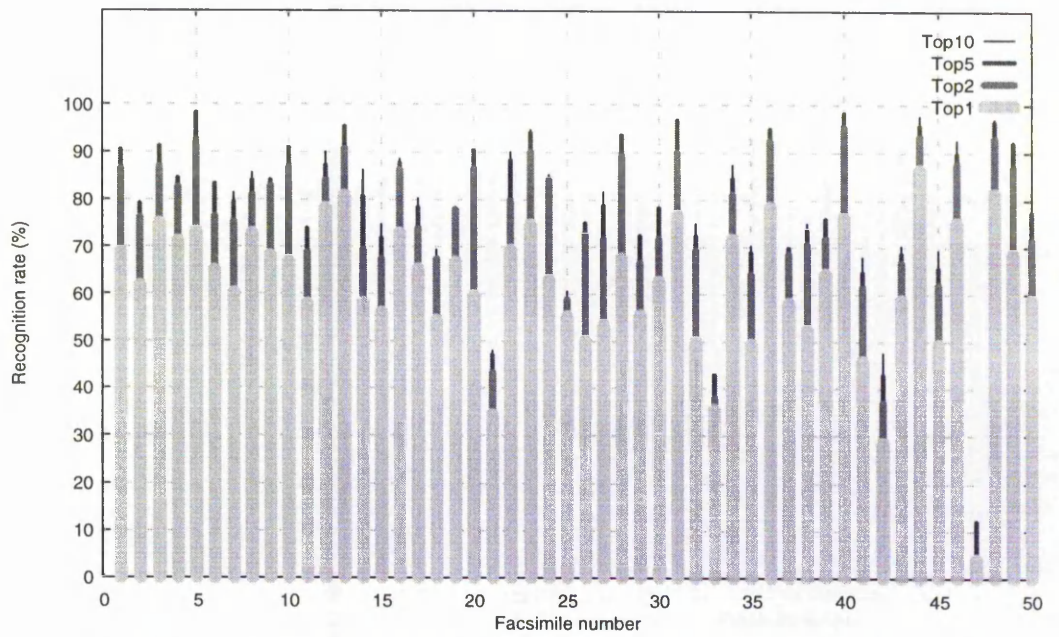


Figure 6.4: Recognition rate obtained using developed recognizer by considering different choices (Top1, Top2, Top5, Top10)

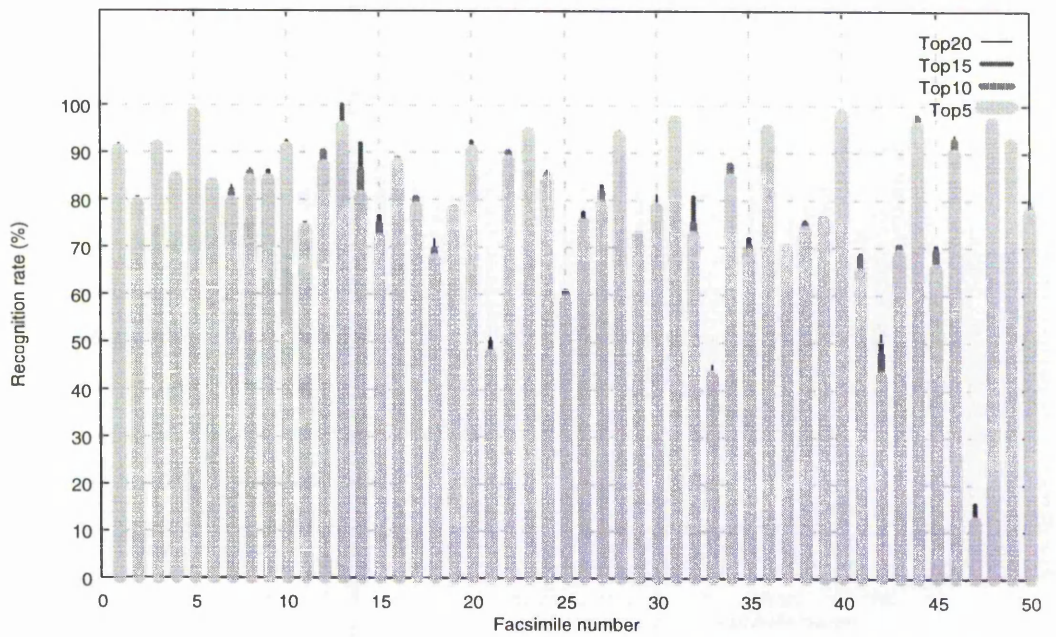


Figure 6.5: Recognition rate obtained using developed recognizer by considering different choices (Top5, Top10, Top15, Top20)

Sample words	READIRIS output	Recognizer output
ATTENTION	~IIINIION	ATTENTION
NOTTINGHAM	NOIIN~H~N	NOTTINGHAM
UNIVERSITY	IJNIVIR~IY	UNIVERSITY
approximately	approxifrlatelv	approximately
every	~ver.l}l	every
yours	yolu.S	yours
programme	programttle	programme
curriculum	cmriculUln	curriculum
information	Infonation	information
department	departntent	department
TRENT	IRINl	TRENT
mentioned	menlorled	mentioned
that	fut	that
intend	ir.ltgnd	intend

Figure 6.6: Comparing output for some sample words

6.3 Experiment 2

This experiment deals with the recognition of artificially created sample documents of varying qualities.

6.3.1 Sample data collection

A document containing 100 words of varying lengths was chosen as a sample. Three versions of this document were produced, one in Courier, one in Helvetica and one in Times Roman font as these are widely used fonts. A 12 point size was used also due to its common use in printed documents.

For each of these fonts, 5 versions of the document were automatically created by adding varying amounts of noise (see Section 5.4.3). The first contained no noise (i.e. 0% noise), then noise was introduced at 5%, 10%, 20% and 30%. Having added noise these documents were then blurred ($N=2$) and binarized ($T=20\%$). This gives a total of 15 documents, which are to be processed using both the developed recognizer and READIRIS.

By introducing noise in this way, the documents range from good quality (no noise) through to poor quality. In the literature, some methods have also been reported for document degradation models, for example, [Baird 90], [Baird 92], [Baird 93].

6.3.2 Results

This series of tests deals with varying quality, artificially created documents of different fonts. These documents were again processed using both commercial OCR software (READIRIS) and the developed recognizer in order to obtain a

comparative study. The results obtained are presented in Table 6.3 and Table 6.4 respectively.

Serial number	Font	Total words	Recognized words (%)	Not recognized (%)	Percent font recog (%)
01	Courier	100	75	25	89.2
02		100	100	0	
03		100	100	0	
04		100	92	8	
05		100	79	21	
06	Helvetica	100	100	0	87.0
07		100	100	0	
08		100	96	4	
09		100	90	10	
10		100	49	51	
11	Times Roman	100	100	0	69.8
12		100	87	13	
13		100	82	18	
14		100	55	45	
15		100	25	75	
overall results		1500	82	18	82

Table 6.3: Recognition results for artificially created sample documents obtained using commercial software (READIRIS)

The performance of these recognizers is shown in Figure 6.7. It is clear from the Figure 6.7 that a varied recognition rate is obtained for different documents, ranging from 25% to 100% using READIRIS, and 65% to 100% using the developed recognizer (considering the top1 alternatives) respectively. The main

Serial number	Font	Total words	Recognized words			Percent not recog (%)	Percent font recog (%)
			Top1 (%)	Top2 (%)	Top5 (%)		
01	Courier	100	100	100	100	0	97.6
02		100	100	100	100	0	
03		100	100	100	100	0	
04		100	98	99	99	1	
05		100	90	94	94	6	
06	Helvetica	100	100	100	100	0	87.6
07		100	100	100	100	0	
08		100	93	93	94	6	
09		100	80	80	86	14	
10		100	65	71	76	24	
11	Times Roman	100	100	100	100	0	91.6
12		100	97	97	97	3	
13		100	95	95	95	5	
14		100	90	92	92	8	
15		100	76	77	77	23	
overall results		1500	92.3	93.2	94.0	6.0	92.3

Table 6.4: Recognition results for artificially created sample documents obtained using the developed recognizer

reason for these variations is the number of touching characters. A 100% recognition rate is obtained for documents with good quality (documents with little or no noise) as the number of touching characters was low. However, a lower recognition rate is obtained for documents in which more noise was added, as the quality of these documents was poor. As the quality of documents decreases with the increased noise, recognition rates decrease as expected, due to the increased number of touching characters. NB: The surprisingly low recognition rate for

READIRIS on the Courier document with no noise is due to the character 'd' being mistaken for an 's'.

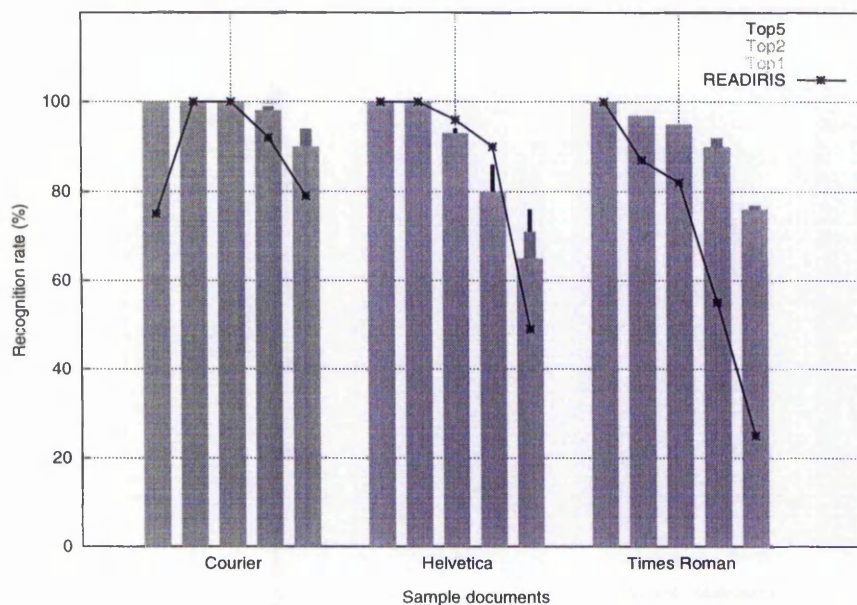


Figure 6.7: Performance of the commercial software and the developed recognizer on artificially created sample documents (considering Top1, Top2 and Top5 choices)

Overall, a 92.3% recognition rate is obtained using the developed recognizer by considering top1 alternatives (93.2% for top2 and 94.0% for top5 alternatives) and 82% for using the READIRIS (see Table 6.3 and Table 6.4). Overall performance of the developed recognizer is 10.3% better than READIRIS when considering top1 alternatives. This indicates the effectiveness of the developed approach.

The developed recognizer performed as well as or better than the commercial system for 13 out of 15 documents considered. READIRIS performed better for only 2 documents, all in Helvetica (Figure 6.7). This was mainly due to a Times Roman font database for touching characters being used during the current research. This was done both for efficiency and also to demonstrate that a single font database can be used for the recognition of text of different font styles and

point sizes. However, if one database per font is used, the recognition rate for these Helvetica fonts documents in particular, and also for other documents in general, will increase.

If we compare the recognition rates for different font styles, then it can be seen from Figure 6.8 that overall, Courier font gave the highest recognition rate of the three chosen fonts, using both systems. The behaviour illustrated in Figure 6.8 was as expected, as Courier font has the lowest number of touching characters compared to Helvetica and Times Roman. The maximum number of touching characters was observed in documents of Times Roman font, thus making them difficult to recognize.

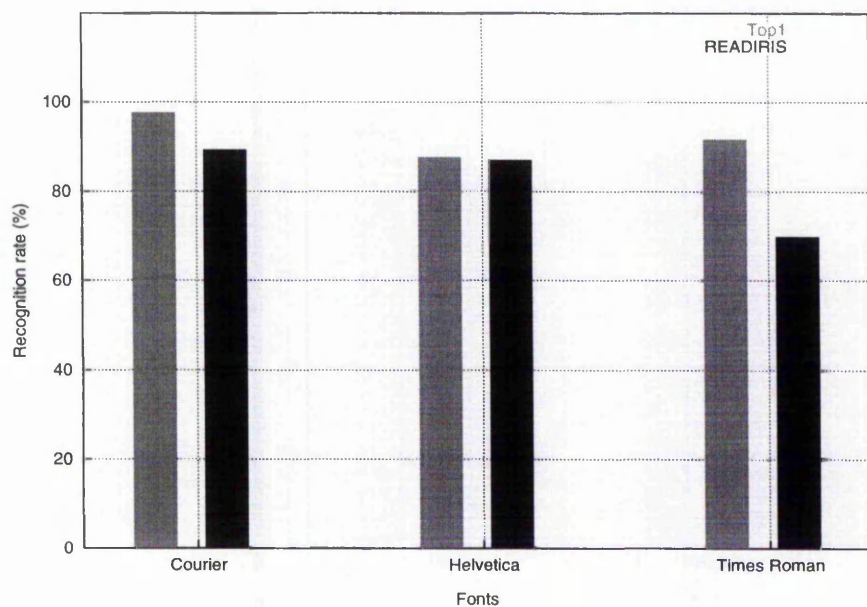


Figure 6.8: Average recognition rate for different fonts using the READIRIS software and the developed recognizer by considering top1 choice

It can also be seen from Figure 6.8 that overall the developed recognizer gave a better recognition rate for all fonts (97.6% for Courier, 87.6% for Helvetica and

91.6% for Times Roman font by considering top1 alternatives) compared to the commercial software (89.2% for Courier, 87.0% for Helvetica and 69.8% for Times Roman font).

6.4 Conclusions

The aim of the present research is to find algorithms for the recognition of poor quality documents such as facsimile messages, screen text, low quality prints, photocopies and old documents. Much work has already been done for the recognition of printed text. The systems perform quite well for the recognition of reasonably good quality documents. The performance of such algorithms for poor and degraded quality data is not satisfactory. One of the major problems of poor quality documents is that they have a lot of touching and broken characters.

The developed approach is more useful and powerful if the number of words to be recognized are limited. The developed method can also be used for the recognition of screen text [Raza et al 97a]. However, the developed approach can be used on general documents containing nouns and names of the cities, provided they are added to the dictionary.

A method for finding all possible objects in a given image using colouring connected components method has been implemented and presented. The method can find and locate every object in a given image and returns a bounding rectangle enclosing that object. It can locate objects within another object, for example, text inside a table etc. A method for finding gaps between words has been found. The method is capable of finding the gap between words, if text is written with proportional spacing. However, if the gap between different objects is not proportional, then the developed method will fail to find word gap correctly. To deal with such documents other methods need to be used. Development of such

methods i.e. document layout analysis, is not the aim of the present research. A word finding method based on a pre-calculated word gap has also been developed and presented.

Various methods have been developed in order to extract different independent features, which represent the ideal form of characters. The features extracted during the current research are top side open, bottom side open, left side open, right side open, top left corner open, top right corner open, bottom left corner open, bottom right corner open, vertical bars, horizontal bars, centre of gravity, dots of 'i' and 'j' and zones. For every feature a number of pieces of information such as its position within the object (origin) and its length and width (extents) are found. The features extracted are expected to be the same in the characters of different fonts. Therefore a database of one font is used for the recognition of all different fonts. However, the single letter database was intentionally created for each font. This is mainly because the features of Helvetica 'I' and 'g' are quite different from those characters in Times Roman. Again the database features of one point size is capable of recognition of letters of different point sizes, as the features found are normalized.

The present research deals with the recognition of poor quality words without segmenting touching characters into single letters. In order to recognize single as well as touching objects in a sample word image, a database of single and touching objects is used. A method for the automatic creation of this database has been developed and explained in detail. This method can automatically create a database for single as well as touching objects of different fonts and sizes and styles.

In order to obtain touching objects consisting of two or more letters and storing their features in the database, two methods for artificially joining single letters into

touching letters have been modelled and presented [Raza et al 97b]. These methods are: (a) Undercut that physically move characters closer to each other to a certain value, thus forcing them to touch, and (b), Adding noise, that grows letters towards each other until they touch. The effect of adding noise for obtaining touching objects has been seen to be similar to that observed in the touching objects found in poor quality documents. These two methods can create touching objects of varying quality. The touching objects obtained using these two methods are stored in the database along with their features, which are used for the recognition of unknown touching objects without segmentation. The developed method therefore bypasses errors incurred during the segmentation stage of traditional segmentation-based OCR software.

A machine printed word recognition algorithm (recognizer), which can be used for the recognition of general documents containing different font styles and point sizes and highly variable quality has been developed and presented [Raza et al 96a], [Raza et al 96b]. This recognizer tries to recognize whole word using multiple independent features and dictionary lookup without trying to segment touching characters.

The developed recognizer has two main steps, finding object alternatives, and building dictionary words using found alternatives. Each object of the sample word whether single or touching is recognized as a whole by comparing its features with the features of single and touching objects database. A number of alternatives from the database are selected based on best feature match as likely candidates for that object. Dictionary words are then tried to build using found alternatives. The dictionary words which can be built are considered as the possible candidates for that sample word. The candidate with maximum features matched is considered to be the most likely and hence the recognized sample word.

The developed recognizer has been tested on fifty different facsimile messages containing 6029 machine printed words. Words printed in these facsimile messages were of different font styles, point sizes and image qualities. Brief description regarding the facsimile messages has also been given. An overall 61.8% (81.0% by considering up top20 choices) recognition rate is achieved for all facsimile messages by considering top1 choice. These facsimile messages were also tested using commercial software, 'READIRIS' and results obtained are described. It gave an average recognition rate of 55.5% for all facsimile messages. READIRIS produces only one output for each sample word. An improvement of 6.3% is found using the developed recognizer, which shows the recognizer's efficiency in dealing with poor quality documents.

The developed recognizer was also tested on fifteen artificially created documents with different levels of quality ranging from good to poor. These documents were written in three different fonts, Courier, Helvetica and Times Roman, of point size 12. These documents were also tested using commercial software. With the exception of 2 out of 15 documents, the developed recognizer gave as good or better recognition rates compared to the READIRIS. Overall, the developed recognizer achieved an average 92.3% recognition rate (considering top1 alternatives only) for all considered documents compared to 82% for the commercial software. Hence an improvement of 10.3% is achieved using the developed recognizer compared with the commercial software. Furthermore, recognition rate of each font using the developed recognizer is better than the respective font recognition rate using the commercial software. These results confirm the effectiveness of the developed recognizer compared with the commercial system. Also bear in mind that the developed recognizer only used a Times Roman font database for touching characters, for reasons outlined previously (Section 5.3.3.3) and therefore recognition rates for Helvetica and

Courier font documents could be further improved by using touching character databases in those fonts.

The developed method found correct candidates although they were not ranked at top choice sometimes. In such cases, higher level linguistic information such as language syntax and semantics may be used to identify correct choice. A higher recognition rate can be obtained by reducing the size of the dictionary and using a larger database containing touching objects of different fonts and print qualities. Further more, the developed recognizer can be used as a preprocessor to produce a list of possible candidates. Another segmentation-based method can then segment poor quality words based on the features. Correct candidates can then be found within the list produced having most features matched.

6.5 Summary

In this chapter, the results obtained using the developed recognizer are described. In first series of test, fifty different facsimile messages were considered to evaluate the performance of the recognizer. They included letters, tables, advertisements and forms with both machine and hand printed text. Second series of tests involved the recognition of artificially created sample documents of varying fonts and qualities. The facsimile messages and artificially created documents were also tested using an OCR-based commercial software package in order to obtain a comparative study of the developed recognizer and the commercial software. Overall recognition rate of both series of experiments using the developed recognizer is higher than the commercial software. Conclusions based on the results achieved are also presented.

Chapter 7

Future work

7.1 Introduction

The developed recognizer involves extraction of multi independent features and dictionary look-up without segmenting touching characters. The results achieved, and the work of other researchers, demonstrate that a lot of work remains to be done in the area of OCR, described earlier. Although a significant amount of work has been done during the present research and encouraging recognition results have been achieved from poor quality facsimile messages as compared to the commercial software, still at the moment, there is a wide gap between the human capabilities and machine capabilities for reading text, especially poor quality text documents. Humans can read very poor quality documents. In order to narrow this gap, if not bridge it, further research effort is required.

As mentioned earlier, the approach used during this research is satisfactory and hence so far we have achieved fruitful results, but still it is not applicable for general documents of any font style and image qualities. Bearing in mind these limitations it is suggested to expand the system, so that it is valid and useful for the

recognition of poor quality documents of any kind. The proposed future work involves line and words extraction, feature extraction, document and context layout analysis, postulation algorithm, methods for dealing with touching objects and postprocessing. It is envisaged that implementation of this work will improve the performance of the developed recognizer in particular, and narrow the gap between human and machine capabilities of reading text in general. Hence it will enable us to achieve the prime aim of OCR research.

7.2 Line and word extraction

An attempt has been made to find the lines of text in documents. After that, words within a line are found. Methods have been developed and used to extract this information. Since the project mainly concentrates on the recognition of poor quality documents, we did not aim to develop more powerful methods in order to find text lines and word boundaries. The developed methods for finding word boundaries and text lines in poor quality documents require improvement.

7.3 Feature extraction

Feature extraction is an important stage in the recognition of characters. There is no doubt that the central issue in character recognition lies in the detection of features [Suen et al 68]. In order to develop a recognition system, it is necessary to have a mechanism to extract various important features of an object, which will exhibit the distinctive characteristics of the character. The extracted features should have the capability to discriminate one class of objects from another. The more reliable the features, the more successful the recognition system will be.

Bearing in mind the importance of feature extraction in the recognition system, it is suggested that research to improve existing methods and extract new features should be considered.

7.3.1 Improving existing methods

In the present research work, a number of important features are extracted such as holes, top side open, bottom side open, left side open, right side open, vertical bars, horizontal bars, centre of gravity and zone. Although most of the method for the extraction of these features works well, the methods need to be modified for general use. For example, the method for finding vertical bars can only extract absolute vertical bars. Tilted verticals bars (in skewed text or in italic text) cannot be successfully extracted using the developed method. Similarly the zone extraction method is not very reliable for extracting letter zones in general text. Future work may involve the modification and improvement of these two methods in order to get better results.

7.3.2 Extracting new features

Although the number of extracted features for the current recognizer are sufficient to classify objects most of the time, an increase in the number of features would improve the recognition. Hence future work may include the extraction of some more reliable features to achieve the desired aim. Some of the features which seem useful to extract are described as follows:

i. End point

An end point is a very common feature. The number of end points can be used to group a character image. An end point is often, but not always, indicated by one-connected pixel in the thinned version of the image. The letter 'A' for example has

two end points, located at the bottom. The letter 'x' or 'X' have four end points one on each corner. Combination of end point feature with other features will give a better classification of the sample objects.

ii. Density

Density is another feature, which can be used as one component of a feature vector to classifying different objects, since some characters are more dense than others. Density of a given object is defined as the ratio of the number of black pixels to the total area of the region.

7.4 Document and context layout analysis

Future work may include the extraction of auxiliary context information from document layout analysis. This will extract the main information about a particular document to help and improve the recognition. The information extracted from the document includes the quality of the document, detection of font family and point size, start of paragraph, end of paragraph, start and end of sentence, heading and subheadings, foot notes, page numbering etc.

7.5 Postulation algorithm

7.5.1 Introduction

Often, in poor quality documents, some part/parts of the word is broken, touching or missing. There could also be some unwanted extra parts in the word. The features extracted from such objects are not similar to the features of such objects in their ideal shape. Hence objects with such properties shall be known weak objects. Other objects will be called strong objects. If we directly apply the

recognition algorithm to such words, then we may get inaccurate recognition results, simply because of weak objects. In such circumstances, word postulation may be useful. Word postulation is a step towards whole word recognition. However, this does not mean that a given word cannot be segmented into smaller parts i.e. into characters. We can still segment the word into smaller objects and also can segment touching objects into single letters. We then recognize the strong objects and use postulation algorithm to postulate weak objects and then finally recognize the whole word.

The whole aim of word postulation is to recognize the strong objects of a given word and to try to postulate (guess) some possible candidates for the weak objects guided by the presence of strong objects. Lexicon lookup can be used to support postulation. In this way we may be able to discover the weak objects and hence recognize the whole word correctly. After developing a potential power word postulation algorithm, it can be combined with the developed recognizer to see its effect in coping with poor quality documents.

There are different types of word postulation to be used, for example postulation based on word ending, postulation based on vertical bars. Word ending postulation is generally applicable and useful in handwritten text recognition. Sometimes people write the beginning of the word clearly and tend to get sloppy near the end of the word especially when the words have got the common ending such as “ed, ing, etc.” [Powalka 95]. Word ending postulation is useful for such cases. However, if the stem of the word is not legible then word ending postulation will fail to produce the correct result. The following is a suggested method for using word postulation in poor quality documents.

7.5.2 Proposed method

The features of each object of a given sample word are extracted. The strong and weak objects are marked based on the quality of the features extracted. The strong objects are recognized and different alternatives for them are found. Weak objects are marked as unknown. Note that a weak object could consist of more than one character.

We now have a situation where a given word consists of strong objects, for which there may be alternative characters, and weak objects, for which any character or characters may be substituted. The dictionary is then used on this word, by looking up words with character alternatives at the position of strong objects and any other character, or characters at the position of weak objects. This should provide possible words, which in turn give possibilities for the weak objects. Thus we have used strong objects to postulate weak objects with the help of dictionary and hence recognize the whole word correctly. This can be explained by considering the following example:

Suppose we want to recognize sample word shown in Figure 7.1.

Features of each object of this word are extracted. We analyze the features to distinguish between strong and weak objects. We find that object 1, 2 and 4 are strong objects and object 3 is a weak object.

We now find alternatives for the strong objects and put a question mark (?) at the position of the week object. This is shown in Figure 7.2.

We then use the dictionary to look for the words having found letter alternatives of objects 1, 2, 4 at their positions, and any other letter or letters at the position marked by '?'. Some of the words found using this method are ("WOOD",

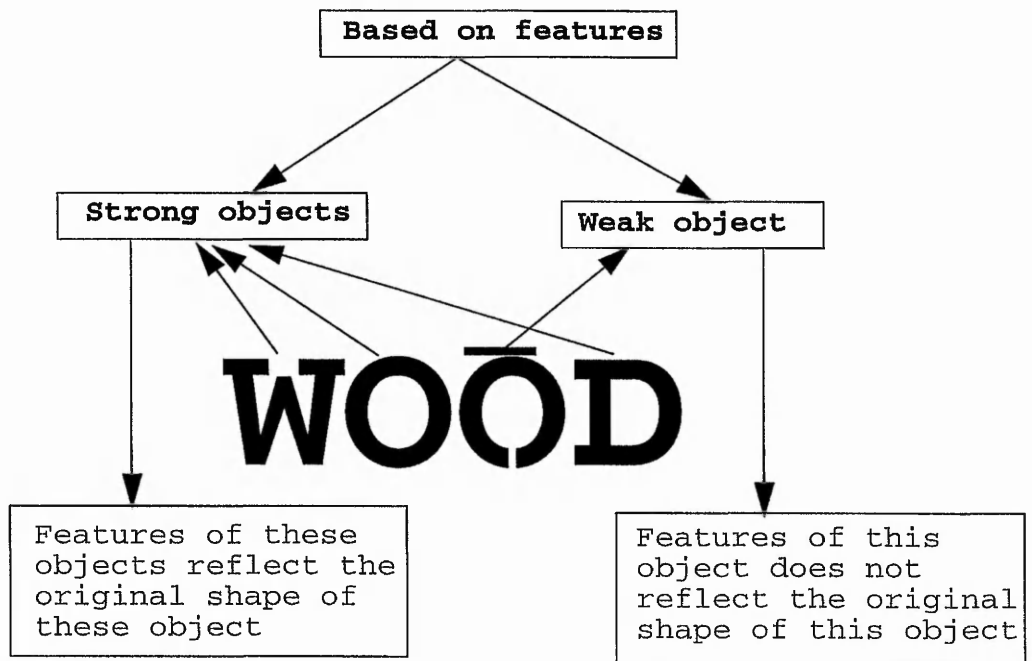


Figure 7.1: Sample word for recognition consisting of strong and weak objects

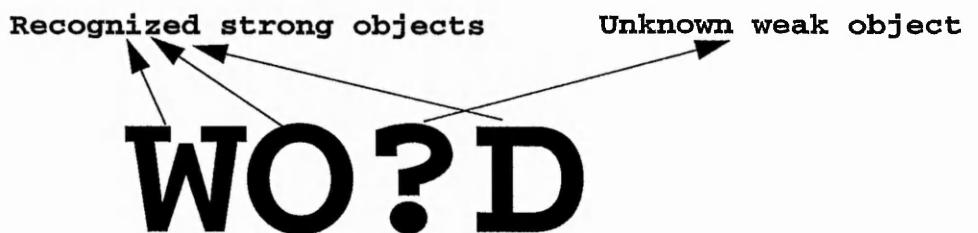


Figure 7.2: Recognized strong objects and unknown weak objects

“WORD”, “WORLD”, etc.). The letter/letters contained at the position ‘?’ (i.e O, R, RL) of the word are the letter alternative for the weak object. Hence we have used strong objects to recognize the weak objects in general and hence recognized the whole word correctly using word postulation approach.

7.6 Ideas for dealing with touching objects

7.6.1 Introduction

One of the main problems in coping with poor quality documents is the handling of touching letters. Such documents contain a large number of touching letters. Most OCR systems rely on segmenting these touching letters into single letters. Correct segmentation is very important for getting accurate recognition results.

Bearing this in mind, the following methods for dealing with touching objects are proposed.

7.6.2 Extracting features from upper half of the objects

Underlining often results in whole word or line touching, specially when words contain letters with descenders (e.g. underline causes touching). In order to recognize such words either preprocessing needs to be done to remove these lines and then recognized or it may be recognized as a whole word. Removing lines may result in broken characters. In order to avoid this and successful recognition of underlined words, it is suggested to extract features only from the top half of these words. Although this may result in the extraction of less information about a word, the problems of getting broken letters may be avoided. Another approach to cope with the recognition of underlined words can be developing a separate database for underlined words. This database can then be used for the recognition of underlined words. The later approach however needs identification of underlined words prior to their recognition and also it needs a separate database. Future work may involve the implementation of these two proposed methods for the recognition of underlined words.

7.6.3 Extracting features from upper and lower halves separately

In the present research, different features are extracted from the whole object and are used for its recognition. Future work may also include dividing the word into two halves. Features from the upper and lower halves of the words may be extracted separately. Two different recognition results may be obtained using upper and lower half features. These results may then be compared to identify the correct solution.

7.6.4 Segmenting touching objects based on features

The developed recognizer produces a list of alternatives for each sample word, which is substantially smaller as compared to the dictionary. Also this list often contains correct candidates, which may be at lower rank. In order to select the correct alternative from the list, the touching objects in the sample word can be segmented based on found features and recognition attempted by considering the list produced by the recognizer. In this way segmentation can stand a better chance and the correct alternative can be found easily from the small list as compared to the whole dictionary. The developed recognizer can therefore be used as a preprocessor to produce a small list of desired words.

It has been observed from poor quality documents with many touching characters that top side open features in touching characters can be considered as segmentation points in order to segment and separate single letters from touching letters. Many touching characters are seen to touch at the bottom or in the middle. Therefore these touching characters make a top side open feature after touching together. This top side open feature may mean that the character is not a single character, rather it is a combination of two or more letters. However, there are a few single letters in English writing which have the top side open feature.

Examples of such letters are U, u, V, v, W, w. A check for the presence of such letters in a word can be done. If present, then segmentation is not necessary.

Therefore, if we find all top side open features in touching characters, then the number of top side open may be considered as number of letters touching together. Hence top side open features may be considered as segmentation points for segmenting touching letters into single letters.

Limitations of the proposed Method

At present there is no single method with 100% accuracy for dealing with touching letters. Therefore, like any other method for character recognition problem, this method has also some limitations. These limitations are explained below:

1. The proposed method looks for top side open features in touching characters in order to segment them. Therefore, if we find an object which has got touching letters in it, but has no top side open feature, it will remain unsegmented and will be considered a single letter. Of course, such touching letters occur frequently in practice. Examples of such touching objects are shown in Figure 7.3.



Figure 7.3: Touching characters with no top side open feature

2. There are some single characters which have got top side open features like u, v and w etc. If we directly apply the proposed method to such letters, then these single letters will be segmented further, which is misleading and will give wrong recognition results, as for example 'u' may be segmented into two 'i's. In order to

solve this problem, a pre-check will be made. If a letter has got a top side open feature only, and no other features, then the object will be considered as a single letter and will be left unsegmented.

3. Another problem in using this method is touching characters containing letters such as U, u and V etc. If we get such a touching object, then according to the proposed segmentation method, 'u' will also be segmented from the middle, as it has got a top side open feature. Therefore 'u' will be segmented into two 'i's or 'l's. Hence the proposed segmentation method will give wrong segmentation points for the objects containing such single letters. Examples of such objects can be seen Figure 7.4.



Figure 7.4: Objects containing the letters 'U' or 'u'

7.7 Postprocessing

In advanced OCR systems, the performance of a recognition system that consists only of a single-character recognition unit is not sufficient. It is necessary to use contextual information. The application of context makes it possible to detect errors and even to correct them and hence improve the performance.

In the area of OCR the potential aid of contextual information such as language syntax and semantics appears promising. Therefore, future work may include the integration of the recognizer with higher level linguistics analyzers [Evelt et al 91],

[Evelt et al 93] [Jobbins et al 96]. In this way recognition rate can further be improved.

7.8 Summary

Future work described in this chapter has been divided into six different areas. These are line and word extraction, feature extraction, methods for dealing with touching objects, document and context layout analysis, integration of a postulation algorithm into the developed recognizer, and postprocessing. Different ideas for dealing with touching objects are mentioned, for example, extracting features from the upper half of the word in order to recognize underline words, recognizing poor quality words based on upper and lower half features separately and then comparing the results obtained, and lastly segmenting the touching objects based on features. Implementation of this work will hopefully improve the performance of the developed recognizer in particular and achieve the aim of the OCR research in general.

References

[Adachi et al 88]

Adachi, F., Kawanishi, H. and Kobayashi, T.: (1988) "Improvement of character recognition and generation techniques in facsimile communication systems", Proceedings - IEEE INFOCOM, pp. 732-738.

[Al-Badr and Haralick 94]

Al-Badr, B. and Haralick, R. M.: (1994) "Symbol recognition without prior segmentation", SPIE, Vol. 2181, Document Recognition, pp. 303-314.

[Al-Badr and Haralick 95]

Al-Badr, B. and Haralick, R. M.: (1995) "Segmentation-free word recognition with application to Arabic", Third International Conference on Document Analysis and Recognition, Vol. 1, pp. 355-359.

[Alcorn and Hoggar 69]

Alcorn, T. M. and Hoggar, C. W.: (1969) "Pre-processing of data for character recognition", Marconi Rev. Vol. 32, pp. 61-81.

[Al-Yousefi and Udpa 92]

Al-Yousefi, H. and Udpa, S. S.: (1992) "Recognition of Arabic characters", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 14, No. 8, pp. 853-857.

[Amiri et al 94]

Amiri, A., Downton, A. C., Hanlon, S. J., Leedham, C. G., Lucas, S. M. and Monger, D.: (1994) "OSCAR: A visual programming toolkit for offline handwritten forms recognition", Proc. 4th Int. Workshop on Frontiers in handwriting recognition, Taiwan, pp. 441-448.

[Anderson 69]

Anderson, P. L.: (1969) "Optical character recognition-a survey", Datamation, pp. 43-48.

[Anderson 71]

Anderson, P. L.: (1971) "OCR enters the practical stage", Datamation, pp. 22-27.

[Arakawa et al 78]

Arakawa, H., Okada, K. and Masuda, J.: (1978) "On-line recognition of handwritten characters-Alphanumerics, Hiragana, Katakana, Kanji", Proc. 4th Int. Joint Conf. on Pattern Recognition, pp. 810-812.

[Arcelli 81]

Arcelli, C.: (1981) "Pattern thinning by contour tracing", Comput. Graph. Image Process, Vol. 17, pp. 130-144.

[Arcelli and di Babi 85]

Arcelli, C. and di Babi, G. S.: (1985) "A width-independent fast thinning algorithm", IEEE Trans Pattern Anal. Mach. Intell. Vol. 7, pp. 463-474.

[Badoux 85]

Badoux, R. D.: (1985) "DELTA [text reader for the blind], Computerised braille production", Proc. 5th Int. Workshop, Winterthur, Switzerland, pp. 21-25.

[Baird 88]

Baird, H. S.: (1988) "Feature identification for hybrid structural/statistical pattern classification", Computer Vision, Graphics and Image Processing, Vol. 42, No. 3, pp. 318-333.

[Baird 90]

Baird, H. S.: (1990) "Document image defect models", Proc. IAPR Workshop on Syntactic and Structural Pattern Recog., Murray Hill, N. J.

[Baird 92]

Baird, H. S.: (1992) "Document image defect models", in Structured Document Image Analysis (ed H Baird, H Bunke and K Yamamoto) Springer-Verlag.

[Baird 93]

Baird, H. S.: (1993) "Document image defect models and their uses", In Proceedings of the Second International Conference on Document Analysis and Recognition ICDAR-93, pp. 62-67.

[Baird and Nagy 94]

Baird, H. S. and Nagy, G.: (1994) "A self-correcting 100-font classifier", In Proceedings of the Conference on Document Recognition of 1994 S&T/ SPIE Symposium.

[Baykal et al 91]

Baykal, N., Yalabik, N. and Goktogan, A. H.: (1991) "Character recognition using Kohonen's features map", Computer and Information Sciences VI, pp. 923-932.

[Baykal and Yalabik 92]

Baykal, N. and Yalabik, N.: (1992) "Object orientation detection and character recognition using optimal feedforward network and Kohonen's features map", SPIE, Applications of Artificial Neural Networks III, Vol. 1709, pp. 292-303.

[Berger et al 85]

Berger, A., Dunbar, P. and Robert, C.: (1985) "Machine vision recognition in the electronic packaging industry, three case studies", VISION 85 Conf. Proc., Detroit, MI, U.S.A.

[Bernstein 68]

Bernstein, M. I.: (1968) "A method for recognizing handprinted characters in real-time", Pattern Recognition, Ed. L. A. Kanal, Thompson, pp. 109-114.

[Bernsen 86]

Bernsen, J.: (1986) "Dynamic thresholding of grey-level images", Proc. Int. Conf. on Pattern Recognition, pp. 1251-1255.

[Berthod 78]

Berthod, M.: (1978) "Experimentations sur l'échantillonnage de traces manuscrites en temps réel", Congrès AFCET-INRIA, Traitement des Images et Reconnaissance des Formes, Gif sur Yvette, Fevrier.

[Bertille and Gilloux 95]

Bertille, J-M. and Gilloux, M.: (1995) "A probabilistic approach to automatic handwritten address reading", Third International Conference on Document Analysis and Recognition, Vol. 1, pp. 368-371.

[Beun 73]

Beun, M.: (1973) "A flexible method for automatic reading of handwritten numerals", Philips, Tech. Rev., Vol. 33, pp. 89-101, 130-137.

[Bhate et al 95]

Bhate, A., Lam, S. W. and Srihari, S. N.: (1995) "A sliding window technique for word recognition", SPIE, Vol. 2422, pp. 38-46.

[Bliss 69]

Bliss, J. C.: (1969) "A relatively high-resolution reading aid for the blind", IEEE T. Man, Mach. System, Vol. 10, pp. 1-9.

[Boccignone et al 93]

Boccignone, G., Chianese, A. and Cordella, L. P.: (1993) "Recovering dynamic information from static handwriting", Pattern Recognition, Vol. 26, No. 3, pp. 409-418.

[Bojman 70]

Bojman, W.: (1970) "Detection and/or measurement on complex patterns", IBM Technical Disclosure Bull. (U.S.A.), Vol. 13, pp. 1429-1430.

[Bolton and Bayle 77]

Bolton, R. and Bayle, A. R.: (1977) "Interactive digitization of sounding values on charts", in Proc. 5th Man-Computer Communications Conf., pp. 53-62.

[Bokser 92]

Bokser, M.: (1992) "Omnidocument technologies", Proceedings of the IEEE, 80(7), pp. 1066-1078.

[Bose and Kuo 92]

Bose, C. B. and Kuo, S.: (1992) "Connected and degraded text recognition using hidden Markov model", Pattern Recognition Methodology and Systems, pp. 116-119.

[Brown 64]

Brown, R. M.: (1964) "On-line computer recognition of handprinted characters", IEEE Trans. Electron. Comput., Vol. 13, pp. 750-752.

[Bruyne 86]

Bruyne, P. de.: (1986) "Compact large-area graphic digitizer for personal computer", IEEE Comput. Graph. Appl., pp. 49-53.

[Buckle and Strand 81]

Buckle, D. and Strand, T. D.: (1981) "Processing of information", United States Patent, 4.262.281.

[Campigli et al 91]

Campigli, P., Cappellini, V., Paoli, C. and Pareschi, M.: (1991) "Omnifont character recognition", Proceedings of the International Conference on Digital Signal Processing, pp. 484-487.

[Casey 95]

Casey, R. G.: (1995) "Character segmentation in document OCR: Progress and hope", In Symposium on Document Analysis and Information Retrieval, pp. 13-40.

[Casey and Nagy 82]

Casey, R. G. and Nagy, G.: (1982) "Recursive segmentation and classification of composite character patterns", Proceedings of the 6th International Conference of Pattern Recognition, Munich, West Germany, pp. 1023-1026.

[Cash and Hatamain 87]

Cash, G. L. and Hatamain, M.: (1987) "Optical character recognition by the method of moments", Computer Vision Graphics and Image Processing, Vol. 39, pp.291-310.

[Carau and Tremblay 81]

Carau, F. P. and Tremblay, M. A.: (1981) "Travelling wave digitizer", United States Patent, 4.225.617.

[Chanda et al 86]

Chanda, B., Chaudhuri, B. B. and Dutta Mayumder, D.: (1986) "Some modified algorithms for greyscale thresholding", Int. Conf. on Pattern Recognition, pp. 984-986.

[Chatterji 86]

Chatterji, B. N.: (1986) "Feature extraction methods for character recognition", IETE Technical Review, Vol. 3, No. 1, pp. 6-22.

[Chen and DeCurtins 93]

Chen, C. H., and DeCurtins, J. L.: (1993) "Word recognition in a segmentation-free approach to OCR", Second International Conference on Document Analysis and Recognition, pp. 573-576.

[Chen and Lui 92]

Chen, C. C. and Lui, Ho. C.:(1992) "A comparison of different learning algorithms for printed character recognition", Second International Conference on Automation, Robotics and Computer Vision, Vol. 1, pp. CV-18.7.1-CV-18.7.5.

[Chen et al 95]

Chen, F. R., Bloomberg, D. S. and Wilcox, L. D.: (1995) "Spotting phrases in lines of imaged text", SPIE, Vol. 2422. pp. 256-269.

[Cheng et al 92]

Cheng, Y. Q., Jiang, R., Wu, Y. G. and Yang, J. Y.: (1992) "Character recognition by algebraic invariant matrix", SPIE, Visual Information Processing, Vol. 1705, pp. 124-131.

[Chou and Kopec 95]

Chou, P. A. and Kopec, G. E.: (1995) "A stochastic attribute grammar model of document production and its use in document image decoding", SPIE, Vol. 2422, pp. 66-73.

[Christ and Schrag 76]

Christ, E. and Schrag, G.: (1976) "New tasks for the mark sheet reader", Data Rep. Vol. 11, pp. 27-31. (In German)

[Cohen et al 94]

Cohen, E., Hull, J. J. and Srihari, S. N.: (1994) "Control structure for interpreting handwritten addresses", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 16, No. 10, pp. 1049-1055.

[Congedo et al 95]

Congedo, G., Dimauro, G., Impedovo, S. and Pirlo, G.: (1995) "A structural method with local refining for handwritten character recognition", Third International Conference on Document Analysis and Recognition, Vol. II, pp. 853-856.

[Cook 92]

Cook, R.: (1992) "Seeing the benefits of OCR and OCV", Managing Automation, pp. 44-47.

[Crawford 72]

Crawford, J. L.: (1972) "Pictorial information disector and analyser system (PIDAS)", IBM Technical Disclosure Bull, Vol. 15, pp. 61-62.

[Dasarathy 91]

Dasarathy, B. V.: (1991) "Nearest Neighbor(NN) norms", NN Pattern Classification, Techniques. IEEE Computer Society Press.

[Dasarathy and Kumar 78]

Dasarathy, B. V. and Kumar, K. P. B.: (1978) "CHITRA: Cognitive handprinted input-trained recursively analyzing system for recognition of alphanumeric characters", International Journal of Computer and Information Sciences, 7, No. 3, pp. 253-182.

[Davies and Plummer 81]

Davies, E. R. and Plummer, A. P.: (1981) "Thinning algorithm: Critique and a new methodology", Pattern Recognition, Vol. 14, pp. 53-63.

[DeCurtins and Chen 95]

DeCurtins, J. and Chen, E.: (1995) "Keyword spotting via word shape recognition", SPIE, Vol. 2422, pp. 270-277

[Devoe 67]

Devoe, D. R.: (1967) "Alternatives to handprinting in the manual entry of data", IEEE Trans. Human Factors Electron., Vol. 8, pp. 21-32.

[Dodel and Shinghal 95]

Dodel, J-P. and Shinghal, R.L.: (1995) "Symbolic/Neural recognition of cursive amounts on bank cheques", Third International Conference on Document Analysis and Recognition, Vol. 1, pp. 15-18.

[Doermann and Rosenfeld 92]

Doermann, D. S. and Rosenfeld, A.: (1992) "Temporal clues in handwriting", Proc. 11th IAPR Int. Conf. on Pattern Recog. Vol. II, pp. 317-320.

[Downton et al 91]

Downton, A. C., Tregidgo, R. W. S. and Kabir, E.: (1991) "Recognition and verification of handwritten and hand-printed British postal addresses", Int. Jnl. Pattern Recognition and Artificial Intelligence, Vol. 5, No. 1&2, pp. 265-291.

[Downton et al 95]

Downton, A. C., Hanlon, S. J. and Amiri, A.: (1995) "A visual programming toolkit for offline handwritten forms recognition", Third International Conference on Document Analysis and Recognition, Vol. II, pp. 707-710.

[Duda and Hart 73]

Duda, R. O. and Hart, P. E.: (1973) "Pattern classification and scene analysis", Addison-Wesley, New York.

[Eggiman 74]

Eggimann, W.: (1974) "Electronics in U.S.A.-the computer in the supermarket", *Electroniker* (Switzerland), Vol. 13, pp. 28-29. (In German.)

[Elliman and Lancaster 90]

Elliman, D. G. and Lancaster, I. T.: (1990) "A review of segmentation and contextual analysis techniques for text recognition", *Pattern Recognition*, 23, No. 3/4, pp. 337-346.

[ERA 57]

ERA: (1957) "An electronic reading automaton", *Electronic Eng.*, pp. 189-190.

[Evelt et al 91]

Evelt, L. J., Keenan, F. G., Rose, T., Wells, C. J. and Whitrow, R. J.: (1991) "The use of linguistic information to aid script recognition", *Second International Workshop on Frontiers in Handwriting Recognition*, Chateau de Bonas, France, pp. 303-311.

[Evelt et al 93]

Evelt, L. J., Bellaby, G. A., Wells, C. J., Keenan, F. G., Rose, T. and Whitrow, R. J.: (1993) "The use of linguistic information in script recognition", *Handwritten Character and Script Recognition One Day Technical Meeting*, London, Joint Meeting of BMVA and IEE.

[Fang and Hull 95]

Fang, C. and Hull, J. J.: (1995) "A modified character-level deciphering algorithm for OCR in degraded documents", SPIE, Vol. 2422, pp. 76-83.

[Favata et al 94]

Favata, J. T., Srikantan, G. and Srihari, S. N.: (1994) "Handprinted character/digit recognition using a multiple feature/resolution philosophy", The Fourth International Workshop on Frontiers in handwriting recognition, pp. 57-66.

[Fletcher and Kasturi 88]

Fletcher, L. A. and Kasturi, R.: (1988) "A robust algorithm for text string separation from mixed text/graphics images", IEEE Trans. Pattern Anal. Mach. Intell., Vol. 10, No. 6, pp. 910-918.

[Fisher et al 62]

Fischer, G. L., Jr., Pollock, D. J., Raddack, B. and Stevens, M. E.: (1962) "Optical character recognition", Washington, DC: Spartan.

[Fisher et al 90]

Fisher, J. L., Hinds, S. and D'Amato, D. P.: (1990) "A rule-based system for document image segmentation", Proceedings of 10-th International Conference of Pattern Recognition, Atlantic City, pp. 567-572.

[Focht and Burger 76]

Focht, L. R and Burger, A.: (1976) "A numeric script recognition processor for postal zip code application", in Proc. Int. Conf. Cybernetics and Society, pp. 489-492.

[Freedman 74]

Freedman, M. D.: (1974) "Optical character recognition", IEEE Spectrum, pp. 44-52.

[Fu 82]

Fu, K. S.: (1982) "Syntactic pattern recognition and applications", Prentice-Hall, Engelwood Cliffs, N.J.

[Fukushima and Wake 91]

Fukushima, K. and Wake, N.: (1991) "Handwritten alphanumeric character recognition by the neocognitron", IEEE Trans. on Neural Networks, Vol. 2, No. 3, pp. 355-365.

[Fukushima et al 91]

Fukushima, K., Imagawa, T. and Ashida, E.: (1991) "Character recognition with selective attention", International Joint Conference on Neural Networks, pp. I-593-I-598.

[Gaillat and Berthod 79]

Gaillat, G. and Berthod, M.: (1979) "Panorama des techniques d'extraction de traits caracteristiques en lecture optiques des characters", Proc. Developments Recents en Reconnaissance des Forms, pp. 9.1-9.20.

[Garris et al 91]

Garris, M. D., Wilkinson, R. A. and Wilson, C. L.: (1991) "Analysis of a biologically motivated neural network for character recognition", Analysis of Neural Network Applications, pp. 160-175.

[Genchi et al 68]

Genchi, H., Mori, K. I., Watanabe, S. and Katsuragi, S.: (1968) "Recognition of handwritten numeral characters for automatic letter sorting", Proc. IEEE, Vol. 56, pp. 1292-1301.

[Genchi 69]

Genchi, H.: (1969) "Data communication terminal apparatus, optical character and mark reader", Denshi Tsushin Gakkai Zasshi 52, pp. 418-428.
(In Japanese)

[Genchi et al 70]

Genchi, H., Watanabe, S., Matsunaga, S. and Tamada, M.: (1970) "Automatic reader-sorter for mail with handwritten or printed postal code numbers", Toshiba Rev., pp. 7-11.

[Gentric 95]

Gentric, P.: (1995) "Experimental results on improved handwritten word recognition using the Levenshtein metric", Third International Conference on Document Analysis and Recognition, Vol. 1, pp. 364-367.

[Gibson and Talmage 80]

Gibson, W. and Talmage, J.: (1980) "Nonplanar transparent electrographic sensor, United States Patent, 4,220,815.

[Govindan and Shivaprasad 90]

Govindan, V. K. and Shivaprasad, A. P.: (1990) "Character recognition - A review", Pattern Recognition, Vol. 23, No. 7, pp. 671-683.

[Granlund 72]

Granlund, G. H.: (1972) "Fourier preprocessing for hand print character recognition", *IEEE Transactions on Computers*, pp. 195- 201.

[Gray 71]

Gray, P. J.: (1971) "Optical readers and OCR", *Modern Data Jan.*, pp. 66-82.

[Gronmeyer 79]

Gronmeyer, M. L.: (1979) "Recognition of handprinted characters for automated cartography: a progress report, *Proc. Soc. Photo-Opt. Instr. Engng (U.S.A.)*, Vol. 205, pp. 165-174.

[Gudesen 76]

Gudesen, A.: (1976) "Quantative analysis of preprocessing techniques for the recognition of handprinted characters", *Pattern Recognition*, 8, pp. 219-227.

[Guillevic and Suen 95]

Guillevic, D. and Suen, C. Y.: (1995) "Cursive script recognition applied to processing of bank cheques", *Third International Conference on Document Analysis and Recognition*, Vol. 1, pp. 11-14.

[Gyrfas 74]

Gyrfas, A.: (1974) "Experiments concerning the inspection and control of car and truck in France", *Koezlekedes Tud. Sz.*, Vol. 24, pp. 85-91. (In Hungarian)

[Haaley 69]

Haaley, J. D.: (1969) "National giro document reading and sorting optical character recognition", Datafair 1969, Manchester, England.

[Handel 33]

Handel, P.W.: (1933) "Statistical machine", U.S. Patent 1915993.

[Hannan 62]

Hannan, W. J.: (1962) "R. C. A. multifont reading machine", in Optical Character Recognition. G. L. Ficher et al., Eds. McGregor & Wemer, pp. 3-14.

[Haralick 78]

Haralick, R. M.: (1978) "Statistical and Structural approaches for texture", Proc. 4th Int. Joint Conf. on Pattern Recognition, Kyoto, pp. 45-69.

[Harmon 72]

Harmon, L. D.: (1972) "Automatic recognition of print and script", Proc. IEEE, Vol. 60, pp. 1165-1176.

[Harness et al 93]

Harness, S., Pugh, K., Sherkat, N. and Whitrow, R. J.: (1993) "Enabling the use of windows environment by the blind partially sighted", IEE Colloquium, London.

[Hemphill 75]

Hemphill, B. R.: (1975) "Optical character recognition-the future is here", AEDS Monit. (U.S.A.), Vol. 13, pp. 8-9.

[Hendrawan and Downton 94]

Hendrawan and Downton, A. C.: (1994) "Verification of handwritten British postcodes using address features", in 'Fundamentals in handwriting recognition' (ed. S. Impedovo), pp. 313-317.

[Hennig et al 97]

Hennig, A., Raza, G., Sherkat, N. and Whitrow, R. J.: (1997) "Detecting a document's skew: A simple stochastic approach", Eleventh Canadian Conference on Computer Vision, Signal and Image Processing, and Pattern Recognition, pp. 97-102.

[Herst and Liu 80]

Herst, N. M. and Liu, C. N.: (1980) "Card-based personal identification system", IBM Technical Disclosure Bull. (U.S.A.), Vol. 22, pp. 4291-4293.

[Hilgert 70]

Hilgert, G.: (1970) "Method of dealing with orders on the IBM 1287 multifunction document reader at the decentralised sales organization of the continental Gummi-Werke Aktiengesellschaft, IBM Nachr., Vol. 20, pp. 122-125. (In German)

[Ho 92]

Ho, T. K.: (1992) "A theory of multiple classifier systems and its application to visual word recognition", PhD thesis, Computer Science Department of SUNY at Buffalo.

[Ho and Baird 93]

Ho, T. K. and Baird, H. S.: (1993) "Perfect metrics", In Proceedings of the Second International Conference on Document Analysis and Recognition ICDAR-93, pp. 593-597.

[Ho and Barid 94]

Ho, T. K. and Baird, H. S.: (1994) "Asymptotic accuracy of two-class discrimination", In Symposium on Document Analysis and Information Retrievals, pp. 413-422.

[Hodges 83]

Hodges, A.: (1983) "Alan Turing: The Enigma", Simon & Schuster, New York.

[Ho et al 92]

Ho, T. K., Hull, J. J. and Srihari, S. N.: (1992) "A word shape analysis approach to lexicon based word recognition", in Pattern Recognition Letters, Vol. 13, pp. 821-826.

[Hong and Hull 94]

Hong, T. and Hull, J. J.: (1994) "Degraded text recognition using word collocation", SPIE, Vol. 2181, pp. 334-341.

[Hong and Hull 95]

Hong, T. and Hull, J. J.: (1995) "Character segmentation using visual inter-word constraints in a text page", SPIE, Vol. 2242, pp. 15-25.

[Horowitz and Pavlidis 74]

Horowitz, S. L. and Pavlidis, T.: (1974) "Picture segmentation by a directed split-and-merge procedure", Proc. 2nd Int. Joint Conf. on Pattern Recognition, Copenhagen, pp. 424-433.

[Houle and Eom 91]

Houle, G. and Eom K-B.: (1991) "On the use of a priori knowledge to character recognition", IEEE International Joint Conference on Neural Networks, pp. 1415-1420.

[Hsieh and Lee 92]

Hsieh, C. C. and Lee, H. J.: (1992) "Off-line recognition of handwritten Chinese characters by on-line model-guided matching:", Pattern Recognition, 25, No. 11, pp. 1337-1352.

[Hull et al 92]

Hull, J. J., Khoubyari, S. and Ho, T. K.: (1992) "Word image matching as a technique for degraded text recognition", Pattern Recognition Methodology and Systems, pp. 665-668.

[Impedovo et al 91]

Impedovo, S., Ottaviano, L. and Occhinegro, S.: (1991) "Optical character recognition -A survey", International Journal of pattern Recognition and Artificial Intelligence, Vol. 5, No. 1 & 2, pp. 1-24.

[Inoue et al 73]

Inoue, S., Kurematsu, A., Wada, T. and Nakabo, S.: (1973) "Studies on optical character recognition of international telegraph, KDD Tech. J., Vol. 77, pp. 51-61. (In Japanese.)

[Jagota 90]

Jagota, A.: (1990) "Applying a Hopfield-style network to degraded text recognition", International Joint Conference on Neural Networks, Vol. 1, pp. 27-32.

[Jobbins et al 96]

Jobbins, A. C., Raza, G., Evett, L. J. and Sherkat, N.: (1996) "Postprocessing for OCR: Correcting errors using semantic relations", in L. J. Evett and T. G. Rose (Eds.) Language Engineering for Document Analysis and Recognition (LEDAR), AISB96 Workshop, Sussex, England, pp. 154-161.

[Joe and Lee 95]

Joe, M. J. and Lee, H. J.: (1995) "A combined method on the handwritten character recognition", Third International Conference on Document Analysis and Recognition, Vol. I, pp. 112- 115.

[Joshi 74]

Joshi, C. P.: (1974) "Role of electronics in law enforcement", I. Inst. Elec. and Telecom. Engng (India), Vol. 20, pp. 500-503.

[Kahan et al 87]

Kahan, S., Pavlidis, T. and Baird, H. S.: (1987) "On the recognition of printed characters of any font and size", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 9, pp. 274-288.

[Kalberg et al 93]

Kalberg, R. J. N., de Jong, J. B. and Essink, H. P. M.: (1993) "Automatic reading of handwritten postcodes in boxes", First European Conference on Postal Technology JET POSTE 93, pp. 20-25.

[Kam and Kopec 95]

Kam, A. C. and Kopec, G. E.: (1995) "Separable source models for document image decoding", SPIE, Vol. 2422, pp. 84-97.

[Kanai 90]

Kanai, J.: (1990) "Text line extraction using character prototypes", Proceedings of Syntactic and Structural Pattern Recognition, Murray Hill, NJ, pp. 182-191.

[Kertesz and Kertesz 92]

Kertesz, A. and Kertesz, V.: (1992) "Dynamically connected neural network for character recognition", International Joint Conference on Neural Networks, Vol. 3, pp. 672-676.

[Kondraske and Shennib 86]

Kondraske, G. V. and Shennib, A.: (1986) "Character pattern recognition for a telecommunication aid for the deaf", IEEE T. Biomed. Engng., Vol. 33, pp. 366-370.

[Kopec et al 95]

Kopec, G. E., Chou, P. A. and Maltz, D. A.: (1995) "A Markov source model for printed music decoding", SPIE, Vol. 2422, pp. 115-125.

[Kovalevsky 68]

Kovalevsky, V. A.: (1968) "Character readers and pattern recognition",
New York: Spartan Books.

[Kroger 87]

Kroger, S.: (1987) "Scanner in practice, Chip (Germany), Vol. 5, pp. 94-96.
(In German)

[Kucera and Francis 67]

Kucera, H. and Francis, W. N.: (1967) Computational Analysis of Present-day American English, Brown University Press.

[Kupriyanov 72]

Kupriyanov, S.: (1972) "Electronic handwriting analyzer", Tekh. Misul
(Bulgaria), Vol. 9, pp. 7-13. (In Bulgarian)

[Kwan et al 79]

Kwan, C. C., Pang, L. and Suen, C. Y.: (1979) "A comparative study of
some recognition algorithms in character recognition", Proceedings -
International Conference on Cybernetics and Society, pp. 530-535.

[Lam et al 95]

Lam, S. W., Bhate, A. and Srihari, S. N.: (1995) "A sliding window
technique for word recognition", SPIE, Vol. 2422, pp.38-46.

[Lai and Suen 81]

Lai, M. T. Y. and Suen, C. Y.: (1981) "Automatic recognition of characters
by Fourier descriptors and boundary line encoding", Pattern Recognition,
Vol. 14, pp. 383-393.

[Lecolinet and Moreau 90]

Lecolinet, E., Moreau, J. V.: (1990) "Off-line recognition of handwritten cursive script for the automatic reading of city names on real mail", Proceedings - International Conference on Pattern Recognition, Vol. 1, pp. 674-676.

[Lee et al 93]

Lee, S-W, Park, J-S and Tang, Y. Y.: (1993) "Performance evaluation of nonlinear shape normalization methods for the recognition of large-set handwritten characters", Second International Conference on Document Analysis and Recognition, pp. 402-407.

[Leedham and Downton 86]

Leedham, C. G. and Downton, A. C.: (1986) "On-line recognition of Pitman's handwritten shorthand-an evaluation of potential", Int. J. Man Mach. Stud. (U.K.), Vol. 24, pp. 375-393.

[Leedham and Downton 87]

Leedham, C. G. and Downton, A. C.: (1987) "Automatic recognition and transcription of Pitman's handwriting shorthand-an approach to short forms", Pattern Recognition, Vol. 20, pp. 341-348.

[Leroux and Salome 90]

Leroux, M. and Salome, J. C.: (1990) "Recognition of cursive writing, A method of segmentation of words into letters", ICARV, pp. 1211-1215.

[Lethelier et al 95]

Lethelier, E., Leroux, M. and Gilloux, M.: (1995) "An automatic reading system for handwritten numeral amounts on French cheques", Third

International Conference on Document Analysis and Recognition, Vol. 1, pp. 92-97.

[Liang et al 93]

Liang, S. Ahmadi, M. and Shridhar, M.: (1993) "Segmentation of touching characters in printed document recognition", In Proceedings of Second International Conference on Document Analysis and Recognition ICDAR-93, pp. 569-572.

[Lijima et al 63]

Lijima, T., Okumura, Y., and Kuwabara, K.: (1963) "New process of character recognition using sieving method", Information and Control Research", Vol. 1, No. 1, pp. 30-35.

[Lindgren 65]

Lindgren, B.: (1965) "Machine recognition of human language, Part III-Cursive script recognition", IEEE Spectrum, pp. 104-116.

[Lippman 87]

Lippman, P. P.: (1987) "An introduction to computing with neural nets", IEEE ASSP Magazine, 4, pp. 4-22.

[Lui et al 90]

Lui, H. C., Lee, C. M. and Gao, F.: (1990) "Neural network application to container number recognition", IEEE 11th Annual International Computer Software & Application Conference", Chicaco, IL, pp. 190-194.

[Lu 93]

Lu, Yi: (1993) "On the segmentation of touching characters", Second International Conference on Document Analysis and Recognition, pp. 440-443.

[Lukis and Duhing 85]

Lukis, L. J. and Duhing, G. P.: (1985) "Character recognition device", United States Patent, 4.493.104.

[Lybanon and Gronmeyer 78]

Lybanon, M. and Gronmeyer, L. K.: (1978) "Recognition of handprinted characters for automated cartography", Proc. Photo-Optical Instrum. Eng., Vol. 155, Image Understanding & Industrial Applications, pp. 56-65.

[Mantas 86]

Mantas, J.: (1986) "Methodologies in pattern recognition and image analysis-A brief survey", Pattern Recog., Vol. 20, No. 1, pp. 1-6.

[Mantas 87]

Mantas, J.: (1987) "Methodologies in pattern recognition and image analysis-A brief survey", Pattern Recog., Vol. 20, No. 1, pp. 1-6.

[Matsui et al 93]

Matsui, T., Yamashita, I., Wakahara, T. and Yoshimuro, M.: (1993) "State of the art of handwritten numeral recognition in Japan", First European Conference on Postal Technology JET POSTE 93, pp. 3-10.

[McAbee 67]

McAbee, J. C.: (1967) "OCR application at United Air Lines, Data Processing XII", Proc. 1967 Int. Data Process. Conf. and Business Exposition, Boston, MA, U.S.A., pp. 255-260.

[Moret 82]

Moret, B. M. E.: (1982) "Decision trees and diagrams", ACM Computing Surveys, 14, pp. 593-623.

[Mori et al 70]

Mori, K. I., Genchi, H., Watanabe, S. and Katsuragi, S.: (1970) "Micro-program controlled pattern processing in a handwritten mail reader-sorter", Pattern Recognition, Vol. 2, pp. 175-185.

[Mori et al 92]

Mori, S., Suen, C. Y. and Yamamoto, K.: (1992) "Historical review of OCR research and development", Proceedings of the IEEE, Vol. 80, No. 7, pp. 1029-1058.

[Mostert and Brand 92]

Mostert, S and Brand, J. A.: (1992) "Omni-font character recognition using templates and neural networks", Proceedings of the South African Symposium on Communications and Signal Processing, pp. 249-257.

[Moukrim and Muller 91]

Moukrim, A. and Muller, C.: (1991) "NN and heuristic approach to character recognition", Artificial Neural Networks, Elsevier Science Publishers B. V. (North-Holland), pp. 1099-1102.

[Munson 68]

Munson, J. H.: (1968) "Experiments in the recognition of hand-printed text: Part I-Character recognition", in Proc. Fall Joint Computer Conf. Vol. 33, pp. 1125-1138.

[Murphy and Stohr 75]

Murphy, F. H. and Stohr, E. A.: (1975) "Optimal check sorting strategies", Bull, oper. Res. Am., Vol. 23 (Supplement 1), B145.

[Naccache and Shinghal 84a]

Naccache, N. J. and Shinghal, R.: (1984) "An investigation into the skeletonization approach of Hilditch", Pattern Recognition, Vol. 17, pp. 279-284.

[Naccache and Shinghal 84b]

Naccache, N. J. and Shinghal, R.: (1984) "STPA: A proposed algorithm for thinning binary patterns", IEEE Trans. Syst. Man Cybern, vol. 14, pp. 409-418.

[Nagy and Tuong 70]

Nagy, G. and Tuong, M.: (1970) "Normalization techniques for handprinted numerals", Commun. ACM, Vol. 13, pp. 475-481.

[Nagy 86]

Nagy, G.: (1986) "Efficient algorithms to decode substitution ciphers with application to OCR", Proceedings of 8th International Conference of Pattern Recognition, Paris, France, pp. 352-355.

[Nakamura et al 86]

Nakamura, Y., Suda, M., Sakai, K., Takeda, Y. and Udaka, M.: (1986)
“Development of an high performance stamped character reader”, IEEE T.
Ind. Electron. (U.S.A.), Vol. 33, pp. 144-147.

[Nakamura et al 87]

Nakamura, Y., Suda, M., Hayashi, T., Tanaka, A. and Watanabe, S.: (1987)
“An optical character recognition system for industrial application:
TOSEYE-1000, Proc. Int. Workshop on Industrial Application of Machine
Vision and Machine Intelligence, Seiken Symp., Tokyo, Japan, pp. 364-
368.

[Nassimbene 72]

Nassimbene, E. G.: (1972) “Digital compare circuitry”, IBM Technical
Disclosure Bull, Vol. 14, pp. 3421-3422.

[Neill 69]

Neill, J.: (1969) “Numeric script mail sorter”, Proc. Automat. Pattern
Recognition, pp. 49-65.

[Neisser and Weene 60]

Niesser, U. and Weene, P.: (1960) “A note on human recognition of
handprinted characters”, Inf. Control, 3, pp. 191-196.

[Nevatia 86]

Nevatia, R.: (1986) “Image segmentation”, Handbook of Pattern
Recognition and Image Processing, Academic Press, Inc., pp. 215-231.

[Notbohm and Hanisch 86]

Notbohm, K. and Hanisch, W.: (1986), "Automatic digit recognition in a mail sorting machine", Nachrichtentech, Electron, Germany, Vol. 36, pp. 472-476. (In German)

[O’Gorman and Clowes 76]

O’Gorman, F. and Clowes, M. B.: (1976) "Finding picture edge through collinearity of feature points", IEEE Trans. Comput., Vol. 25, pp. 449-456.

[Pavlidis 80]

Pavlidis, T.: (1980) "Thinning algorithm for discrete binary images", Comput. Graph. Image Process, Vol. 13, pp. 142-157.

[Pavlidis 82]

Pavlidis, T.: (1982) "An asynchronous thinning algorithm", Comput. Graph. Image Process. Vol. 20, pp. 133-157.

[Pepper 78]

Pepper, W.: (1978) "Human-machine interface apparatus", United States Patent, 4.071.691.

[Perret 80]

Perret, U.: (1980) "Computer assisted forensic linguistic system 'TEXTOR'", Proc. 3rd Int. Conf. Security through Science Engng, Lexington, KY, U.S.A., pp. 139-149.

[Pintsov 93]

Pintsov, L. A.: (1993) "Handwritten character recognition, some observations concerning principles and modus operandi", First European Conference on Postal Technology JET POSTE 93, PP. 26-34.

[Pokluda 77]

Pokluda, M.: (1977) "Optical pattern recognition in NHKG Ostrava", Mech. Autom. Adm. (Czechoslovakian), Vol. 19, pp. 218-220. (In Czechoslovakian)

[Poliakoff et al 95]

Poliakoff, J. F., Thomas, P. D., Razzaq, S. M. and Shaw, N. G.: (1995) "3-D reconstruction for correction of errors and imperfections in scanned engineering drawings", Combined Proceedings of (EDUGRAPHICS '95) and (COMPUGRAPHICS '95), pp. 98-107.

[Polizzano 83]

Polizzano, P. F.: (1983) "OCR and electronic mail", Computer World, Vol. 17, pp. 49-52.

[Powalka 95]

Powalka, R. K.: (1995) "An algorithm toolbox for on-line cursive script recognition", PhD Thesis, The Nottingham Trent University.

[Prather 70]

Prather, R. C.: (1970) "Handwritten character recognition and related topics", Computing Rev., pp. 291-302.

[Prugh and Fadden 80]

Prugh, R. W. and Fadden, B. J.: (1980) "Graphic digitizer", United States Patent, 4.206.314.

[Rabinow 69]

Rabinow, J. C.: (1969) "Whither OCR and whence?", Datamation, pp. 38-42.

[Raza et al 96a]

Raza, G., Sherkat, N. and Whitrow, R. J.: (1996) "Word recognition using multiple independent features", International Conference on Natural Language Processing and Industrial Applications, Vol. II, (1996), pp. 233-236.

[Raza et al 96b]

Raza, G., Sherkat, N. and Whitrow, R. J.: (1996) "Recognition of poor quality words without segmentation", International Conference on Systems, Man and Cybernetics, Vol. 1, pp. 64-69.

[Raza et al 97a]

Raza, G., Hennig, A., Sherkat, N. and Whitrow, R. J.: (1997) "Applying feature based word recognition approach to screen text recognition", Eleventh Canadian Conference on Computer Vision, Signal and Image Processing, and Pattern Recognition, pp. 103-107.

[Raza et al 97b]

Raza, G., Hennig, A., Sherkat, N. and Whitrow, R. J.: (1997) "Recognition of facsimile messages using database of robust features" to be published (International Conference for Document Analysis and Recognition).

[Ress 75]

Ress, Z.: (1975) "Some experience with optically readable handwriting in solving the MIKROCENSUS 73", Mech, Autom. Adm. (Czechoslovakia), Vol. 15, pp. 131-138. (In Czechoslovakian.)

[Ricker and Winkler 94]

Ricker, G. and Winkler, A.: (1994) "Recognition of faxed documents", SPIE, Vol. 2181, Document Recognition, pp. 371-377.

[Rocha and Pavlidis 95]

Rocha, J. and Pavlidis, T.: (1995) "Character recognition without segmentation", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 17, No. 9, pp. 903-909.

[Rosenfeld 87]

Rosenfeld, A.: (1987) "Picture processing: 1986", in Comput. Vision Graph. Image Process, Vol. 38, pp. 147-225.

[Rosenfeld and Davis 76]

Rosenfeld, A. and Davis, L. S.: (1976) "A note of thinning", IEEE Trans. Syst. Man Cybern. Vol. 25, pp. 226-228.

[Rosenfeld and Thurston 71]

Rosenfeld, A. and Thurston, M.: (1971) "Edge and curve detection for visual scene analysis", IEEE Trans. Comput., Vol. 20, pp. 562-569.

[Romein 81]

Romein, J. J.: (1981) "Acoustic writing combination, comprising a stylus with an associated writing tablet", United States Patent, 4.246.439.

[Rumelhart and McClelland 82]

Rumelhart, D. E. and McClelland, J. L.: (1982) "An interactive activation model of context effects in letter perception", Psychological Review, 89(1), pp. 60-94.

[Sabourine and Mitiche 92]

Sabourin, M. and Mitiche, A.: (1992) "Optical character recognition by a Neural Network", *Neural Networks*, Vol. 5, pp. 843-852.

[Schacht 78]

Schacht, D.: (1978) "Control of outside workers in sales and distributions using the optical document reader IBM 3886", *IBM Nachr. (Germany)*, Vol. 28, pp. 131-138. (In German.)

[Schafer 73]

Schafer, H.: (1973) "Mechanized document reading in a textile and clothing manufacture enterprise, *IBM Nachr.*, Vol. 23, pp. 776-782. (In German.)

[Schmitt 90]

Schmitt, L.: (1990) "Neural networks for OCR", *Photonics Spectra*, pp. 114-115.

[Seni et al 95]

Seni, G., Kripasundar, V. and Srihari, R. K.: (1995) "Generalized edit distance for handwritten text recognition", *SPIE*, Vol. 2422, pp.54-65.

[Shaw 94]

Shaw, N. G.: (1994) "3D reconstruction and correction of objects described by engineering drawings", PhD Thesis, The Nottingham Trent University.

[Shchepin and Nepomnyashchii 91]

Shchepin, E. V. and Nepomnyashchii, G. M.: (1991) "Character recognition via critical points", *International Journal on Imaging Systems and Technology*, Vol. 3, pp. 213-221.

[Sherkat et al 93]

Sherkat, N., Whitrow, R. J., Pugh, K. and Harness, S.: (1993) "Fast icon and character recognition for providing universal access to a WIMP environment for the blind", *Studies in Health Technology and Informatics*, IOS Press, Amsterdam, pp. 19-23.

[Shillman et al 74]

Shillman, Cox, C., Kuklinski, T., Ventura, J., Eden, M. and Blesser, B.: (1974) "A bibliography in character recognition: techniques for describing characters", *Visible Language*, Vol. 8, pp. 151-166.

[Skalski 67]

Skalski G. L.: (1967) "OCR in the publishing industry", *Data Processing XII, Proc. 1967 Int. Data Process. Conf. and Business Exposition*, Boston, MA, U.S.A., pp. 255-260.

[Smitch 73]

Smitch, G. C.: (1973) "The stereotoner reading aid for the blind, a progress report", *1973 Carnahan Conf. on Electronic Prosthetics*, Lexington, U.S.A., pp. 74-76.

[Smith and Merali 85]

Smith, J. W. T. and Merali, Z.: (1985) "Optical character recognition: the technology and its applications in information units and libraries Report 33", *British Library*, Boston, Spa, Wetherby, West Yorks, England.

[Spitz 93]

Spitz, A. L.: (1993) "Generalized line, word and character finding", Proceedings of the Seventh International Conference on Image Analysis and Processing, Bari, Italy.

[Spronsen and Bruggeman 85]

Spronsen, C. J. V. and Bruggeman, F.: (1985) "Raised type reading, Mini and Microcomputers and their applications", Proc. ISMM Int. Symp., Sant Feliu de Guixols, Spain, pp. 274-277.

[Srihari 92]

Srihari, S. H.: (1992) "High-performance reading machines", Proceedings of the IEEE, 80(7), pp. 1120-132.

[Srihari 93]

Srihari, S. H.: (1993) "From pixels to paragraphs: the use of models in text recognition", In Symposium on Document Analysis and Information Retrieval, pp. 47-64.

[Stefanelli and Rosenfeld 71]

Stefanelli, R. and Rosenfeld, A.: (1971) "Some parallel thinning algorithms for digital pictures", J. AMC, Vol. 18, pp. 255-264.

[Stentiford and Montimer 83]

Stentiford, F. W. M. and Montimer, R. G.: (1983) "Some new heuristics for thinning binary handprinted characters for OCR", IEEE Trans. Syst. Man Cybern, vol. 13, pp. 81-84.

[Sternberg 75]

Sternberg, J.: (1975) "Automatic signature verification using handwriting pressure", 1975 WOSCON Technical Papers- Western Electronic Show and Convention 19, San Francisco, California, U.S.A., pp. 4-31.

[Stevens 61]

Stevens, M. E.: (1961) "Automatic character recognition-A state-of-the-art report", National Bureau of Standards, Tech. Note 112.

[Stevens 70]

Stevens, M. E.: (1970) "Special issue on optical character recognition", Pattern Recognition, Vol. 2, pp. 145-239.

[Strathy and Suen 95]

Strathy, N. W. and Suen, C. Y.: (1995) "A new system for reading handwritten zip codes", Third International Conference on Document Analysis and Recognition, Vol. 1, pp. 74-77.

[Suen 78]

Suen, C. Y.: (1978) "Advances in optical character recognition", in Proc Canadian Computer Conf., pp.263-268.

[Suen 82]

Suen, C. Y.: (1982) "Distinctive features in automatic recognition of handprinted characters", Signal Process, Vol. 4, pp. 193-207.

[Suen et al 68]

Suen, C. Y., Berthod, M. and Mori, S.: (1968) "Automatic recognition of handprinted characters-the state of the art", Proc. IEEE, Vol. 14, pp. 226-233.

[Suen et al 77]

Suen, C. Y., Shinghal, R. and Kwan, C. C.: (1977) "Dispersion factor: A quantitative measurement of the quality of handprinted characters", Proc. Int. Conf. on Cybernetics and Society, pp. 681-685.

[Suen et al 80]

Suen, C. Y., Berthod, M. and Mori, S.: (1980) "Automatic recognition of handprinted characters", Proc. IEEE, Vol. 68, pp. 469-487. Also Proc. 4th Int. Jt. Conf. Pattern Recognition.

[Suen and Shillman 77]

Suen, C. Y. and Shillman, R. J.: (1977) "Low error rate optical character recognition of unconstrained handprinted letters based on a model of human perception", IEEE Transactions on Systems, Man, and Cybernetics, pp. 491-495.

[Sun and Wee 82]

Sun, C. and Wee, W.: (1982) "Neighboring gray level matrix textural classification", Comput. Vision Graph. Image Process, Vol. 23, pp. 341-352.

[Swonger 69]

Swonger, C. W.: (1969) "An evaluation of character normalization, feature extraction and classification techniques for postal mail reading", Proc. Automatic pattern Recognition, Washington, D.C., U.S.A., pp. 67-87.

[Takahashi et al 90]

Takahashi, H., Itoh, N., Amano, T. and Yamashita, A.: (1990) "A spelling correction method and its application to an OCR system", Pattern Recognition, Vol. 23, No. 3/4, pp. 363-377.

[Tanaka et al 86]

Tanaka, E., Kohashiguchi, T. and Shimamura, K.: (1986) "High speed correction for OCR", Proceedings of International Conference of Pattern Recognition, Paris, France, pp. 340-343.

[Tang and Suen 92]

Tang, Y. Y. and Suen, C. Y.: (1992) "Parallel character recognition based on regional projection transformation (RPT)", 11th IAPR International Conference on Pattern Recognition, Proceedings, Vol. 2, pp. 631-634.

[Tappert et al 88]

Tappert, C. C., Suen, C. Y. and Wakahara, T.: (1988) "On-line handwriting recognition-A survey", Proc. 9th Int. Conf. on Pattern Recognition, Rome, pp. 1123-1132.

[Tappert et al 90]

Tappert, C. C., Suen, C. Y. and Wakahara, T.: (1990) "The state of the art in on-line handwriting recognition", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 12, No. 8, pp. 787-808.

[Tauschek 35]

Tauschek, G.: (1935) "Reading machine", U.S. Patent 2026329.

[Thomas et al 95]

Thomas, P. D., Poliakoff, J. F., Razzaq, S. M. and Whitrow, R. J.: (1995) "Interpreting scanned engineering drawing - A high level approach", Proceedings of the International Workshop on Graphics Recognition, IAPR, pp. 179-188.

[Throssell and Fryer 74]

Throssell, W. R. and Fryer, P. R.: (1974) "The measurement of print quality for optical character recognition system", Pattern recognition, Vol. 6, pp. 141-147.

[Tian et al 91]

Tian, Q., Zhang, P., Alexander, T. and Kim, Y.: (1991) "Survey: Omnifont printed character recognition", SPIE, Vol. 1606, Visual Communications and Image Processing, pp. 260-268.

[Timm 73]

Timm, H.: (1973) "Registering of health insurance data using the IBM 1288 page reader", IBM Nachr., Vol. 23, pp. 789-792. (In German)

[Togawa et al 91]

Togawa, F., Ueda, T., Aramaki and Tanaka, A.: (1991) "Receptive field neural network with shift tolerant capability for Kanji character recognition", IEEE International Joint Conference on Neural Networks, Vol. 2, pp. 1090-1099.

[Tou and Gonzalez 72]

Tou, J. T. and Gonzalez, R. C.: (1972) "Recognition of handwritten characters by topological feature extraction and multi-level categorization", IEEE Trans. Comput., C-21, pp. 776-785.

[Tsujiimoto and Asada 91]

Tsujiimoto, S. and Asada, H.: (1991) "Resolving ambiguity in segmenting touching characters", Proceedings of the 1st International Conference on Document Analysis and Recognition, pp. 701-709.

[Tsukumo 92]

Tsukumo, J.: (1992) "Handprinted Kanji character recognition based on flexible template matching", IEEE 11th IAPR International Conference on Pattern Recognition, pp. 483-486.

[Turner and Ritchie 70]

Turner, J. A. and Ritchie, G. J.: (1970) "Linear current division in resistive areas, its application to computer graphics", Proc. SJCC, Vol. 36, pp. 613-620.

[Ufer 70]

Ufer, J.: (1970) "Direct data processing with the IBM 1287 multipurpose document reader for standard article-fresh service to Joh. Jacob and Co., Breman, IBM Nachr. (Germany), Vol. 20, pp. 35-40. (In German)

[Umeno and Zhu 87]

Umeno, M. and Zhu, X.: (1987) "Neural Model for character Recognition", Bulletin of Nagoya Institute of Technology, Vol. 39, pp. 231-238.

[Verikas et al 89]

Verikas, A. A., Bachauskene, M. I. and Darshkus, E. R.: (1989) "Comparative study of some character recognition algorithms", The 6th Scandinavian Conference on Image Analysis, pp. 599-606.

[Vlontzos and Kung 89]

Vlontzos, J. A. and Kung, S. Y.: (1989) "Hidden markov models for character recognition", 1989 International Conference on Acoustics, Speech and Signal Processing, Vol. 3, pp. 1719-1722.

[Vlontzos and Kung 92]

Vlontzos, J. A. and Kung, S. Y.: (1992) "Hidden Markov models for character recognition", IEEE Transactions on Image Processing, Vol. 1, No. 4, pp. 539-543.

[Vossen 86]

Vossen, M.: (1986) "Electronic page reader in use", Office Management (Germany), Vol. 34, pp. 1148. (In German)

[Waite 89]

Waite, M.: (1989) "Data structures for the recognition of engineering drawings", PhD Thesis, Nottingham Polytechnic.

[Wang and Jean 91]

Wang, J. and Jean, N.: (1991) "Automatic rule generation for machine printed character recognition using multiple neural networks", IEEE International Conference on Systems Engineering, pp. 343-346.

[Wang et al 93]

Wang, A-B., Huang, J. S. and Fan, K-C.: (1993) "Optical recognition of handwritten Chinese characters by partial matching", Second International Conference on Document Analysis and Recognition, pp. 822-825.

[Wang and Jean 94]

Wang, J. and Jean, J.: (1994) "Segmentation of merged characters by neural networks and shortest path", Pattern Recognition, Vol. 27, No. 5, pp. 649-658.

[Weaver 72]

Weaver, J. A.: (1972) "Reading machines", London, England: Mills & Boon.

[Webb and Kreutzer 72]

Webb, T. C. and Kreutzer, A. L.: (1972) "Computer reads meter readers' handwriting without help", Transmission and Distribution, pp. 58-60.

[Wells 92]

Wells, C. J.: (1992) "The use of orthographic and lexical information for handwriting recognition", PhD Thesis, Nottingham Polytechnic.

[Wiener 48]

Wiener, N.: (1948) "Cybernetics", Wiley, New York.

[Wilson 66]

Wilson, R. A.: (1966) "Optical page reading devices", New York: Reinhold.

[Wilson and Blue 92]

Wilson, C. L. and Blue, J. L.: (1992) "Neural network methods applied to character recognition", *Social Science Computer Review*, Vol. 10, ISS. 2, pp. 173-195.

[Wolberg 86]

Wolberg, G.: (1986) "A syntactic omni-font character recognition system", *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 168-173.

[Wright 88]

Wright, P. T.: (1988) "Algorithms for the recognition of handwriting in real-time", PhD Thesis, Trent Polytechnic.

[Xu et al 92]

Xu, L., Krzyzak, A. and Suen, C. Y.: (1992) "Method of combining multiple classifiers and their application to handwritten character recognition", *IEEE Transactions on Systems, Man and Cybernetics*, SMC-22, pp. 418-435.

[Yoshida 74]

Yoshida, K.: (1974) "Optical character reader for telephone exchange charge billing system", *Japan, Telecom. Rev. (Japan)*, Vol. 16, pp. 105-110.

[Yung and Green 88]

Yung, H. C. and Green, I. M.: (1988) "Omnifont character recognition based on fast feature vectorization", *SPIE, Visual Communications and Image Processing*, Vol. 1001, pp. 633-640.

[Zhao and Srihari 94]

Zhao, Z. and Srihari, S. N.: (1994) "Word recognition using ideal word patterns", IS&T/SPIE Symposium on Electronic Imaging Science and Technology, pp. 24-34.

[Zhao and Srihari 95a]

Zhao, S. X. and Srihari, S. N.: (1995) "Word recognition using a Lexicon constrained by first/last character decisions", SPIE, Vol 2422, pp. 98-104.

[Zhao and Srihari 95b]

Zhao, S. X. and Srihari, S. N.: (1995) "A word recognition algorithm for machine-printed word images of multiple fonts and varying qualities", Third International Conference on Document Analysis and Recognition, Vol. 1, pp. 351-354.

Appendix A. OCR for machine-printed documents

Machine printed character recognition research and applications have been active for more than three decades. It is believed to be a solved problem when the quality of the text is acceptable (minimum fragmented or touching characters). In the case of broken and touching characters, character segmentation represents the biggest challenge. Humans are still capable of recognizing very degraded characters.

A hierarchical system for character recognition with Hidden Markow model knowledge sources which solve both the context sensitivity problem and the character instantiation problem, is presented by [Vlontzos and Kung 89]. This system achieves 97 to 99% accuracy using a two level architecture and has been implemented using a systolic array thus permitting real time (1ms per character) multi font and multi size printed character recognition as well as handwriting recognition.

[Moukrim and Muller 91] described an application of the Artificial Neural Networks (ANN) to the recognition of printed characters. As a matter of originality they added to this connectionist approach an algorithm based on the principle of generation-elimination to improve the process. For their test data, the recognition was 96% with the ANN alone and better than 99% with the whole process. Those characters which were not recognized did not show any sign of mis-classification and hence remained unclassified.

[Houle and Eom 91] proposed a machine printed character recognition model based on a *priori* knowledge of character shapes. It was inspired by tremendous human capabilities to recognize noisy characters, and by the lack of robust

algorithms to recognize degraded characters found in address mail. They mainly focused on the creation of curvature primitives, which are the core of the proposed model, as the classification performance is strongly dependent on the input feature set. They used a set of clean characters from multiple fonts to emphasize their belief that a clean set of characters can be used to build an inference engine.

[Campigli et al 91] presented a complete prototype concerning the recognition of printed documents. The prototype provides an automatic windowing of the page in order to separate written parts from figures, formulas or other graphic elements. The written parts of the sheet are analyzed and recognized. An automatic searching of the appropriate font is carried out. An improvement in the performance of the statistical recognizer by means of an spelling aid support (especially important for broken or connected characters) may be achieved enabling recognition of above 99%.

A neural network model, namely, Kohonen's Feature Map, together with the Optimal Feedforward Network are used for variable font machine-printed character recognition with tolerance to rotation, shift in position and size errors [Baykal and Yalabik 92]. This network together with the Learning Vector quantization approach are able to implement an inspection system which determines the orientation of the fonts. Rectangular and Minimal Spanning Tree (MST) neighborhood topologies are experimented with. It was observed that similar characters are mapped close to each other at the output plane, as expected. Out of eighty test samples of four different fonts, 87% are recognized correctly. The results are encouraging and the system might be improved by better normalization, large output planes and improved MST concept.

Most recently [Sabourine and Mitiche 92] presented an OCR system, which uses a multilayer perceptron (MLP) neural network classifier. The classifier uses shape

information (tangent) to classify characters. One test set of five unseen documents, the MLP classifier was capable of recognizing 96.7% of characters compared to 95.9% for an OCR system based on dynamic contour warping (DCW). This performance is considered significant.

More work in this using Artificial Neural Networks with varying degree of success has been carried out by a number of researchers including [Wang and Jean 94], [Garris et al 91], [Wang and Jean 91], [Togawa et al 91], [Baykal et al 91], [Umeno and Zhu 87], [Mostert and Brand 92].

[Cheng et al 92] introduced a robust method which can be used to recognize Machine-printed English characters. In this method, they first presented an Invariant Matrix (IM) corresponding to a unique character image under the polar system. This matrix is good in the sense that it is insensitive to image translation, scaling, rotation and noise. Based on invariant matrix, a set of similar discrimination functions (SDF) of English characters were determined. Considering these SDF functions they proposed a feature extraction and recognition method. Finally, according to their recognition model, they designed a hierarchical classifier to recognize English characters. The results obtained from the experiments performed in this work indicated achievement of 100% recognition accuracy for all English characters.

In another study by [Shchepin and Nepomnyashchii 91] a new approach to image coding, based on ordinate-preserving plane homeomorphism invariants, is presented. A critical-points graph bearing full information on these invariants is outlined. Appropriate noise-smoothing techniques are developed. The proposed method has been implemented and tested with a new optical character recognition program CRIPT (CRITICAL PoinTs). The results showed that recognition rate of 99.5-99.9% was achieved for any laser-printed text. It reduced to 99.0% for NLQ-

printed texts and dramatically dropped for texts with fractured characters or with numerous touching between symbols.

[Verikas et al 89] presented a comparative study of eight structural-statistical character recognition algorithms. They have discussed results and various merits and demerits of these algorithms. The comparison of these algorithms indicated that the recognition rates of printed and handwritten alphanumeric characters varied from 90 to 99.9 per cent.

Major achievements were made towards the development of a high-speed optical character recognition workstation for characters of various fonts and sizes by [Yung and Green 88]. The system is based upon an efficient feature extraction concept centered around an edge-vectorization technique. The technique has been demonstrated on an IBM-PC/XT (without coprocessor) to operate at least 25 times the speed of conventional OCR techniques, achieving a 100 per cent recognition rate with learned characters and 87 per cent with unlearned.

Not only recognition algorithms but also post-processings are important for increasing the total accuracy of the character recognition and speech recognition. One such post-processing is spelling correction [Takahashi et al 90]. In this work two approaches have been proposed for correcting spelling by means of word dictionary: (1) exact match methods and (2) best-match methods. After having a comparative study, the research workers designed and developed a fast spelling correction method based on the best-match method. It is a two-step procedure, consisting of candidate-word selection and approximate string matching between the input word and the selected candidate words. They applied this spelling-correction method to the post-processing of a printed alphanumeric OCR on a personal computer, thus making their OCR more reliable and user-friendly.

[Al-Badr and Haralick 95] have given the design and implementation of a system that recognizes machine-printed Arabic words without prior segmentation. This technique is based on describing symbols in terms of shape primitives. The advantage of using this whole word approach over segmentation approach is that the result of recognition is optimized with regard to the whole word. The experimental results obtained using a lexicon of 42,000 words show a recognition rate of 99.4% for noise-free text and 73% for scanned text.

[Zhao and Srihari 95b] proposed a word recognition algorithm which can be used in a general domain involving word images containing a wide range of font types and highly variable qualities. It bypasses the errors committed in character segmentation and identification. The first and last letters of the word were recognized to reduce the size of given lexicon. The experimental results show that the recognition performance has remarkable improvement by using lexicon reduction. The overall performance of this algorithm is better compared with the OCR approach, especially when image quality is degraded.

[Chen and Lui 92] performed experiments of printed character recognition using four different classifiers: back propagation network (BP), k- nearest-neighbour, (kNN), learning vector quantization (LVQ), and radial basis functions (RBF). The training set consists of 5,000 characters, the test set has 2,973 characters. There are 256 input dimensions and 36 output classes. Preliminary results show that for the recognition rate on the test set BP is about 99%, kNN is 97.71%, LVQ and RBF are about 97%. Although there are small differences between their performances, basically we think they all can reach roughly the same recognition rate if enough hidden units, training examples, codebook vectors, and basis functions are given. The real differences are memory size required, training time, and classification time.

A new neural model is proposed for pattern recognition, suggested by the structure of the visual nervous system [Umeno and Zhu 87]. This model consists of a moving fixation point and a feature detecting system. The feature detection is performed in the neighbouring area of fixation point. The simulation on computer and the results of recognizing a set of alphabet characters showed that the rate of correct pattern recognition to be 99.3 per cent. Because of its simplicity the model is of use in the realization of pattern recognition devices.

In many OCR systems, character segmentation is a necessary preprocessing step for character recognition. It is a critical step because incorrectly segmented characters are not likely to be correctly recognized. The most difficult cases in character segmentation are broken and touching characters.

[Tsujiimoto and Asada 91] have presented an efficient and powerful character segmentation method for touching characters. This approach employs knowledge about character composition as well as knowledge about omni-fonts to resolve ambiguity in segmenting touching characters. Experiments were conducted for documents commonly encountered in daily use, in order to evaluate the presented approach. Software realization achieved a speed of 120 characters per second with 99.7% accuracy.

[Lu 93] has discussed the problem of segmenting touching characters in various fonts and size in machine-printed text. He classified the touching characters into five categories: touching characters in fixed-pitch fonts, proportional and serif fonts, ambiguous touching characters, and string with broken and touching characters. He developed different methods for detecting multiple character segments and for segmenting touching characters in these categories. His methods use features of characters and fonts and profile models. He mentioned that grey-

scale images contain information for character segmentation and that segmentation directly from gray-scale images will give promising results.

Appendix B. OCR for hand-printed documents

The ability of reading handwritten characters provides a very attractive and economic means of using machines to capture and process the ever increasing volumes of data generated by mankind, e.g. mail, cheques, account sheets, transaction statements, computer programs, and many other business and scientific applications [Suen 78]. Over the past two decades, serious efforts have been devoted to this challenging subject by a large number of scientists all over the world. A number of OCR machines with limited handprint reading capabilities have been put into practical applications. Advances to this field have been presented in the paper [Suen et al 80]. In that paper, the recognition of numerics, alphanumeric, FORTRAN and Katakana characters was analyzed together with on-line handprint recognition, substitution and rejection rates were also discussed.

The recognition of handwritten characters consists of a number of processes such as data capture by the scanner, smoothing and cleaning by the preprocessor and feature extraction by the feature detector for subsequent classification.

[Kwan et al 79] studied comparison of four recognition algorithms; namely geometrical and topological features, n-tuples, moments, and characteristic loci. A large number of handprint samples were tested, comprising 12000 ANSI and non-ANSI samples. The recognition rates obtained from these algorithms varied from 91 to 99 per cent. The method employing geometrical and topological features appeared to be superior to the other three schemes in terms of recognition rate and computer time.

[Hsieh and Lee 92] presented a model-guided structural matching for recognizing handwritten off-line Chinese characters. According to the stroke

writing sequence, each character is described by an on-line model, which is a one-dimensional string consisting of a stroke sequence interleaved with relationships between two consecutive strokes. After an unknown input is thinned and line approximated, all possible strokes are extracted. In the recognition process, they matched the strokes with this defined in the on-line model. The matching process is formulated as a tree searching algorithm guided by the relationships in the model. The modelled character is taken to be a candidate if there is a feasible path which satisfies the relationship defined in the on-line model and the number of missing strokes in the input is less than a given threshold. The input is recognized as the character with the greatest number of matched strokes among all candidates. Experimental results on 300 frequently used characters in a database which contains 5401 Chinese characters and about 250 variation for each one, showed the recognition rate of more than 90%.

In a work presented by [Neisser and Weene 60] nine human observers were given the task of identifying isolated hand-printed characters. Their individual accuracies ranged from 94.9% to 96.5%, and even their pooled best guess was right only 96.8% of the time. These figures can serve as standards for the accuracy of mechanical devices for letter-recognition.

[Dasarathy and Kumar 78] developed and tested a novel system for the recognition of handprinted alphanumeric characters. The system can be employed for recognition of either the alphabets or the numeral by contextually switching on to the corresponding branch of the recognition algorithm. The two major components of the system are the multistage feature extractor and the decision logic tree-type categorizer. An information feedback path is provided between the decision logic and the feature extractor units to facilitate an interleaved or cursive mode of operation. This ensures that only those features essential to the

recognition of a particular sample are extracted each time. Test results showed reliability of the system in recognizing a variety of handprinted alphanumeric characters with an accuracy close to 100 per cent.

[Tang and Suen 92] presented a new approach called regional projection transformation, which converts a compound pattern into an integral object. They used features from the diagonal-diagonal regional projection transformation (DDRPT) to recognize a large set of characters which contain a lot of compound patterns. An impressive result was achieved in an experiment of recognizing Chinese characters. They reported high recognition rates, for recognizing 3,000 characters, with an accuracy of 99.28%.

It has been seen that Neural Network methods show great promise for providing highly accurate, noise resistant, parallel algorithm and data organization for a wide range of problems where 'humanlike' recognition is needed. One specific area of recognition, the conversion of images of handwritten to computer representation is used by [Wilson and Blue 92] to investigate the two major types of machine learning i.e. supervised learning and self-organization and to demonstrate the capabilities of neural network algorithms. The network can be trained to characterize noise as well as data. This type of behaviour is sometimes called overtraining in the neural network literature. In the character recognition problems, the source of the error is usually under sampling rather than overtraining. The use of real character recognition problems illustrates that historically difficult problems in character recognition can be solved with sufficient accuracy to be of commercial value.

[Pintsov 93] discussed the general methodology of structural handwriting character recognition systems. A model of handwriting is formulated as a human to human communication model and various implications of this model for

handwriting recognition algorithms are considered including optimal criteria for digitization, size and representativeness of training database and ultimate performance level.

[Matsui et al 93] presented a variety of combination methods for the multi-expert system using the best recognition algorithms and described their promising results. Their results showed that the highest recognition rate was 96.22% while its substitution rate was 0.37%, demonstrating that Japanese researchers have reached a considerably high level of handwritten numeral recognition when using writing frames.

[Gudesen 76] presented results of handprinted recognition experiments where different preprocessing techniques were coupled to the front end of a recognition system. Preprocessing turned out to be a very powerful tool in order to reduce error rates of recognition system. Normalizing character dimensions yielded the best results reducing error rates by a factor of 6. Computational effort caused by preprocessing was low in comparison to recognition system.

[Suen and Shillman 77] tested a computer algorithm on the U's and V's from an IEEE data set of unconstrained handprinted characters. These are the most difficult characters for humans to accurately label. The machine recognition rate of over 94% is higher than the average human recognition rate obtained on the same digitized characters and is well above typical recognition rates reported in the literature on unconstrained handprinted characters. The excellent performance of this simple machine algorithm on these troublesome characters demonstrates the superior power of incorporating psychologically based features into character recognition algorithms.

[Kalberg et al 93] retrofitted the AEG AL880 automatic reading of handwritten postcodes in the boxes on postal items such as envelopes and postcards. The retrofit uses 36 T800 transputers and can process 10.8 postal items per second with a mean processing time of 3.3 second per item. It was tested on a batch of 216,000 cards with the handwritten postcodes in boxes. The retrofit was able to read 75% of these postcodes (25% were rejected), with a one per cent error in the yield. The algorithm used to read handwritten postcode printed in boxes has as its input a black-and-white image of a postal item recorded by the AEG camera. The image has a resolution of 200 dpi, both horizontally and vertically. The algorithm consists of 6 steps; (1) connected component labelling (2) location of postcode boxes (3) elimination of postcode boxes (4) segmentation of postcode characters (5) recognition of characters and (6) verification of postcode's existence.

In this work [Adachi et al 88] new techniques are described for handprinted numeral recognition and generation of various sized high-quality graphic characters, which enable efficient communications between facsimile terminals and computer centres. To achieve extremely precise numeral recognition despite the poor characteristics of pattern obtained by facsimile terminals because of their inferior resolution and quantizing scheme, a supplementary pre-processing method and a similar-shaped character discrimination methods have been developed. Using these procedures, about 80% of the characters which can not be recognized by the basic recognition procedure are rendered recognizable.

[Gentric 95] demonstrated that word recognition rate can be greatly improved by enhancing the nature of the formation provided by the character recognition classifier to the lexical processor. Their experimental results clearly indicate that the use of Radial Basis Function (RBF) classifier is advantageous over the use of the classifier confusion matrix for substitution cost estimation. He reported that the

major improvement in system performance was from 94.7% to 97.4% and mentioned that this is due to the highly multi-modal and ambiguous nature of handwriting.

A number of approaches/algorithms used for recognition of different types of handwritten characters in different applications, and the results obtained from them may be found from a lot recent publications such as [Lecolinet and Moreau 90], [Granlund 72], [Tou and Gonzalez 72], [Leroux and Salome 90], [Al-Yousefi and Udpa 92], [Fukushima et al 91], [Verikas et al 89], [Vlontzos and Kung 92], [Kertesz and Kertesz 92], [Tsukumo 92], [Lee et al 93], [Wang et al 93], [Favata et al 94], [Hendrawan and Downton 94], [Joe and Lee 95] and [Congedo et al 95].

Appendix C. Sample facsimile messages

Sample facsimile messages used to evaluate the performance of the developed recognizer are given in this appendix.

Note: Facsimile images have been reduced to fit on an A4 page for presentation purposes.

(1) Facsimile number 1



KOLEJ BANDAR UTAMA

No. 50, Jalan SS 21/82, Damansara Utama, 47400 Petaling Jaya, Selangor, Malaysia. Tel: 03-7173200 Fax: 003-7172733

Our Ref : KBU/2-B-02/661/94

20 October 1994

Professor RJ Whitrow
Head, Department of Computing
The Nottingham Trent University
Burton Street
Nottingham NG1 4BU
United Kingdom

Dear Professor Whitrow

It was a great pleasure for me to meet with you again during my recent trip to the University.

I am pleased to learn from you that your department is able to admit more than 20 students into the B Eng (Hons) in Electronics and Computing. That being the case and if the initial figure can be increased to 40 with possibility of further increase in the future, the college is interested to begin planning for the commencement of this course in April 1995. I understand that this course has been accredited by IEE, UK but not by BCS.

For your kind information, please be informed that the college has been accredited by the University during the accreditation visit in January 1993 to conduct Year 1 and Year 2 of this honours degree.

I understand that this course has been modularised. Although I have the old definitive course document, I believe this document will probably be outdated. Can you kindly send me the latest copy please.

For your kind information, attached herewith are the curriculum vitae of our lecturers in Engineering and Computer Science.

I look forward to receiving your reply.

With best wishes

Yours sincerely
KOLEJ BANDAR UTAMA

LEE KER FOON
Principal

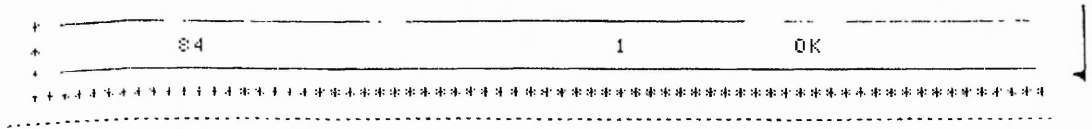
Enc



SEE HOY CHAN HOLDINGS GROUP

P. D. 7

(2) Facsimile number 4



FOR THE ATTENTION OF : Mr L McMannus, THE NOTTINGHAM & TRENT UNIVERSITY
THE DIGITAL SIGNAL PROCESSING EXHIBITION AND CONFERENCE.

**Ramada Hotel
Heathrow
London**

December 6th & 7th

CAR PARKING FOR DSP UK '94 - REDUCED CHARGE

Thank you for pre-registering for DSP UK '94.

We have arranged alternative car parking with The Pink Elephant car park which is one of Heathrow's Long Term parks on the perimeter road. It will be well sign-posted by yellow AA signs and courtesy buses will be on-hand to take you to the Ramada. Alternatively it is about half a mile so walking is not out of the question (for some!).

Pink Elephant will charge you £7 for the day. However when you arrive at the Hotel please present your car park ticket at the DSP UK Information Point in the stand area and we will refund £3 in cash.

If you intend to stay for two days at the exhibition, we suggest that you try to move your car into the Ramada car park on Tuesday evening.

We apologise for any inconvenience caused but we expect this event to be very popular.

TRAVELLING BY UNDERGROUND.

If you wish to travel to the Ramada by Underground, take the Piccadilly line to Heathrow and then take the Ramada courtesy bus from terminal 1, 2 or 3 to the Hotel. These leave approximately every 20 minutes.

RAMADA ROOM RATES

We have negotiated a special reduced room rate of £75 with the Ramada. They are on 081-897-6363. Please mention this reduced rate for the show.

(3) Facsimile number 6

Attached herewith is the "Proposed Schedule for Teaching the Bridging Course Leading to Final Year of TNTU B Sc (Honours)" for your comments please. Please note items 3 and 4. Because of the short period of time available to conduct the Bridging Course, we find it necessary to commence the course immediately after the KBU Examination Board meeting. In the event that TNT-U Board changes any recommendation made by the KBU Board of Examiners resulting in any student(s) not qualifying for the bridging course he will have to leave the bridging course. This is the same procedure agreed by the University for the Bridging Course in Electrical and Electronic Engineering.

To apply for student visa to UK, the student must have an unconditional admission offer letter from the University. When the students of the Bridging Course are confirmed, I shall write to you again to inform you who they are. This unconditional admission offer letter should be issued earlier but appropriately dated and kept in the Malaysian office of the University. As soon as the KBU Examination Board meeting for the Bridging Course is held, qualified students will be given the letters to enable them to apply for student visas. We have tentatively fixed 11 September 1995 for students to see the Immigration Officer of the British High Commission. The application is on block basis and provided everything is alright, the students will get the approved visas by 15 September 1995, a few days before departure. Please be informed that KBU is expecting to have about 100 students applying for these visas in September and these students come to Kuala Lumpur from all over Malaysia. It is an enormous task.

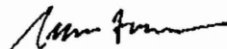
You mentioned that the second year of the Computer Studies Degree course is currently being franchised to North Lincoln College, UK and that the University has developed teaching material for the course. As the bridging course is based on the degree programme, these materials will be of help to us. Would you kindly send them to us please ?

I hope that the proposals made in this letter are acceptable to you.

I look forward to receiving your response.

With best wishes

Yours sincerely
KOLEJ BANDAR UTAMA



LEE KER FOON
Principal

(4) Facsimile number 7

TC 00744602492513

P.06



KOLEJ BANDAR UTAMA

No. 50, Jalan SS 21/62, Damansara Utama, 47400 Petaling Jaya, Selangor, Malaysia. Tel: 03-7172200 Fax: 003-7172733

Our Ref : KBU/2/A-02/741/95

11 February 1995

Mr John Smith
Department of Computing
The Nottingham Trent University
Burton Street
Nottingham NG1 4BU
United Kingdom

Dear Mr Smith

Thank you for your fax of 8 February 1995. I am pleased to learn that you and the BTEC moderator, Mr Hind will be coming to the college. The dates of your visit namely, 28 February and 1 March are suitable for us. I have got in touch with Tunku Abdul Rahman College Principal, Dr Lim Khaik Leang regarding your visit to the college for one day. He has kindly agreed to the visit on 1 March 1995. I informed the Principal, Dr Lim that both of you would like to get to know TAR College better and to familiarise with the Computer Science programmes at TAR College. In particular, you would like to learn more about the Certificate in Computer Studies and the Certificate in Computing and Accounting.

Miss Anizah has left the College and gone to Penang Science University to do the Master's in Artificial Intelligence. Miss Christine Lee Siew Ken joined the college recently. Her curriculum vitae is attached.

<u>Name</u>	<u>Qualification</u>	<u>Academic/Industrial Experience</u>	<u>Subjects Teaching at KBU</u>
Christine Lee Siew Ken	B Sc (Computer Sc) U of Wollongong Australia M Sc (Computer Sc) U of Wollongong	Systems Engineer at Numedia - 6 months	Systems Methodology Basic Compt Principles Computer Programming Information Processing

I look forward to seeing you and Mr Hind in the college.

With best wishes

Yours sincerely
KOLEJ BANDAR UTAMA

LEE KER FOON
Principal

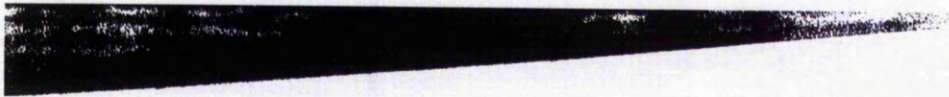


SEE HOY CHAN HOLDINGS GROUP

ERROR: timeout
OFFENDING COMMAND: timeout

STACK:

(5) Facsimile number 8



23-02 '95 10:45

0815234002

SIMPLY COMPUTERS

001



Nottingham Trent College
Finance Dept
Burton Street
Nottingham
NG1 4BU

Quote No. : S182180

Dear MR BARGIELA

I am delighted to enclosed the peripherals quotation you recently requested.

As you probably already know we're unlike other suppliers. For we've taken the time to find out exactly what you want - before asking you to part with any money. That's because not only do we strive to offer you the most competitive price possible, but also products that won't let you down, or fall short of your requirements.

You'll also be pleased to hear that peripherals aren't on the periphery when it comes to after-sales service. All of them are backed by a 12 month Return-To-Base Warranty.

We also won't ever try and fob you off with second best equipment. If the peripheral you want isn't in stock, we'll tell you - the moment you place your order. You can then decide for yourself whether you want to wait, or select an alternative product.

Of course, if you have any queries regarding the enclosed quotation please don't hesitate to call us today on 0181 523 4020.

I look forward to hearing from you soon.

Yours sincerely,

Carol

P.S. For extra fast service please use this quote number S182180 when confirming your order.

(6) Facsimile number 9



TRANSTECH
Parallel Systems

Facsimile Cover Sheet

To : Victor Smith
Company : Pacer Systems Ltd

Tel. No :
Fax. No : 01159 486518

From : Duncan Curry
Company : Transtech Parallel Systems
Tel. No : 0494 464303
Fax. No : 0494 463686

Date : 29 March 1995
No. of Pages including Cover Sheet :

CONFIDENTIALITY NOTICE: THIS FAX TRANSMISSION IS INTENDED ONLY FOR THE INDIVIDUAL OR ORGANIZATION NAMED ABOVE. IF YOU ARE NOT THE INTENDED RECIPIENT, YOU ARE HEREBY NOTIFIED THAT ANY DISCLOSURE, COPYING, DISTRIBUTION OR THE TAKING OF ANY ACTION IN RELIANCE ON THE CONTENTS OF THIS FAX IS STRICTLY FORBIDDEN. IF YOU HAVE RECEIVED THIS FAX IN ERROR, PLEASE CALL US IMMEDIATELY AND DESTROY THE ENTIRE FAX.

Ref: DC.NW813

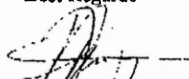
Dear Mr Smith

I have attached our quotation for our *TDMB436*.

This price is valid for 30 days but is open to change as Transtech runs different promotions and discounts.

In the meantime, if I can be of any further assistance, please do not hesitate to contact me.

Best Regards


Duncan Curry
Internal/Sales Engineer

Transtech Parallel Systems Limited

12 Manor Court Yard • Hughenden Avenue • High Wycombe • Buckinghamshire • HP11 5RE
Telephone +44 (0) 494 464303 • Facsimile +44 (0) 494 463686

(7) Facsimile number 10

資訊工程研究所

INFORMATION ENGINEERING
NATIONAL CHENG KUNG UNIVERSITY

中華民國台灣省台南市大學路一號

1, TA-HSUEH ROAD, TAINAN, TAIWAN, R.O.C.
TEL: (06)2757575 EXT 62500
TEL/FAX: (06)2747076

Fax to: Professor Robert Withrow
Trent University
Department of Computing
Fax No: 44-602-486-518

From		
11 NOV 1994		
Ref No.	File Ref	File
4737		

Nov. 15, 1994

Dear Prof. Withrow,

As you are one of the invited honorable guests of the 4th IWFHR, I am pleased to inform you that the 4th IWFHR will provide you an economy class of round-trip airplane ticket, lodging and meals in Grand Hotel from Dec. 6 to Dec. 10, 1994.

Please first purchase the ticket by yourself then we will reimburse your ticket later in Taiwan. And we have already reserved the room for you, please go directly to check in at the reception desk of the Grand Hotel.

If you have any question, please do not hesitate to contact me. Looking forward to seeing you at the conference.

Sincerely yours,

Jhing Fa Wang
Jhing Fa Wang
General Chair
4th IWFHR

For Prof. Withrow
Withrow RTO
overseas

+886 6 2747076

P.01

(8) Facsimile number 11

15-02-1995 17:58

FPCM

TO 00744602486518

P.05



KOLEJ BANDAR UTAMA

No. 90, Jalan 98 21/82, Damansara Utama, 47400 Petaling Jaya, Selangor, Malaysia. Tel: 03-7172600 Fax: 03-7172733

Our Ref : KBU/1-B-02/735/95

9 February 1995

Mr John Smith
Department of Computing
The Nottingham Trent University
Burton Street
Nottingham NG1 4BU
United Kingdom

Dear Mr Smith

I hope that you have received my fax to you dated 13 January 1995 proposing the conduct of the bridging course. My proposal requires that Mathematics for Software Engineering be conducted in parallel with the HND Computer Science, final year and the College plans to begin teaching the subject on 13 February 1995 which is next Monday. I hope that you can give us the approval as soon as possible.

With best wishes

Yours sincerely
KOLEJ BANDAR UTAMA

LEE KER FOON
Principal



SEE HOY CHAN HOLDINGS GROUP

P. 05

(9) Facsimile number 12



KOLEJ BANDAR UTAMA

No. 80, Jalan SS 21/62, Damansara Utama, 47400 Petaling Jaya, Selangor, Malaysia. Tel: 03-7173200 Fax: 03-7172738

Our Ref : KBU/2/A-01/729/95

13 January 1995

Mr John Smith
Department of Computing
The Nottingham Trent University
Burton Street
Nottingham NG1 4BU
United Kingdom

Dear Mr Smith

Thank you for the Bridging Course which arrived by courier. We propose to conduct the bridging course as follows :

A.(i) Mathematics for Software Engineering to be conducted in parallel with the HND Computer Studies Final Year. The assessment which is valued at 50% will be made as the subject progresses but the end test will be given together with the other subjects of the bridging course.

ii) The proposed structure of the time-table for Mathematics is as shown below :

<u>Subject</u>	<u>Contact Time Per Week</u>	<u># of Weeks</u>	<u>Period</u>
Mathematics for Software Engineering	2.5 hours	15 weeks	13-2-95 to 19-5-95

B. All other subjects to be conducted after the HND final year end test. The proposed structure of the time-table is as follows :

<u>Subject</u>	<u>Contact Time Per Week</u>	<u># of Weeks</u>	<u>Period</u>
Systems Software	3	8	3-7-95 to 25-8-95
Advanced Program Design	3	8	"
Functional Programming	3	8	"
Formal Methods I	3	8	"
Business Computer Interaction	2.5	8	"

1.



SEE HOY CHAN HOLDINGS GROUP

(10) Facsimile number 13



Message To: Leo McManus @ Nottingham Trent University

Message From: Karl Wale

Fax No: 0602 486518

Date: 20/02/95

Ref: LSI 2002/kw/02

No of Pages (including this one): 1

(Please call if your copy is illegible or incomplete)

Dear Leo

Following our conversation please find attached our quotation as requested.

Part number	Description	Quantity	Unit price
DSK-3L	3L Parallel C Debugger Support Kit. Allows 3L software to be debugged using DB40	1	650.00

Availability :

Currently please allow 1-2 weeks assuming prompt placement of order

Notes :

1. This quotation is valid for 30 days.
2. LSI standard terms and conditions apply.
3. All prices in Pounds Sterling excluding VAT.
4. Carriage is charged at £17.50 per shipment including insurance.

I hope this information is of use, if however I can be of any further assistance, then please do not hesitate to contact me.

Best regards.

Internal Sales Engineer

Loughborough Sound Images plc . Loughborough Park . Ashby Road . Loughborough . Leicestershire . LE11 3NE . England
Sales - Tel. +44 (0)1509 634300 . Fax: +44 (0)1509 634333 . General - Tel: +44 (0)1509 634444 . Fax: +44 (0)1509 634450

(11) Facsimile number 14



WORKERS INSTITUTE OF TECHNOLOGY
Jalan Pandamaran, 42000 Port Klang,
Selangor Darul Ehsan, Malaysia.
Tel: 03-3688859
Fax No: 603-3675046

FACSIMILE COVER SHEET

TO : Mr. John Smith
COMPANY : Nottingham Trent University
FAX NO : 115 - 9486518

FROM : Dr. H. J. Mohamed Thalha
DATE : 21/2/1995

TOTAL NUMBER OF PAGES 1 INCLUDING THIS COVER SHEET.
IF YOU DO NOT RECEIVE ALL COPIES, PLEASE TELEPHONE: 03-3688859
IMMEDIATELY.

THANK YOU.

MESSAGE:

Dear Mr. John Smith

Thank you for your fax note of 20th February, 1995. The week starting 27th February, 1995, WIT is closed for vacation because of the Ramadhan festival on the 3rd and 4th of March, 1995 therefore all staff will be away to their respective home towns to celebrate the festival on the 3rd of March, 1995 and hence I regret that we may not be able to meet you on Thursday, 3rd of March, 1995 as requested. However, I look forward to another visit of yours where we may be able to take up the matter referred in your fax for discussion.

Yours sincerely,

Michael. h.

603 3675046

(12) Facsimile number 15

15-02-1995 17:57 FROM

TO 00744602486513

P.01

**PROPOSED SCHEDULE FOR TEACHING THE BRIDGING COURSE
LEADING TO FINAL YEAR OF TNTU B SC (HONOURS)
COMPUTER STUDIES**

- | | | |
|----|---|----------------------|
| 1. | Year 2 Final Examination Ends | 23 June 1995 |
| 2. | KBU Examination Board Meeting | 30 June 1995 |
| 3. | TNTU Official Results Communicated to KBU | 10 July 1995 |
| 4. | Commencement of Bridging Course | 3 July 1995 |
| 5. | End of Bridging Course | 25 Aug 1995 |
| 6. | Bridging Course Examination Dates | 1, 2, 4, 5 Sept 1995 |
| 7. | KBU Examination Board Meeting | 7 Sept 1995 |
| 8. | Successful candidates leave for TNTU, UK | 20 Sept. 1995 |

Subject to approval

10 January 1995

P.01

(13) Facsimile number 17

DIFFICULTY IN SOURCING EXABYTE?



CALL THEIR TOP EUROPEAN DISTRIBUTOR OF THE YEAR

EXABYTE® 

ALL DRIVES, LIBRARIES & MEDIA AVAILABLE EX-STOCK



TEL: 0171 704 0202

FAX: 0171 704 1100

(14) Facsimile number 19

AZTECH NOW PARTNERS *THE* STORAGE SPECIALIST

6X IDE CD-ROM OEM VERSION	£ 43.00
8X IDE CD-ROM OEM VERSION	£ 81.00
6X MULTIMEDIA KIT	£119.00
16 bit sound card with 3D	£ 28.00
32 bit sound card with 3D	£ 49.00

AVAILABLE EX-STOCK



TEL: 0171 704 0202

FAX: 0171 704 1680

(15) Facsimile number 21

18/04/1996 08:29 441159486838
17/04 '96 10:59 FAX 01793 444005

EPSRC P A C

PAGE 02
003

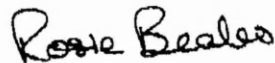
those nominated for any particular venue, but we may have to restrict numbers if certain venues are oversubscribed, London in particular.

Please complete the attached form and return by 29 March 1996. Should you have any queries please contact Mrs Tanya Cottrell on 01793 444075.

I hope that your institution will be able to participate in these events and we look forward to meeting your representatives.

You may wish to note that the Chemistry and Process Engineering Programmes will be holding a joint seminar on their activities during the afternoon of the regional seminars. Separate invitations will be sent in due course.

Yours faithfully



Rosie Beales

rb781.doc

(16) Facsimile number 24

PRESS AND PUBLIC RELATIONS - PRESS OFFICE
THE NOTTINGHAM TRENT UNIVERSITY
FAX NUMBER (0115) 948 6558 E MAIL:- PRONTO@NTU.AC.UK

FAX

DESTINATION FACULTY OF ENG + COMPUTING,
FOR THE ATTENTION OF PAULINE
FAX NUMBER 6506 6518
FROM LYNN REDGEWELL
DATE / TIME 1 MAY, 1996 12:07 PM
NUMBER OF PAGES (INCLUDING THIS ONE) 2.

MESSAGE FOLLOWS:-

RE STUDENT CHOICES

Attached is a brief for all the students involved. Hopefully this will give them an idea of the background of the programme, and what to expect from Friday morning.

I'd appreciate it if you could circulate this to the students involved, or to the course tutors who can then distribute it.

Many thanks for your help,

L Redgewell.

IN THE CASE OF ANY PROBLEMS PLEASE CONTACT:-
PRESS AND PUBLIC RELATIONS, THE NOTTINGHAM TRENT UNIVERSITY
TEL: - (0115) 948 6542

P 01

(17) Facsimile number 25

THE NOTTINGHAM TRENT UNIVERSITY

DEPARTMENT OF COMPUTING
FAX NUMBER +44 (0)115 9486518

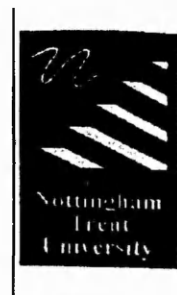
For the attention of: MR TONY CLEWETT
Company: DRAGON SYSTEMS UK LTD

Fax Number: 01242 678301

Date: 8.5.96

From: LINDSAY EVETT

No. of Pages
(including header) 1



Message:

Dear Mr Clewett,
Please could you
fax me prices and details of
DragonDictate systems?

Thanks,
Lindsay Evett

IN CASE OF ANY PROBLEMS WITH THIS FAX MESSAGE, PLEASE CONTACT THE NOTTINGHAM TRENT
UNIVERSITY, DEPARTMENT OF COMPUTING, NOTTINGHAM. NG1 4BU. ENGLAND.

(18) Facsimile number 27

THE NOTTINGHAM TRENT UNIVERSITY

DEPARTMENT OF COMPUTING
FAX NUMBER +44 (0)115 9486518



For the attention of: MR BRIAN FIELDER

Company: WACOM

Fax Number: 0049 2131 101 760 Date: 8.5.96

From: LINDSAY EVETT No. of Pages
(Including header) 1

Message:

Dear Mr Fielder,
Please fax me
details and prices for WACOM
electronic paper products?

Thanks,
Lindsay Evett.

IN CASE OF ANY PROBLEMS WITH THIS FAX MESSAGE, PLEASE CONTACT THE NOTTINGHAM TRENT
UNIVERSITY, DEPARTMENT OF COMPUTING, NOTTINGHAM. NG1 4BU. ENGLAND.

(19) Facsimile number 29

FACSIMILE TRANSMISSION

DATE:



LINCOLN CENTRE
Monks Road
Lincoln
Lincolnshire
LN2 5HQ
Telephone (01522) 510530
Fax (01522) 512930

TO: Pat Smith x 2150

ATTN:

FAX NO: 0115 9486518

FROM: Pam Doherty

NO. OF PAGES
(incl. cover sheet)

2

MESSAGE:

see attached

Associate College of the Nottingham Trent University

512930

P. 01

(20) Facsimile number 33

PRODUCT SPECIFICATION & PRICING

Quantity	Product & Description	Unit Price
1	TDMB436 <ul style="list-style-type: none">● Monochrome framegrabber/8 bit RGB display● Two 1024 x 1024 video stores● 1024 x 1024 overlay plane● TMS320C40 processor● 4MBytes zero wait state local memory and global bus expansion connector● Programmable capture and display up to 1024 x 1024 pixels● RGB/composite/S-Video/mono capture● Trigger input & output for event sync.	£4550.00

(21) Facsimile number 34



Make the right connection

Transceivers	Thinwire	£ 27
	Thickwire	£ 36
	Twisted Pair	£ 28
	Fibre Optic	£ 145
Modular Repeaters	Twisted Pair to Thinwire	£ 237
	Twisted Pair to Thickwire	£ 223
	Twisted Pair to Fibre Optic	£ 312
Hubs	8 port Twisted Pair Hub	£ 225
Printservers	2 port, 60 KByte/sec	£ 459
	2 port, DMA, 130 KByte/sec	£ 659

Products from Solair and Milan

Offer valid until 30th April 1995 All prices are exclusive of VAT & delivery, E & OE.

DATAMAN UK Ltd, Tel: 01423 358226 Fax: 01423 358262

DATAMAN
A EUROPEAN COMPANY

Return by fax to: **01423 358262**

- I am interested in the above offer, please call me.
 I am interested in the networking products above and would like further information

Name: _____
Company: _____
Address: _____

(22) Facsimile number 35

PACER
SYSTEMS LIMITED

High Technology Design & Computer
Controlled Machine Systems

UNIT 6
ROBIN HOOD INDUSTRIAL ESTATE
NOTTINGHAM NG3 1GE
Telephone: 0115 948 3128
Fax: 0115 948 3304

FOR THE ATTENTION OF:

DAVID MOORE.

COMPANY:

AXIOMATIC

FAX NO:

9480518

FROM:

Andrew Craig

NO. OF PAGES (including this page):

1 ✓

DATE:

13/3/95

MESSAGE:

DAVID

HAD PROBLEMS WITH OUR 2500 MACHINE
(PRODUCTION USE) "HANGING UP" SINCE IT WAS
CONVERTED BACK TO ROUTER USE FROM THE LASER
OPERATION.

BASICALLY AN AREA FILL GIVES UP WHEN IT
FEELS LIKE IT.

I DO NOTE THAT IT HAS U6 EPROMS IN IT
SO I HAVE BACKDATED THEM TO U5.30 AND
ALL SEEMS OK (ALTHOUGH I DON'T KNOW IF I
HAVE DONE ENOUGH TEST TO FULLY CONFIRM
THAT)

BUT THE U6 EPROMS ARE PENCILLED ONTO THE
EPROM WITH NO DATE SO I DO NOT KNOW WHETHER
THEY ARE UP TO DATE.

DO YOU NEED TO PROVIDE LATEST SO WE CAN MAKE
THE TESTING A LITTLE MORE METHODICAL?

Registered in England No 1879037
Directors: P G Marshall, MA, C.Phys, MInstP P D Thomas, B.Sc, M.Sc, Ph.D, C.Eng, M.I.E.E M K Bye

Andrew

(23) Facsimile number 37

SENT BY: CORPORATE SALES

3-22-95 15:08

FAG BEARINGS CORP. -

0602486518:# 1 / 1

*Refer To
Engineering Dept.*

Mr. K. Ayandokun,
Dept. of Computing,
The Nottingham Trent University,
Burton Street,
Nottingham,
NG1 4BU
United Kingdom.
Tel: 0115 9418418 ex4193
Fax: 0115 9486518
Email: kaa@doc.ntu.ac.uk

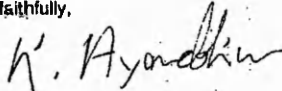
FAG Bearings Corp.,
118 Hamilton Ave.,
Box 811,
Stamford,
CT 06904,
USA.

Dear Sir / Madam,

Your company was mentioned in an article in the January 1992 edition of Machine Design as a manufacturer of instrumented bearings. The context was wheel speed sensing using bearings with integrated electronic sensors. Force sensing using bearings with integrated strain gauges was also mentioned in this article. I am looking for technical details of any instrumented bearings manufactured by your company of either type. In particular I wish to have details of the sensor types employed, maximum number of counts per revolution, dimensions of the bearings in your range and the type of output the sensor produces. I need this information for a research project that I am currently engaged in. I feel that aspects of this project might be of interest to technical development personnel within your company. I would be grateful if you could also provide me with the address and a contact name in your research or technical department so that I can send a detailed explanation of the work our group is doing.

Thank you for your assistance.

Yours faithfully,



Mr. K. Ayandokun.

To: Mr. K. Ayandokun

Unfortunately, we do not manufacture instrument bearings. Note your contact

FAG UK Limited
1 Hollinshead Court
Stafford Park Telford
CH12 9LJ Telford
Stafford Park Telford

(24) Facsimile number 38

**TRANSTECH SERVICE AND AFTER SALES
SUPPORT**

At Transtech we understand that the proposed products will be an important resource.

The post sales support package for these products is, therefore, comprehensive: -

ONE YEAR'S HARDWARE MAINTENANCE

The hardware maintenance plan includes: -

- Full access to Transtech's hardware telephone support hotline.

Primary Contacts: -

- Niel Clausen - Technical Manager
- Simon Smith - Applications Engineer

- On site visits in the event of a problem which cannot be solved on the telephone*

- Return to depot warranty

90 DAYS SOFTWARE MAINTENANCE

The software maintenance plan includes: -

- Upgrades to the latest software revisions as released
- Full access to Transtech software telephone support hotline

* A charge may be incurred

**
*
*
*
*
*
*
*
*
*
*
*
**

(25) Facsimile number 39

0815234002

P. 01

23/02 '95 10:45 ☎0815234002

SIMPLY COMPUTERS

002

QUOTATION



QUOTATION VALID FOR : 2 Weeks

Quotation To:

Nottingham Trent College
Finance Dept
Burton Street
Nottingham
NG1 4BU

QUOTE No.	S182180
QUOTE DATE	23 Feb 95
YOUR REF.	
ACCOUNT No.	N046339
FAX No.	01159486518

PRODUCT CODE	DESCRIPTION	VAT RATE	QTY	UNIT PRICE	NET PRICE
HC0205	Conner CFA850A 850Mb 12ms EIDE	17.5 %	1	295.00	295.00

DELIVERY	£10.00
SUB TOTAL	£305.00
VAT	£53.38
QUOTE TOTAL	£358.38

(26) Facsimile number 40

1995 17:55

FROM

TO 00744602486518

P.01



KOLEJ BANDAR UTAMA, 50 Jalan SS 21/62, Damansara Utama, 47400 Petaling Jaya, Selangor Darul Ehsan, Malaysia. Tel : 6 03-717 3200 Fax : 6 03-717 2733

010 60

FACSIMILE MESSAGE

To : *Mr John Smith*
Department of Computing From : MR LEE KER FOON, Principal
Fax No : *TNT-11* Date : *15-2-95*
of pages incl. this page : *6* Ref No :

Subject :

As spoken I relay here with
my letters to you as follows:
1. my letter dated 13 Jan 95
2. my letter dated 9 Feb 95
3. my letter dated 11 Feb 95.
If there is anything further that
you need, please let me
know.

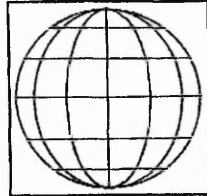
Lee Ker Foon

(27) Facsimile number 41

Tue 21 Feb 95 06:33

Sent from: 0225 775090

Page 1 of 3



**CORPDATA
LIMITED**

*Kestrel House · Mill Street · Trowbridge · Wilts BA14 8BE
Telephone: (0225) 775545 Fax: (0225) 775090*

F.A.O. Dr P.J. Thomas
The Microprocessor Centre

To : 01159486518

I am pleased to let you know about CorpSoft and, as promised, I've faxed details. I expect you'll find it an extremely useful software package to have, but have a look for yourself!

I've appended a trade order form for your convenience.

Regards

Les Lawler

VAT No 600 9669 41 Registered in England No. 2690712

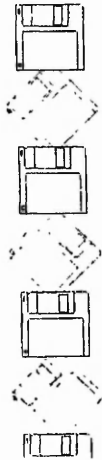
0225 775090

P. 01

Tue 21 Feb 95 06:33

Sent from: 0225 775090

Page 2 of 3



CorpSoft

The Ultimate Software Reference Guide

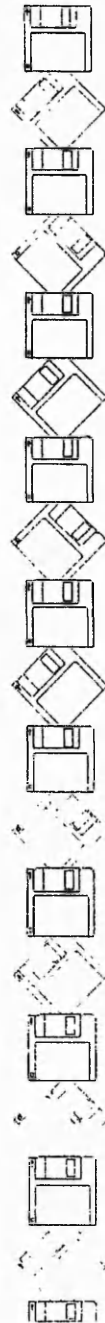
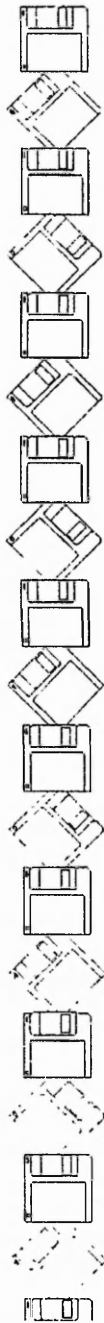
Have you ever wished you had a guide covering all the software products you could want, or need to find out about? Of course you have, because with so much software available, computer professionals like yourself

Corpdata have developed an inexpensive and convenient package for people who need information on available software choices. CorpSoft is designed like a book for intuitive ease-of-use, yet harnesses the power of PC without having to buy



CorpSoft

The Ultimate Software Reference Guide



Have you ever wished you had a guide covering all the software products you could want, or need to find out about? Of course you have, because with so much software available, computer professionals like yourself frequently need to have high quality information - FAST.

Corpdata have developed an inexpensive and convenient package for people who need information on available software choices. CorpSoft is designed like a book for intuitive ease-of-use, yet harnesses the power of your PC without having to buy expensive CD-ROM drives.

Over 18,000 variants of more than 8,500 products are packed into our ready to use software tool which installs itself on a corner of your hard drive for convenient help whenever you need it.

14 Chapters containing over 650 software categories covering vertical systems, applications software, system software and much more. From Mainframes to PC's and from Minis to Macs.

Contact information, publisher information, price guides, technical support numbers, and an extensive description for each product - even appropriate operating systems are covered.

On-line help, key word searches, fast product and publisher indexes, and even the capacity to store products of interest in your own "personal chapter".

Why pay extra for guides which promise so much yet deliver so much less? For a value packed £35 you get one of the fastest and easiest to use reference guides available anywhere - and you also have time to pay with our convenient method of ordering.


Use the appended form and receive your own copy tomorrow.

(29) Facsimile number 44

89F	OPEL	KADETT	1.3 GLS	SEDAN	AUTO	A/C	E/N	C/L	£2695	
88F	NISSAN	SUNNY	1.3 LS	HATCHBACK					£2000	
85G	FORD	SICRA	SDA	HATCHBACK					£995	
82X	FORD	ESCORT	VAN	35	PETROL				£850	
82Y	FORD	ESCORT	1.3	SDA	1060CC				£995	
95M	FORD	MONDEO	ESTATE	1.8i GLX	AUTO	A/C	E/N	C/L	£12495	
93G	MITSUBISHI	SHOGUN	LLB	2.5TD	A/C	ELECTRIC	PAKE	7 SEATS	EXTRAS	£19995
NEW AND LHD CARS AVAILABLE TO ORDER OFTEN IMMEDIATE DELIVERY										
BRITISH REGISTERED TAX'S PAID OR TAX FREE EXPORT										
NHTY NOT DISCUSS YOUR REQUIREMENTS: INSURANCE ARRANGED -										
EUROPEAN DELIVERY . SHIPPING THE WORLD OVER .										
<u>NEW ARRIVALS</u>										
92J	CITROEN	ZX	16i	AUXA	SDA	E/N	C/L	REMOTE	£4695	
90H	CHEVROLET	LUMINAR	MPV	7 SEATS	A/C	GLS	AUTO		£7995	
89F	MITSUBISHI	(Dodge RAIDER)	SHOGUN	SWB	V6 3.0L	A/C	MANUAL		£5995	
93K	VW	PASSAT	ESTATE	2000i 16V	A/C	CRUISE	ELECTRIC	PAKE	CATALYTIC	£8495

Roundabouts Garage, A515 Elmhurst, Lichfield, Staffs. WS13 8HE
Tel/Fax No. 01543 414307
A Division of Maxwell Charles, Lichfield, England.

(30) Facsimile number 46

 <p>Nottinghamshire County Council Construction and Design</p>	FACSIMILE TRANSMISSION	
	TO: <u>David Moor</u> 01153 486518 cc. <u>Andrew Kay</u> 01532 386720.	
NEWARK AREA OFFICE	FROM: <u>Steve Needham</u>	
TOTAL NUMBER OF PAGES TRANSMITTED INCLUDING THIS SHEET: <input type="text"/>		
TELEPHONE FOR QUERIES: (01636) 73625	FAX NO: (01636) 79531	

DATE:

SUBJECT: S38 Development Station Road Loundham

MESSAGE: I refer to recent discussions and confirm that the Highway Authority is prepared to accept granite setts in the maintenance margin adjacent to your property subject to the following:-

- 1/ The margin shall be 750mm wide and delineated for the full length by a raised row of setts or a 20mm step in level.
- 2/ The setts shall be uniform in type and appearance laid on a 25mm mortar bed on 150mm concrete foundation with pointed joints with a 1 in 40 fall to the carriageway. The setts shall be slightly higher than the kerb top for the pointing to be level with the kerb top to ensure satisfactory drainage.
- 3/ The works will need inspecting at all stages by ourselves - please contact Mr I Rowlett on 01636 73625 in this respect.
- 4/ The maintenance margin from the garage to the boundary with Main Street shall be treated in the same manner and curtailed at a position to be identified by Mr Rowlett.

(31) Facsimile number 47

14/05 '96 TI 11:57 FAX -358 55 3556 377 MIKKELI POLYTECHNIC

001



MIKKELI POLYTECHNIC
FACHHOCHSCHULE MIKKELI

To
An

DR STEIGER

Fax

NOTTINGHAM TRENT UNIVERSITY

From
Von

999 - 44 - 115 - 948 65 18

PÄIVI KAPAINEN - HEISKANEN

Message
Nachricht

Dear Dr. Steiger,
attached please find
the promised log data
sheet. Could you pls
have one filled in
for us. Thanks!

Best regards,

Päivi

Date
Datum

14.5.1996

Number of Pages
Anzahl der Seiten

1+2

MIKKELI POLYTECHNIC

Street address:
Patterilankatu 3

Tel. +358-55-366 61
Fax. +358-55 3556 377

(32) Facsimile number 48



SOUTHAMPTON
INSTITUTE

From: BREMA DE HOLLANDER
To: DE NASSER SHEIKAT
Fax No: 0115 9486518
Date: 30/6/96

Fax message of 2 page(s) will follow

3 pages in total

Fast FAX message:

MSc COMPUTING (SOFTWARE
ENGINEERING)

PROGRAMME / PART AS REQUESTED
FULL DOCUMENTATION BEING
SENT BY INTERNET TODAY

Please telephone BREMA DE HOLLANDER No. 01703 319578
if transmission is unclear-

East Park Terrace, Southampton SO14 0YN Telephone: (01703) 319000. Fax: (01703) 222259.

01703 319515

P.01

(33) Facsimile number 53



5400



NOTEBOOKS

NB5400T-CPU-HDD-QUAD

5400 notebook with:

10.4" TFT LCD display and
Quad-speed CD-ROM drive

£1,069.00

The above notebook with:

Intel Pentium® 120MHz CPU,

8 Megabytes of RAM and

540 Megabyte Hard Disk Drive

£1,349.00

Tel: (01952) 428827 Fax: (01952) 428800

(34) Facsimile number 55

P:01



D-Link

Free Prize Draw!

Orders placed on Friday
27th September, 1996
will be entered in our
free prize draw!

Call  the **D-Link**
Distributor

Tel: (01952) 428888

(35) Facsimile number 57

904 822 7161

P.01

FROM : Stetson University Bookstore

PHONE NO. : 904 822 7161

Sep. 27 1996 04:08PM P2

22nd September 1996

Dr. Aldabass,

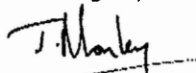
Ref - Final year project, B.Sc.(H) Computing Systems.

I discussed with you last week that I would like to direct my final year project in a Telecommunications field. As I have been on placement with AT&T in Switching Development, I would like to direct my career in this area.

I have been researching for the past month into "Object-Orientated software mechanisms for high speed network programming". I have concluded that the existing "Event Handler" mechanisms could be improved for delivering high bandwidth applications.

I will contact you during the first week of term to discuss in more detail the requirements for this project.

Kind regards,



James Mousley(Mr)

(36) Facsimile number 58

MAY 15 '96 14:27 FACULTE DES SCIENCES UN. MONCTON

P.1/1



Le 15 mai 1996



Reference: Word Recognition Using Multiple Independent
Features, by G. Raza, N. Sherkat, and R. J. Whitrow

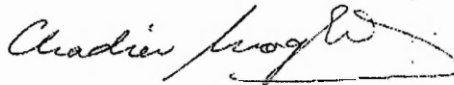
Dear G. Raza,

This is to confirm that your above mentioned paper has been accepted by the International programme committee for presentation at the NLP+IA 96.

This conference will be taking place in Moncton on June 4 to 6 at l'Hôtel Beauséjour.

I hope that this invitation will help you in obtaining the necessary visa and in fulfilling the formalities at your university.

Sincerely,



Dr. Chadia Moghrabi, professeure
Département d'informatique
Faculté des sciences
Université de Moncton
Moncton, NB E1A 3E9 CANADA

sr

Moncton
Nouveau-Brunswick
Canada E1A 3E9

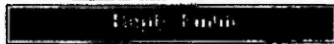
(37) Facsimile number 61



EUROTEAM

Director -
Rachelle Maxwell

Programme	
09.10	Arrival and Coffee
09.25	Introduction and Welcome by DICE
09.30	Update on the Fourth Framework Programme (FP4) Including: update on the four main activities and their relevance to industry.
10.15	The Future of EC R&D - The Fifth Framework Programme (FP5).
10.40	Coffee
11.00	Local Case study - The benefits to SMEs
11.20	SME support measures Including: Technology Stimulation Measures
11.40	Local Case Study - The consultants approach
12.00	Current Calls Including: Technology Transfer & Technology Validation, Main Programmes.
12.20	Sources of Help
12.40	Open discussion
13.05	Lunch
14.05	Open Forum: Representatives of various organisations will be available for one to one discussion sessions with delegates.
15.05	Close



Please reserve [] places at the Fourth Framework Programme Update seminar on October 17th 1996 - DICE, The Nottingham Trent University.

Name(s):	
Organisation:	
Address:	
Tel:	Fax:

Please send your reservation:
by fax to 0115 948 6568
by post to Rachelle Maxwell, The Djanogly Innovation Centre for Europe, The Nottingham Trent University, Burton Street, Nottingham NG1 4BU

Burton Street
Nottingham NG1 4BU
Tel. (0115) 9486845
Fax (0115) 9486568

01159486568

P. 01

(38) Facsimile number 62

12 September 1996

Dear Colleague



The Office of Science and Technology at the Department of Trade and Industry, has asked EUROTEAM to organise a seminar as part of its 1996 Roadshow. The seminar will update local business support organisations, including, Business Links, TECs, EICs, BICs, Chambers of Commerce and Local Councils as well as other intermediaries and multipliers of the latest developments in the European Commissions Fourth Framework Programme.

During the seminar speakers from the Office of Science and Technology and invited experts will brief participants on the current state of play. The speakers will also be available after the event, if delegates wish to discuss particular points in more detail.

As well as an update of the Fourth Framework Programme, the seminars will include an insight into the Fifth Framework Programme and detailed information relating to the current SME support measures provided by the European Commission.

The seminar which is free to invited organisations will take place in Nottingham at The Djanogly Innovation Centre for Europe, The Nottingham Trent University on October 17th.

Attached is a more detailed programme for the day and a reply form for you to reserve places at the seminar. I do hope you or a colleague will be able to attend. If you require any further information concerning the seminar then please do not hesitate to contact my office.

With regards

A handwritten signature in black ink that reads "Rachelle Maxwell".

Rachelle Maxwell
Director



EUROTEAM

Director -
Rachelle Maxwell

Burton Street
Nottingham NG1 4BU
Tel: (0115) 9486845
Fax: (0115) 9486568

01159486568

P. 02

(39) Facsimile number 66



Registered Number: 489727 England

For the Attention of Rick Evans

Our Ref: NOT15029

Nottingham Trent
University
Visual Identity Unit
Burton Street
Nottingham NG1 4BU

Date: 25/09/96

ESTIMATE

Dear Sirs,

Thank you for your recent enquiry, we are pleased to submit the following estimate.

ENGINEERING OPEN DAY LEAFLETS (REF: JB1104/7/96)

Size: A5
Description: 2pp leaflets printed in two colours both sides
Materials: Media Glazed 170gsm
Origination: Final film we hold, no amends.
Finishing: Trim
Delivery: Nottingham

Quantity:	3000	Price:	£125.00
Run On:	500		£5.00

We hope that our prices meet with your approval and look forward to hearing from you.

Yours faithfully,
Progressive Printers (Nottingham)

TONY CONCANNON

Prices quoted are based on information available at this time and may be subject to review on sight of final artwork/copy/film. This estimate is open for acceptance for up to 30 days from the date above unless otherwise stated or earlier withdrawn or modified or the Company agrees to later acceptance. Subject to the Conditions of Contract printed overleaf. VAT will be charged where applicable.

(40) Facsimile number 69

Custom Frames

Traditional and Contemporary Picture Framing

Second Floor
2 Stoney Street
The Lace Market
Nottingham NG1 1LG
Tel: 0115 956 5010
Fax: 0115 956 5011

MRS HIMSWORTH
COMPUTING DEPT
3rd FLOOR
NOTTINGHAM TRENT UNIVERSITY

DATE 8/8/96

Dear Mrs Himsworth,
Please find below prices for A1 size (841x594mm) Aluminium and Aluminium Foil frames as discussed previously. Prices are assessed on 30 frames.

Nielsen Profile 11	Frame only + Backing	23.50 each
Nielsen Profile 15	" " "	27.50 each

ALUMINIUM FOIL OVER WOOD

H00 00038 & H00 0002	Frame only + Backing	13.00 each
----------------------	----------------------	------------

2mm Clear Float Glass	to fit the above size	9.00 each
2mm Acrylic	" " "	8.00 each

TERMS

As we have no account set up at the moment, this would have to be a pro-forma transaction (C.O.D).

On any future orders we can offer a 25% discount off our current price list for "one off frames" to the University (15% to Students) and Special prices for volume/contract work.

PLEASE NOTE

We will be closing for our annual break from the 9th to the 24th of September 96.

We look forward to being of service to you in the near future.

Yours sincerely,



Alan Carlisle (Proprietor)

(41) Facsimile number 74

02/08 '96 FRI 11:50 FAX 01159770053

BIRCH PRINT

001



Birch Print

QUOTATION

No. 1452

Colour Printing
Business Forms
Computer Stationery
Packaging

Date: 2.8.96

For the attention of Nick Freestone.

Dear Sirs,

We thank you for your recent enquiry and have pleasure in submitting our quotation as follows:

Job Title: Leaflets - Faculty of Engineering.

Quantity: 2,000 with 500 run-on.

Size: A5

Description: 170gsm gloss art printed 2 colours on front and single colour on reverse.

Origination: You to supply artwork on disk we will play directly to film.

Delivery: 10 days.

Price: 2,000 @ £190.00. 500 run-on @ £29.00.

This quotation is subject to acceptance within 21 days. Please refer to the Quotation Number when ordering or revising this quote.

We trust this quotation meets with your approval and look forward to your valued order which will receive our best attention.

Kindly note our conditions of sale as printed on the reverse.

We remain,
Yours faithfully,

For Birch Print

Birch House
Southglade Business Park
Hucknall Road

M.J. Birch Limited

(42) Facsimile number 80

17/10/96 10:52 0115 9552201

NTU FAX UNIT

0001

ERIC POTTER CLARKSON

To: NTU, Department of Computing
Attention: Ian Allison
From: Phil Morris
Date: 17 October 1996
Our Ref: Final Year Project
Your Ref:
No. Pages: 3
(Inc. this one)
Our Fax No: (0115) 9552201
Your Fax No: (0115) 9486518

Ian,

Sorry for the delay in forwarding this to you, I am still having discussions at work to finalise the exact nature of the development work.

I have attached a preliminary specification for the project but would welcome any input from yourself on this matter. I am still not convinced that the title is correct, as I intend to develop a working system, but I do not want to move towards a purely HCI based project.

Please contact me if there are any problems.

Regards.

Phil Morris

CONFIDENTIALITY NOTE:

The information contained in this teletype message is legally privileged and confidential information intended only for the use of the individual or entity named above. If the reader of this message is not the intended recipient, you are hereby notified that any dissemination, distribution or copy of this teletype is strictly prohibited. If you have received this teletype in error, please immediately notify us by telephone: 0115 - 9552211.

(43) Facsimile number 81



KOLEJ BANDAR UTAMA

No.1, Persiaran Utama, Bandar Utama Damansara, 47800 Petaling Jaya, Selangor Darul Ehsan, Malaysia. Tel:03-7173200 Fax: 03-7172703

19 October 1996

Mr John Smith
Department of Computing
Faculty of Engineering and Computing
The Nottingham Trent University
Burton Street
Nottingham NG1 4BU
United Kingdom

Dear Mr Smith

Please be informed that I have sent the following to you by courier today :

- i) Systems Methodologies (B) End Test question paper and Marking Scheme
- ii) Systems Software (OS) End Test question paper and Marking Scheme

for the BTEC HND in Computer Studies Year 2, Semester 1 end test.

Please acknowledge the receipt of the above.

Thank you

With kind regards
KOLEJ BANDAR UTAMA

Christine Lee (Miss)
Course Leader



SEE HOY CHAN HOLDINGS GROUP

03 7172733

P. 01

(44) Facsimile number 82

MANAGEMENT ACCOUNTS

FAX NUMBER 0115 948 6553



URGENT FAX MESSAGE

FOR THE ATTENTION OF : Bob WHITROW.

FAX NUMBER : 6578

COUNTRY IF OUTSIDE UK : -

DATE : 24-SEP-96

FROM : John BILLINGTON.

NUMBER OF PAGES : 2
(including this)

MESSAGE : ACEF figures - taken straight from HEFCE data.

If there are any problems with this FAX message, please contact
The Nottingham Trent University, Management Accounts
Tel: 0115 948 6423

max

001-002

MAN ACCOUNTS

01159486553

24 09 96 00:21

(45) Facsimile number 88

Memorandum

From	Professor John Stancer	Date	9 June 1997
		File Ref	JDS/EAD
To	All Deans	Ext	5546/6803



European Funding for TCSs

It has been pointed out to me that my memo of 4 June 1997 about European funding could read as if I am saying that Professor Thompson's meeting on 12 June is of no consequence - or not needed. This was not my intention. Those involved with TCSs should obviously make every effort to attend Professor Thompson's meeting. My intention was to allay any possible concerns amongst other colleagues - ESF co-ordinators - that if they missed the 12 June meeting they would be at a disadvantage.

A handwritten signature in cursive script, appearing to read 'John Stancer'.

Professor J D Stancer
Pro Vice Chancellor & Senior Dean

(46) Facsimile number 89

NAME: PAUL CONROY.

DATE OF BIRTH: 28/4/75.

COURSE AND DATES: BSC COMPUTER STUDIES 1994 - present.
NOTTINGHAM TRINITY UNIVERSITY.

POSITION APPLIED FOR: TEMP WORK.

Please confirm

1. Dates at your establishment: From
To

2. Was the applicant to the best of your knowledge:

Honest YES/NO

Trustworthy YES/NO

Conscientious YES/NO

A regular attender YES/NO

3. Do you know of any reason why we should not employ the candidate? Please comment

4. Please give your general impression of the candidate's abilities and character.

Thank you for completing the reference request. You have our assurance that the information will remain confidential and will not be entered onto any computerised file.

Signed:

Name:

Position:

(47) Facsimile number 90

FAX COVER SHEET

DATE: 10/6/97

TO: MAGGIE.

FROM: SHEREE WILSON;

MESSAGE: Please could you complete the
reference for Paul Conroy
and return it to me by fax
ASAP. MANY THANKS.

NUMBER OF PAGES INCLUDING THIS ONE: (2)

IF THIS TRANSMISSION IS INCOMPLETE PLEASE
CONTACT THE ABOVE NAME AND NUMBER.

(48) Facsimile number 92

Content: Dr Evett Internet, E-mail systems, Artificial Intelligence
Dr Smith Office systems, Human Computer Interaction
David Peeks Document Mgt., Data Capture, Document Image Processing
Graham Knight Chair of session - will introduce and run question/answer session

Presented by: Dr Lindsay Evett, Dept. of Computing, The Nottingham Trent University
Tel: 0115 9418418 x6018
Dr Pauline Smith, Dept. of Computing, The Nottingham Trent University
Tel: 0115 9418418 x 2701

Mr David Peeks, Diptec Computer Systems
Tel: 0115 946 4773
Fax: 0115 946 4719

Mr Graham Knight, Internet Development & Design,
Tel: 0115 962 5007 Work
Tel: 0115 962 5604 Home

Venue: Nottingham Law School Ltd, Belgrave Centre, Chaucer St., Nottingham NG1 5LP
Lecture Theatre, Floor B and Board Room for Reception

Contacts: Room booking/equipment/catering Allan Nelder
0115 948 6871 x 2360
AV equipment & reprographics Jeff Morley
0115 948 6871 x 4152

I hope you have all the information you need for the presentation. I will be in the office on x6873 until Thursday evening. After Thursday you can contact

Carol Parkinson President, BPW Nottingham 0115 914 5940 Hme
Karen Hornby Membership Secretary 01773 607385 Hme
01623 426220 Wk
Isabel Hopkins Divisional President, BPW 01773 873143 Hme
01773 602432 Wk

Please join Carol and Karen for the buffet at 1830, you can then sit in on the Committee business prior to the presentation, or wait in the Board Room if you would prefer.

Thanks again, hope all goes well

(49) Facsimile number 94

2. Would you re-employ this person? (If 'no' give details)

3. Reason for Leaving:

4. Is there any more information Elizabeth Michael Associates should know about this person before employing their services?

Signed:

Position:

Company:

Tel Number:

0602 418793

TOTAL P.05
P.05

(50) Facsimile number 95

DATE: 12 June 97

FAX NUMBER: 948 6518

TO: Eva, please pass to Mr Steve King

FROM: Heidi Peplow

NO OF PAGES TO FOLLOW: Two

MESSAGE: Re: REFERENCE REQUEST FOR PAUL ROBERTS

Further to our telephone conversation of today, I enclose a reference table for the above applicant.

I would be grateful if you could fax this back to me as soon as possible, to enable us to find suitable employment for the said applicant.

Many thanks for your help.

Kind Regards

Heidi

HEIDI PEPLow

*** URGENT ***

Recruitment Specialists

J. Hyde (Managing Director) J. Hyde (Director and Company Secretary)

Company Reg. No. 2914870

VAT Reg. No. 588 1198 18

Licence No. M1570

0400 410797

P 03

Appendix D. Published papers

The work carried out during the present research has been published by the author in a number of research papers. References of all published are given below. Full copies of the published papers are also enclosed in this Appendix.

[Raza et al 97]

Raza, G., Hennig, A., Sherkat, N. and Whitrow, R. J.: (1997) "Recognition of facsimile messages using a database of robust features", International Conference for Document Analysis and Recognition, Vol. 1, pp. 444-448.

[Hennig et al 97]

Hennig, A., Raza, G., Sherkat, N. and Whitrow, R. J.: (1997) "Detecting a document's skew: A simple stochastic approach", Eleventh Canadian Conference on Computer Vision, Signal and Image Processing, and Pattern Recognition, pp. 97-102.

[Raza et al 97]

Raza, G., Hennig, A., Sherkat, N. and Whitrow, R. J.: (1997) "Applying feature based word recognition approach to screen text recognition", Eleventh Canadian Conference on Computer Vision, Signal and Image Processing, and Pattern Recognition, pp. 103-107.

[Jobbins et al 96]

Jobbins, A. C., Raza, G., Evett, L. J. and Sherkat, N.: (1996) "Postprocessing for OCR: Correcting errors using semantic relations", in L. J. Evett and T. G. Rose (Eds.) Language Engineering for Document

Analysis and Recognition (LEDAR), AISB96 Workshop, Sussex, England, pp. 154-161.

[Raza et al 96]

Raza, G., Sherkat, N. and Whitrow, R. J.: (1996) "Word recognition using multiple independent features", International Conference on Natural Language Processing and Industrial Applications, Vol. 2, pp. 233-236.

[Raza et al 96]

Raza, G., Sherkat, N. and Whitrow, R. J.: (1996) "Recognition of poor quality words without segmentation", IEEE International Conference on Systems, Man and Cybernetics, Vol. 1, pp. 64-69.

Recognition of facsimile documents using a database of robust features

G. Raza, A. Hennig, N. Sherkat, R. J. Whitrow
Department of Computing, The Nottingham Trent University,
Burton Street, Nottingham NG1 4BU, UK

{ghr,amr,ns,rjw}@doc.ntu.ac.uk

Abstract

A method for the recognition of poor quality documents containing touching characters is presented. The method is based on extraction of independent and robust features of each object of a sample word, where objects consist of single letters or of several touching ones. Thus avoiding letter segmentation the method eliminates errors frequently introduced in segmentation based approaches. Features are attributed by their position and extent in order to facilitate discrimination between different classes of objects. A method for automatic construction of a comprehensive database is presented. From a given dictionary every possible letter combination is obtained and the images of the artificially touching letters created. These images are subjected to noise and their features extracted. For recognition, alternatives for each object are found based on the database. Object alternatives are then combined into valid word alternatives using lexicon lookup. It has been observed that the developed method is effective for the recognition of poor quality documents.

Keywords: Word recognition, Feature extraction, OCR, Segmentation, Automatic database development

1 Introduction

Optical Character Recognition (OCR) has been a topic of research for several decades and as a result, much work has been done in this area[1]. This work has led to the development of many algorithms and systems. The performance of these algorithms and systems is good provided the quality of the document is good, i.e. for documents containing characters which are well formed and separated from their neighbourhoods[2]. However, these systems tend to perform worse on poor quality documents such as facsimile messages, low quality prints, photocopies, etc. There is still a great gap between the capabilities of humans and machines of recognising text. Humans can recognise very poor quality documents. In

order to narrow this gap, if not bridge it, we have a long way to go and hence it is still a challenge to develop a recognition system which can obtain high accuracy rates irrespective of the font type, size and the quality of the document.

Poor quality documents often have a large number of touching or broken characters. As most existing OCR systems and methods rely on the segmentation of touching characters, correct segmentation points are vital for high recognition rates. They can, however, be difficult to obtain in poor quality documents. As a result of erroneous segmentation, two or more characters may be grouped into one character (e.g. 'rn' into 'm' or 'cl' into 'd'), or one character may be segmented into two characters (e.g. 'U' into 'll' or 'h' into 'li'). Several methods have been proposed for segmenting words into their character components. Kahan et al.[3] discussed the method of segmenting touching characters. Tsujimoto and Asada[4] constructed a decision tree for resolving ambiguities in segmenting touching characters. Casey and Nagy[5] proposed a recursive segmentation algorithm to segment touching characters. Liang et al.[6] proposed a dynamic recursive segmentation algorithm for the segmentation of touching characters. It has been observed that half of the errors in character recognition are due to incorrect segmentation [7]. The methods of segmenting touching characters "appears to be ad hoc in nature, and thus not particularly applicable to general" [8], poor quality documents in particular.

Looking at the difficulties faced by the segmentation step in character recognition, algorithms have been proposed for whole word recognition. They are based on an analysis of the word's shape and on computation of neighbourhoods of decisions for each word [9][10][11].

In [12] we described a method for the recognition of poor quality documents using multiple independent features based on objects. Objects are extracted as connected components, which might consist of single letters or several touching ones. Diacritical marks and some parts of some punctuation marks are extracted as

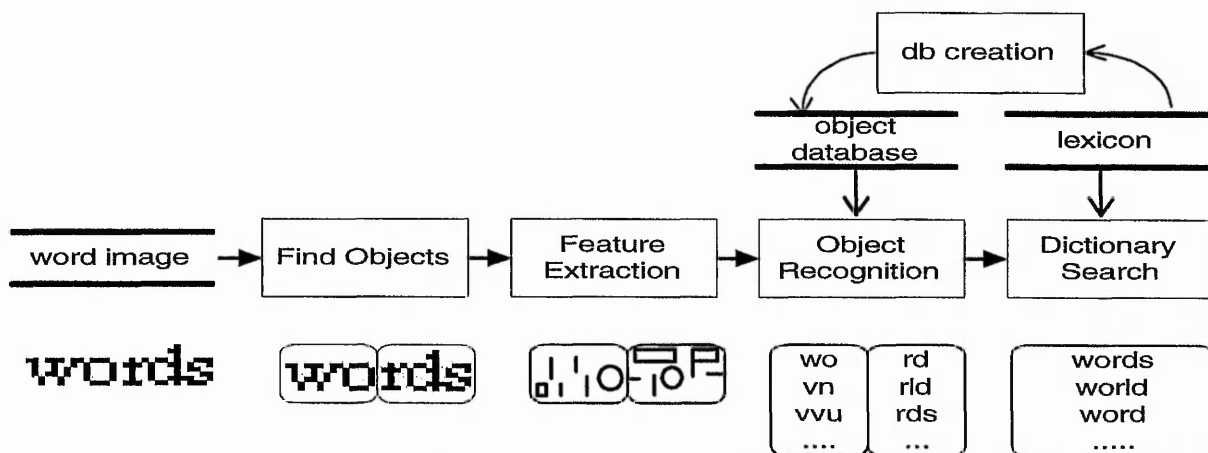


Fig. 1: Outline of the word recognition algorithm

separate objects. This exploits external segmentation wherever it is clearly available without attempting to segment objects of touching letters.

The system — as outlined in Fig 1 — first identifies words in the document image. Objects are then found within and the features are extracted. Different alternatives of each object are found by comparing the features found in the sample object with a database of ideal features, ordered by a similarity measure. Object alternatives are combined to form different words using dictionary lookup ranked by the overall similarity. If more than one word are ranked top higher level linguistics information such as language syntax and semantics can be used to identify the correct word [13][14][15][16].

In this paper, we present improvements to the original approach, improving the recognition of poor quality documents. The restriction to maximal three touching characters per object has been removed. Feature extraction has been improved. Additional features are now used, each of them attributed by position relative to the object and the feature's extent (e.g. its width and height). The database of objects is now created automatically, including objects consisting of artificially touching letters. Two methods are used to artificially join character combinations, these are (a) moving the characters physically closer and (b) introducing noise. The similarity measure between database objects and document objects is now based on the cost of an edit operation rather than simple differences in the number of the respective feature.

2 Feature Extraction

In [12] we have described a system which extracts different independent features of every object of the sample words. The features extracted were: the zone an object is located in (either upper, middle, lower and full),

holes (upper, middle and lower) found in the object, vertical bars, side opens (upper, lower, left and right) and corner opens (top-left, top-right, bottom-left and bottom-right). Our experiments showed that these features are reasonably tolerant to noise such as broken, touching and degraded characters. Furthermore, these features are expected to be consistent in characters of different fonts types and sizes.

However, the method simply determined the presence of absence of a particular feature. It did not employ any other details of the feature such as position, length and width. For example, the first three objects in Table 1 have one hole and are therefore ambiguous with respect to holes. Using position and extent, however, discriminates them easily. As both position and extent of the features are normalised in respect to the object's bounding box, the features remain independent of font size.

In the current research, the existing methods for

Table 1: Old features versus new features

Object	old Feature	new Feature
no	1 hole	
non	1 hole	
onn	1 hole	
a b g ky	zone	
m	3 vertical bars	
u	upper open (lower, left, right)	
b	top-left corner TL (TR, BL, BR)	
E	<i>no horizontal bars</i>	
L	<i>no gravity feature</i>	
i	<i>no dot feature</i>	

extracting features have been improved and new methods for the extraction of additional features have been devised, i.e. horizontal bars, centre of gravity and dots of letters like 'i' and 'j'. For a summary of old and new features refer to Table 1.

3 Automatic Database Creation

Database creation is — after feature definition and extraction — the most important step of this method, as all objects of one or more touching letters must be contained in the database. In [12] a database of approx. 4000 entries was created manually, according to the ideal features and to those found in some example facsimile messages. Due to human limitations, the database (with up to three touching characters) was both incomplete and not always accurate.

The improved method allows automatic database creation and removed the restriction of three letters. The aim of the database is to store feature strings of real life objects. As this would again require extensive manual labour in order to identify and label touching letters in a large number of documents, a different approach has been chosen. The processes that can lead to touching letters has been modelled rather than observing their effects. These processes were (a) modified character kerning to move characters physically closer and (b) noise introduced during scanning or copying. While construction of a single letter database is trivial, letter combination that might touch to form objects have to be extracted from the dictionary.

The first method modifies the character kerning to move letters closer to each other. We call the degree of kerning *undercut* (as used for the PostScript command 'ashow'), a negative undercut moves letters closer together. We apply three different undercut values to a given prototype document constructed from the letter combinations found in the dictionary. The effect of

different undercuts in some two letter combinations is shown in Table 2. The features of these touching combinations are extracted and stored in the database along with their known characters.

The second method models noise that is added to the original image in two steps. First, randomly selected pixels are blackened if they are adjacent to another black pixel (i.e. randomised growth). Second, the image is blurred and thresholded, i.e. if more than a certain percentage of pixels in a $N \times N$ neighbourhood are black, the central pixel is blackened as well (e.g. 30% of a 3×3 neighbourhood, Table 2). This results in objects growing towards each other until they finally bridge gaps between different parts of one single object or between different objects. The first effect can modify the features of a character (e.g. the lower open of an 'h' might become a hole). The second causes previously separated letters to form a single object of touching letters. This is similar to the effects that can be observed in poor quality documents.

4 Finding Object Alternatives

After extracting the features of an object, the object's height and width are used to estimate the approximate number of characters in the object, N_{app} :

$$N_{app} = 1 + \frac{width}{height} \quad (1)$$

Only database entries that describe objects of N_{app} touching characters are considered for recognition of the object. Equation (1) has been found to be accurate for most objects but will fail for certain objects. For example, 'm' might be estimated to be two characters long while 'fi' might yield an N_{app} of 1. Therefore we also consider database entries whose lengths fulfil

$$nr \text{ of letters} \in [N_{app} - 1, N_{app} + 1] \quad (2)$$

The similarity between a database entry and the object under observation is expressed as the total cost of transforming one into the other using a modified form of the edit-distance. Insertion or deletion of one feature has been defined to have a cost of 1. The difference in the number of features of any one type (e.g. one versus three holes) is totalled over all feature types using (3). If the counts are identical, the cost for transforming one feature into another of the same type is obtained and totalled for all every pair of features (4). The transformation cost of two given features is derived from the distance of their respective origins and the differences in their extents. Its value has to be less than 2, the cost for deletion and reinsertion of the feature. The sum of the transformation cost of every feature type and the cost of insertions or

Table 2: Applying kerning and noise

	Kerning		Noise			
			noise	blur		
0.0	rn	cl	—	—	rn	cl
-0.1	rn	cl	10%	20%	rn	cl
-0.2	rn	cl	20%	17%	rn	cl
-0.5	rn	cl	30%	10%	rn	cl
-1.0	m	cl	—	20%	m	cl
-2.0	m	d	—	10%	m	cl

deletions gives the overall cost, describing the closeness of match (5).

$$Cost_{ins/del} = \sum_{f \in featureTypes} |nrFeatures_f^{database} - nrFeatures_f^{sample}| \quad (3)$$

$$Cost_{trans,f} = \sum_{i \in features\ of\ Type\ f} Dist(feature_i^{database}, feature_i^{sample}) \quad (4)$$

$; Dist(\dots) \in [0,2]$

$$Cost_{total} = Cost_{ins/del} + \sum_{f \in featureTypes} Cost_{trans,f} \quad (5)$$

The cost of transforming the sample feature string into a database feature string is obtained for calculated each of the database feature strings fulfilling (2). The best matches (i.e. those with the lowest cost) are maintained in a sorted table, keeping the overall winner at the top. These tables of object alternatives are obtained for every object in the word.

5 Building Words

From the minimum and maximum lengths of the alternatives of each constituting object, the range in length of the word can be derived. Words from the dictionary that fall within that range are selected and attempted to be constructed from the object alternatives. This construction employs a combination of binary search within one table and recursive search across different tables. A dictionary word that can be constructed from tables of alternatives represents a possible solution. Its total cost is computed as the sum of the object's costs. Solutions are stored in a word table ordered by their overall cost. The solution with lowest cost is used as the result of the recognition process. If there are more than one solutions with similar low costs, higher level linguistic information may be used to identify the correct word [15]. We can thus further improve the performance of the recognition system.

6 Experimental Results

In order to evaluate the performance of the developed system, 29 real-life facsimile messages were used. These facsimile messages contained text of different fonts in various size and attributes.

Printouts of the facsimile messages were scanned using a HP Scanjet Plus scanner at 300x300dpi. As our method targets poor quality facsimiles, only words of lower quality have been used for evaluation. Determination of 'poorer' quality words followed a rather pragmatic definition: words are deemed of poor quality if commercial software fails to recognise them. From the original facsimile messages, 972 words of poor quality

were thus marked and the dictionary originally containing 4000 words extended if necessary (see Fig. 2 for some examples).

The developed recognizer gave varied recognition rates for all facsimile messages ranging from 40% to 100%, compared to 21% to 100% using the old method on a much smaller evaluation set. Recognition rates on the small set compared as 37% (old) to 47.9% (new) The overall recognition rate on the full set was 48% top choice (64% top 10 choices). The main reasons for the variations for different facsimile messages are differences in the number of touching characters and underlined words. Low improvement is observed for facsimile messages having many words that contain more touching characters and underlined words. The overall improvement against the old method is due to using improved features, cost evaluation as well as the improved database creation method.

We acknowledge the fact that the developed recognizer uses a dictionary to reduce the number of possible alternatives, which aids recognition significantly. The results of the character based commercial software, however, has been observed to be insufficient as input to an subsequent dictionary search (i.e. spell-checker) in most of the cases.

7 Conclusion and Future Work

An improved word recognition algorithm for the recognition of poor quality word images is presented. This method avoids segmentation of touching characters and hence bypasses errors incurred during the segmentation stage of a typical OCR system. It tries to identify each object of the word based on its features. A method for automatic creation of the feature database is also presented. For every objects candidates are selected from the database based on an improved measure of features similarity. This system is capable of recognising poor quality words of different fonts and sizes. This method is robust and particularly suitable for the recognition of poor quality documents which require a small dictionary.

The work is in progress and future work involves modification and improvement of feature extraction methods and the extraction of additional features. It is also envisaged to test this approach for the recognition of other languages e.g. Arabic, Urdu (Pakistani), which are written in cursive script and are difficult to segment.

8 References

- [1] V. K. Govindan, "Character recognition-a review", Pattern Recognition, 23, No. 7, (1990), pp. 671-683.

- [2] H. S. Baird, "Feature identification for hybrid structural/statistical pattern classification", *Computer Vision, Graphics, and Image Processing*, Vol. 42, No. 3, pp. 318-333.
- [3] S. Kahan, T. Pavlidis and H. S. Baird, "On the recognition of printed characters of any font and size", *IEEE Trans. on Pattern Analysis and Machine Intelligence PAMI*, Vol. 9, No. 2, 1987, pp. 274-287.
- [4] S. Tsujimoto and H. Asada, "Resolving ambiguity in segmenting touching characters", *1st Int. Conf. on Document Analysis and Recognition*, Saint-Malo, France, (1991), pp. 701-709.
- [5] R. G. Casey and G. Nagy, "Recursive segmentation and classification of composite character patterns", *Proc. 6th Int. Conf. Pattern recognition*, Munich, Germany, (1982), pp. 1023-1026.
- [6] S. Liang, M. Ahmadi and M. Shridhar, "Segmentation of touching characters in printed document recognition", *IEBEE*, (1993), pp. 569-572.
- [7] C. H. Chen, J. L. DeCurtins, "Word recognition in a segmentation-free approach to OCR", *IEBEE*, (1993), pp. 573-576.
- [8] D. G. Elliman and I. T. Lancaster, "A review on segmentation and contextual analysis techniques for text recognition", *Pattern Recognition*, 23, No. 3/4, (1990), pp. 337-346.
- [9] T. K. Ho, J. J. Hull and S. H. Srihari, "A word shape analysis approach to lexicon based word recognition", *Pattern Recognition Letters*, Vol. 13, (1992), pp. 821-826.
- [10] J. J. Hull, "Hypothesis generation in a computational model for visual word recognition", *IEEE Expert*, Vol. 1, No. 3, (1986), pp. 63-70.
- [11] C. Fang and J. J. Hull, "A hypothesis testing approach to word recognition using an A* search algorithm", *IEBEE*, (1995), pp. 360-363.
- [12] G. Raza, N. Sherkat and R. J. Whitrow, "Word recognition using multiple independent features", *International Conference on Natural language Processing and Industrial Applications*, Vol. 2, (1996), pp. 233-236.
- [13] F. G. Keenan and L. J. Evett, "Applying syntactic information to text recognition", in L. J. Evett and T. G. Rose (Eds.) *Computational Linguistics for speech and Handwriting Recognition*, AISB Workshop, (1994).
- [14] J. J. Hull, "Feature selection and language syntax in text recognition. in *From Pixels to Features*", J. C. Simon (editor), North Holland, (1989), pp. 249-260.
- [15] A. C. Jobbins, G. Raza, L. J. Evett and N. Sherkat, "Postprocessing for OCR: Correcting errors using semantic relations", in L. J. Evett and T. G. Rose (Eds.) *Language Engineering for Document Analysis and Recognition (LEDAR)*, AISB96 Workshop, Sussex, England, (1996), pp. 154-161.
- [16] T. G. Rose and L. J. Evett, "The use of context in cursive script recognition", *Machine Vision and Applications*, Vol. 8, (1995), pp. 241-248.

paper	and	Marking	Scheme	and
Marking	Scheme	Semester	end	test
Thank	you	With	kind	regards
KOLEJ	BANDAR	UTAMA	Christine	Lee
Miss	Course	Leader	SEE	HOY
CHAN	HOLDINGS	GROUP	MANAGEMENT	ACCOUNTS
FAX	NUMBER	URGENT	FAX	MESSAGE

Fig. 2: Example poor quality words from real-life facsimile messages

Detecting a Document's Skew: A Simple Stochastic Approach

A. Hennig, G. Raza, N. Sherkat, R. J. Whitrow
Department of Computing, The Nottingham Trent University,
Burton Street, Nottingham NG1 4BU, UK
{amr,ghr,ns,rjw}@doc.ntu.ac.uk

Abstract

A simple stochastic approach to the detection of the skew angle of a scanned document is proposed in this paper. The method first estimates the undirected skew within the range of -90° to $+90^\circ$. I-dots and full-stops are then used to determine whether the document has been scanned upside down, yielding the directed skew. Only a fraction of the information available in the document is exploited, without assumptions as to the general layout of the page. The method has been shown to be robust and accurate for a variety of documents containing both hand-written and printed text. The average error of the directed skew angle was observed to be 0.05° for a set of synthetic documents. For facsimile images, however, the upside-down detection failed in 9.6% of the documents examined.

1 Introduction

The detection of the overall skew of an electronic document is a vital early step in its analysis and recognition. In the majority of documents, lines of text are oriented horizontally. In a rotated document, skew angle is defined as the difference between the dominant orientation of the text lines and the horizontal. When a document is scanned, skew might be introduced in several ways. A relatively small skew might be added by imprecise feeding of the paper into the scanner or fax machine. A document in landscape orientation might have to be scanned in portrait orientation if the size of the scanner is insufficient. The resulting image is then rotated by $+90^\circ$ or -90° , depending on the preferences of the person that feeds the document. The document might even be inserted upside down, resulting in an additional skew of 180° . If a paper document is photocopied before the scanning, these effects might accumulate further.

In this paper, the angle of the 'directed skew' (denoted by $\varphi, \varphi \in (-180^\circ, +180^\circ]$) describes the total rotation that has been applied to the original (right way up) document. The 'undirected skew' (denoted with $\bar{\varphi}, \bar{\varphi} \in (-90^\circ, +90^\circ]$),

however, assumes that the document has been scanned with a normal orientation. Most documents have a uniform skew. If they are moved during scanning or copying, or if the original contains curved or non-parallel lines of text, the skew angle is not constant throughout the document. This problem has been addressed in work described in [6][3] and is beyond the scope of this paper.

Various methods have been applied to the detection of a uniform skew. Projection profiles of connected components have been used by [2]. The Hough transform has been widely used, e.g. by [4], [1] and [9], even though it is computationally expensive and usually requires an assumption about the interval that contains the skew. A multi-layer perceptron is used for cursive handwriting in [5], whereas [8] applies a least square linear regression to reference points found on the page. Most of these methods, however, assume a right-way-up document, or even require that lines of text can be identified correctly.

The method presented in this paper uses a simple stochastic method to estimate the overall undirected skew of the document. A second step aims to detect whether the document has been scanned upside down, thus obtaining the directed skew. These steps are described in the following sections, followed by experimental results and concluding remarks.

2 Detection of the undirected skew angle

The undirected skew detection algorithm is based on the fact that elements of one straight line of text correlate with each other. The directions between elements of the same line of text distribute around the skew, whereas the connection to other lines can appear in almost any other direction. The histogram obtained from the angles of all possible connections hence shows a maximum at the desired skew angle. Elements might be black pixels or the centres of regions of connected black pixels. The example in Fig. 1a shows the connections from the centre of the region that forms the letter 'n' to the other regions of the page. Fig. 1b shows the resulting histogram with a clear maximum at the document's skew.

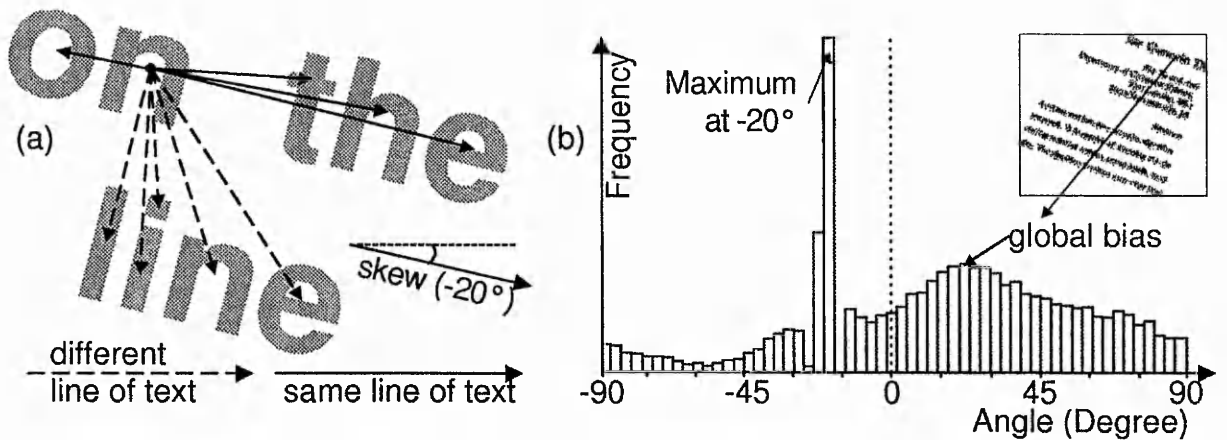


Fig. 1: Connections between black regions: a) Regions belonging to the same line of text appear under the same angle, the regions of other lines can appear at any other angle. b) The histogram shows a clear maximum around the undirected skew of the document.

This method, however, requires an excessive computational effort if all possible connections between regions or even pixels are considered. Furthermore, the histogram is biased by the overall arrangement of the elements. Elements in the corners of the document, for example, 'see' the majority of the remaining ones in direction of the opposite corner. This effect can be observed in the secondary maximum around 25° , derived from the document shown in the inset of Fig. 1b. The extraction of regions of connected pixels can be computationally expensive. If the document contains many lines such as tables, forms or text written on lined paper, only a few large regions can be detected. The centre point then becomes a poor representation of the region and the method becomes inappropriate.

To overcome these problems, the proposed method uses pixels instead of regions. The black pixels Q in a chosen distance r from a selected pixel P are considered (Fig. 2a). The angles the points Q appear at are computed and the histogram is updated accordingly. If this step is repeated for a number of different pixels P , the resulting histogram shows a clear global maximum (Fig. 2b) at the skew angle. Pixels P are selected randomly in order to avoid the exhaustive observation of all elements in the document without favouring a particular area. This allows the computational cost to be reduced dramatically without a substantial loss in accuracy. It also avoids the need of assumptions about the general layout of the document, e.g. the assumption that the lowest line on the page is a stable feature in [1].

Secondary maxima in the histogram are caused by the text lines above and below the line containing P , particularly if the lines are equally spaced. With increasing radii, these secondary maxima move closer to the global one. In order to emphasise the global maximum and to reduce the height of the secondary maxima,

histograms of different radii are superimposed (Fig. 2c). The choice of the allowed radii is limited by the size of the letters in the document and the total size of the document. If r becomes too small, both P and Q might belong to the same letter (to or immediately adjacent ones) and no clear maximum will be observed in the histogram. If to large a radius is chosen, only pairs of that distance can be used, and observation is effectively reduced to the corners and margins of the document.

In order to simplify the computation further, the city-block distance is used instead of the Euclidean one. The shapes described by the points Q then degenerate from a circle around P to a square rotated through 45° . The cells of the histogram are denoted by h_i ; $i = 0(1)H - 1$; H being the size of the histogram.

The complete algorithm for the detection of the skew angle is:

- 1 initialise $h_i = 0$; $i = 0(1)H - 1$
- 2 for n randomly chosen black pixels P
- 3 choose a radius $r \in [r_{min}, r_{max}]$ randomly
- 4 for every pixel Q_i ; $i = 0(1)4r - 1$ in
 city-block distance r from P
- 5 if (Q_i is a black pixel)
- 6 increment the histogram:
 $h_i^* = h_j + 1$; $j = \text{round}\left(i * \frac{H-1}{4r-1}\right)$ (1)
- 7 done
- 8 done
- 9 smooth the histogram by the weighted sum of the s neighbours
 $h_i^* = \sum_{j=i-s}^{i+s} \left(1 - \frac{|i-j|}{s}\right) * h_{j \bmod H}$; $i = 0(1)H - 1$ (2)
- 10 use the position of the global maximum in the histogram to compute the undirected skew angle

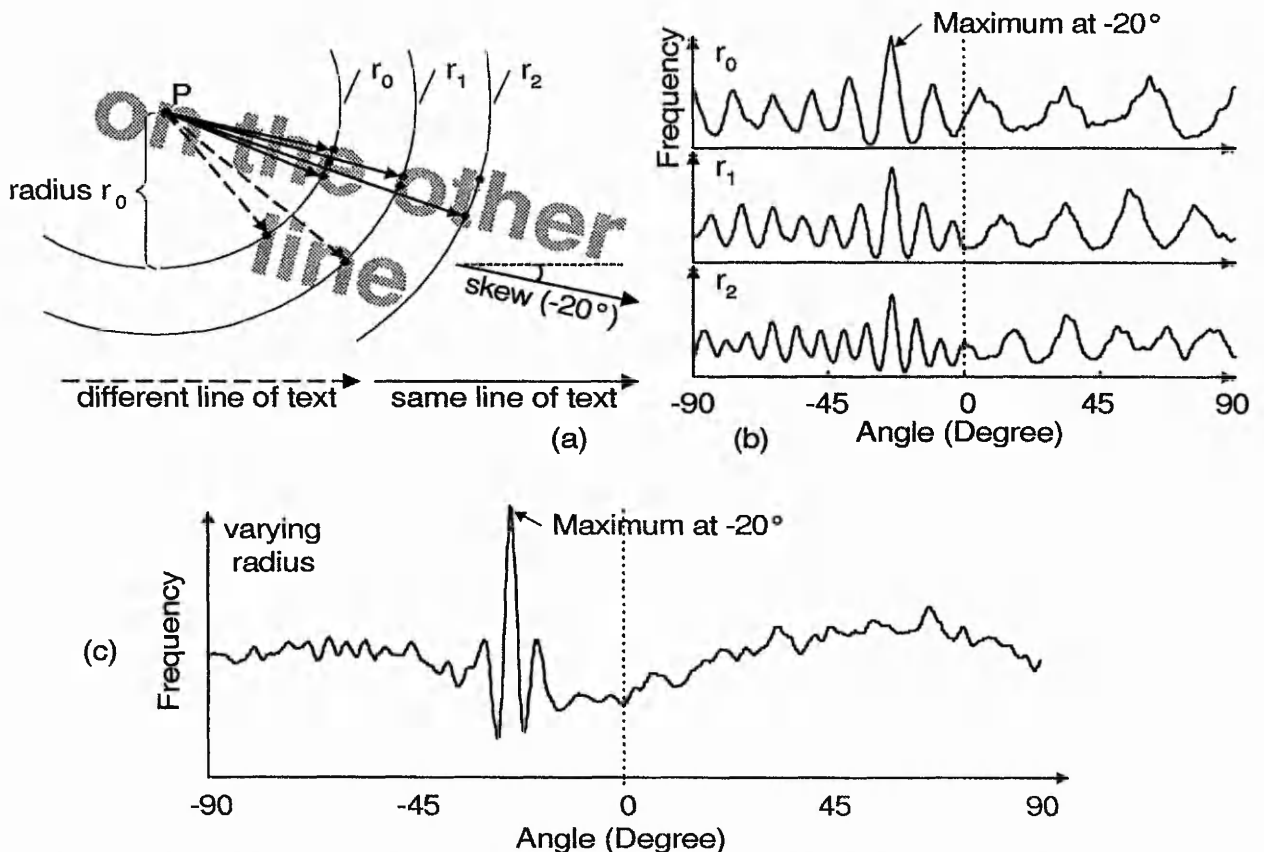


Fig. 2: Observing pixels correlation: a) the black pixels in a chosen distance r from a selected pixel P are investigated. b) The histogram of the angles of the lines they form shows a maximum at the overall skew angle. The positions of the secondary maxima vary with different radii. c) Superimposing the histograms of various radii further emphasises the global maximum.

3 Detection of upside-down documents

In order to detect whether a text document has been scanned upside-down or not, two simple properties of text are used. Firstly, the dots in the letters *i* and *j* are written above their body and are usually closer to their body than to the line of text above. Secondly, the full-stops terminating a sentence or abbreviation are closer to the preceding letter than to the following one. The first property depends on the way *i*'s and *j*'s are written, i.e. on the roman alphabet itself, while the second exploits the correlation between consecutive characters, i.e. language dependent conventions.

To detect the dots in the document, regions are detected. A set of simple constraints is then used to identify the regions that are most likely to be dot-shaped, namely: aspect ratio, area to bounding box ratio, circumference to area ratio, concavity to convex hull area ratio (see Fig. 3 and Table 1).

A dot is ideally a filled circle. The aspect ratio of the bounding rectangle is therefore expected to be close to 1.

The ratio of the area of a circle and the area of its bounding square is approximated by the number of black pixels within the detected region and the area of the bounding box. The ratio between the square root of the region's area and the length of the region's boundary (i.e. the circumference of an ideal circle) has to be within a given interval. The length of the region's boundary line is obtained by a simple edge following algorithm (Fig. 3a). The region's area is expressed as the number of black pixels therein. As an ideal circle is convex, the total area of the concavities found in the region should be zero. The concavities' area is expressed as the total of all areas that have to be added to the polygonal region in order to form a convex hull (Fig. 3b).

The above conditions do not exclude regions that contain a white area, such as simple circles or the letters 'e' or 'a' in poor quality printing. Rather than testing every pixel within the region boundary, only the following ones are verified: the centre of the bounding box should be the centre of the dot and must therefore be a black pixel. The

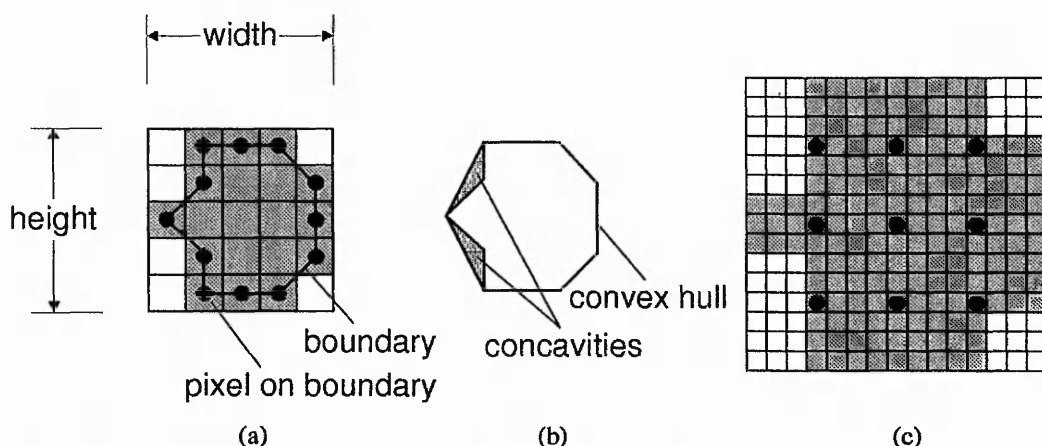


Fig. 3: Identifying dot-shaped regions: a) height/width ratio; black pixels/area of bounding box; boundary length/region area ratio; b) area of concavities/convex hull area ratio c) no white pixels are allowed inside the area (region has been scaled by factor 3 for ease of demonstration)

eight points halfway between the centre and the edges of the box are also tested to check if they are black (Fig. 3c).

Of the remaining list of candidates, the top and bottom 1% are discarded in order to restrict the influence of noise and dot-shaped pictorial elements of the page (e.g. 'bullet points'). In some documents, clouds of well-shaped dots can be observed, for example in scatter diagrams or as a part of raster graphics. As these dots are usually not part of a text line, they are ignored. Their identification uses a simple proximity criterion: If two or more dot candidates are located close to each other, all of them are discarded. The proximity threshold has been experimentally determined to be twice the average of the dot's diagonal extent.

In printed documents, the remaining candidates are mostly i- and j-dots and full-stops, with the exception of

those contained in punctuation marks (e.g. exclamation mark, question mark, semicolon and colon). For the i-dots, one can define the directions in which pixels of the letter's body are expected to be "seen" from the centre of the dot. If the document is in its original orientation, the directions are within $A_{up} = [\bar{\varphi} - 90^\circ - \alpha, \bar{\varphi} - 90^\circ + \alpha]$, the parameter α controlling the width of the aperture A_{up} . Under the hypothesis of an upside-down document the aperture is $A_{down} = [\bar{\varphi} + 90^\circ - \alpha, \bar{\varphi} + 90^\circ + \alpha]$. The body of the letter immediately preceding the full-stop is expected to appear within $A_{up}^* = [\bar{\varphi} + 135^\circ - \alpha, \bar{\varphi} + 135^\circ + \alpha]$ and $A_{down}^* = [\bar{\varphi} - 45^\circ - \alpha, \bar{\varphi} - 45^\circ + \alpha]$ without any further pixels in A_{up} and A_{down} .

The pixels at a common distance from the centre of the

Table 1: Criteria to identify dot-shaped regions of connected black pixels.

Criterion	acceptancy range	ideal value (circle)	used value	value observed in Fig. 3
aspect ratio of bounding box	$\frac{height}{width} \in [\gamma_{lw}^-, \frac{1}{\gamma_{lw}^+}]$	$\gamma_{lw} = 1$	$\gamma_{lw} = 0.75$	$\frac{5}{5} = 1.0$
black pixels to bounding box area	$\frac{blackPixels}{boundingArea} \in [\gamma_{bb}^-, \gamma_{bb}^+]$	$\gamma_{bb}^- = \gamma_{bb}^+ = \frac{\pi}{4} \approx 0.79$	$\gamma_{bb}^- = 0.6$ $\gamma_{bb}^+ = 1.0$	$\frac{19}{25} \approx 0.76$
circumference to area ratio	$\frac{\sqrt{blackPixels}}{circumPixels} \in [\gamma_{cc}^-, \gamma_{cc}^+]$	$\gamma_{cc}^- = \gamma_{cc}^+ = \frac{1}{2\sqrt{\pi}} \approx 0.28$	$\gamma_{cc}^- = 0.2$ $\gamma_{cc}^+ = 1.0$	$\frac{\sqrt{19}}{12} \approx 0.36$
area of concavities	$\frac{concavityArea}{convexHullArea} \in [\gamma_{cv}^-, \gamma_{cv}^+]$	$\gamma_{cv}^- = \gamma_{cv}^+ = 0$	$\gamma_{cv}^- = 0.0$ $\gamma_{cv}^+ = 0.3$	$\frac{1}{13} \approx 0.077$
white pixels inside region	no white pixels allowed	test all pixels in the circle	test selected pixels only	all 9 pixels are black

dot candidate's bounding box are observed. Thrice the length of the diagonal of the bounding box has been used as the radius. This value has been found to be sufficient in most cases (see Fig. 4b for illustration). Every black pixel on the resulting circle (or diamond using the city-block distance) that falls inside one of the four apertures supports the respective hypothesis. This support is accumulated in the variables up , $down$, up^* and $down^*$ respectively. If sufficient pixels are found in A_{up} or A_{down} , the dot is interpreted as an i-dot otherwise it is assumed to be a full-stop. If the count of the winning hypothesis is significantly higher, for example if

$$\frac{|up - down|}{\max(up, down)} > 15\%, \quad (3)$$

the dot as a whole supports the respective hypothesis. Every dot in the document is examined separately and the overall support for each of the hypotheses is calculated. The one with the overall majority is accepted and the directed skew angle is computed as

$$\varphi = \begin{cases} \overline{\varphi} & ; \text{if right way up} \\ \overline{\varphi} + 180^\circ \text{ mod } 360^\circ & ; \text{if upside-down} \end{cases} \quad (4)$$

This process is demonstrated on two dot candidates in the example of Fig. 4a. Around the i-dot (P_1), three black pixels were found in the aperture A_{up} , while none were found within A_{down} . P_1 is therefore interpreted as a i-dot, voting upside-up. The black pixels neighbouring the full-stop (P_2) are not found within the apertures for i-dots. Under the full-stop interpretation, however, a majority of five for the upside-up hypothesis can be observed. The

overall vote therefore totals to 2:0 in favour of the upside-up hypothesis.

4 Experimental Results

In order to evaluate the proposed method, 50 real-life facsimile documents were scanned at 300x300 dpi into bilevel images. They include letters, tables, advertisements and forms with both printed and hand-written text. Text sizes vary from 8 pt up to approximately 40 pt in various fonts. They were scanned in the correct orientation and then rotated electronically through two different angles (20° and 200°), to obtain an upside-up as well as an upside-down version of the document.

The undirected skew was correctly detected for 90% of the documents. Two documents were composed of two areas of different skew, e.g. an unskewed form with an area for hand-written remarks, written with considerable skew. The method yielded the average of these two skew angles. Two faxed advertisements in portrait orientation contained large portions of reverse text printed along the left hand corner, resulting in a skew error of 90° , effectively interpreting them as landscape documents. One document contained more noise than text, which caused the method to fail completely.

The upside-down detection failed in 9.6% of the test-documents. One document was entirely written in capital letters and contained no dots except a single colon. Many of the hand-written dots did not follow the above constraints. The method therefore often failed in cases

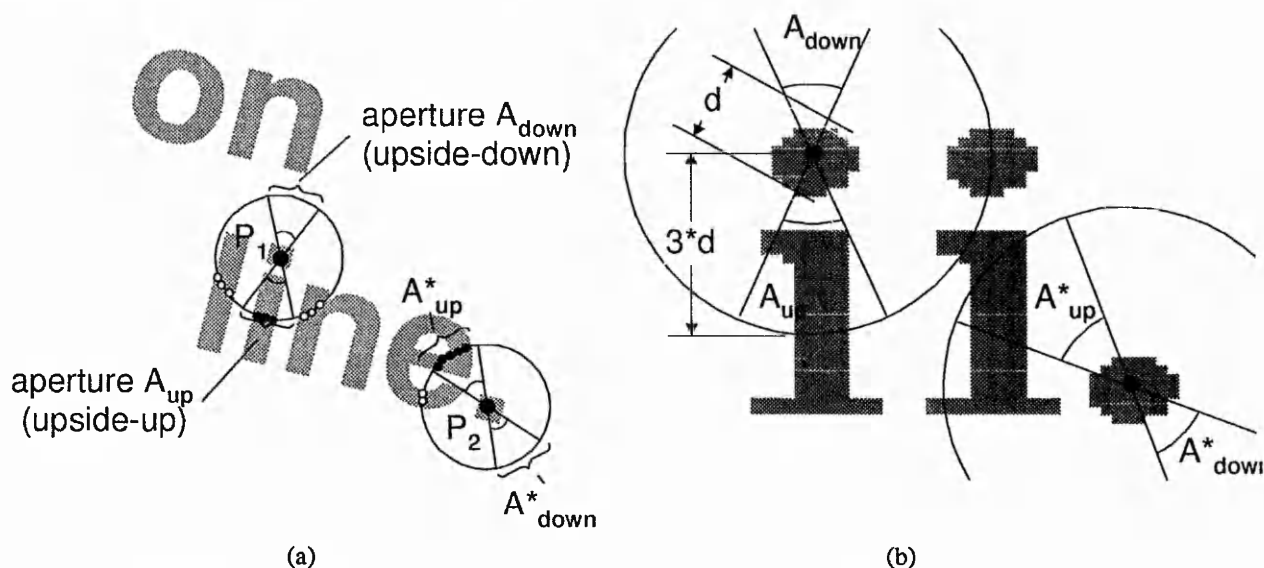


Fig. 4: Upside-down Detection: Testing i-dots and full-stops. a): black pixels marked as filled circles are within one of the hypotheses' apertures and therefore in support. Unfilled circles mark pixels that satisfy the distance constraint but do not appear in any of the apertures. b) the apertures in skewless script.

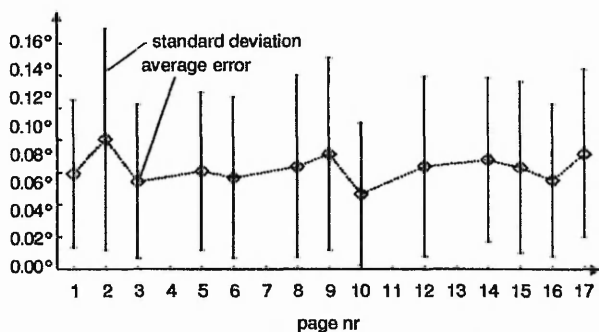


Fig. 5: Average and standard deviation of the skew detection error

were the majority of text was hand-written.

Comparison of the observed accuracy with the results reported for other methods is difficult, as different sets of test images have been used that were acquired under different circumstances. Furthermore, it is not always possible to manually determine the accurate skew angle as a reference. Following the method proposed by Yu[9], the manuscript of that paper has been converted into its electronic version without intermediate steps for printing and scanning. Using the ghostscript software package, 17 images have been produced at a 200x200 dpi resolution. These synthetic documents have then been rotated by -180 to +180 degrees in steps of 9°. The radius r was allowed to vary between 600 and 1000 pixels (i.e. 7.6cm to 15.2cm); $n=1000$ black pixels were observed per page; the opening of the aperture was set to $\alpha=30^\circ$. The average error of the undirected skew angles is shown in Fig. 5 for each page together with the standard deviation. The overall average error was observed to be 0.05° (compared with 0.1° given by Yu) with no upside-down detection errors. The maximum error was 0.28°. Pages 4,7,11 and 13 contained large portions of graphics and were therefore excluded from the evaluation by Yu. The presented method often failed for these pages. The upside-down detection, however, failed only for the pages 7 and 13. On page 4, a page with a medium sized graphics, the undirected skew was still detected correctly (i.e. with less than 0.5° error) in over 75% of the cases after increasing the value of n to 2000.

5 Conclusion

An alternative method for the detection of the skew of an electronic document has been presented. The method is capable of dealing with a variety of document types at any skew angle. No assumption about the layout of the

document has to be made nor has the detectable skew to be restricted to a given interval. The accuracy of the method has been demonstrated to be similar if not superior to alternative approaches, even though the complexity of the algorithm is considerably lower. The proposed method for the detection of upside-down documents, however, rests on the assumption of detectable dots, which does not hold for all types of documents, especially hand-written ones. Alternative methods have to be developed to address this problem, exploiting additional properties of text such as left-alignment of paragraphs or the relationship between the number of ascenders and descenders in the text.

6 References

- [1] Amin, A. and R. Shiu, "New Skew Detection and Correction Algorithms", in Proceedings of the Fifth International Workshop on Frontiers in Handwriting Recognition IWFHR-5, p. 251, Essex, England, September 1996.
- [2] Baird, H. S., B. Yu, and A. K. Jain, "A Robust and Fast Skew Detection Algorithm for Generic Documents", Pattern Recognition, vol. 29, pp. 1599-1629, 1996.
- [3] Hennig, A., N. Sherkat, and R. J. Whitrow, "Zone Estimation for Multiple Lines of Handwriting Using Approximating Spline Functions", in Proceedings of the Fifth International Workshop on Frontiers in Handwriting Recognition IWFHR-5, p. 325, Essex, England, September 1996.
- [4] Hinds, S. T., J. L. Fisher, and D. P. d'Amato, "A Document Skew Detection Method Using Run-Length Encoding and the Hough Transform", in International Conference on Pattern Recognition, vol. 1, pp. 464-468, Atlantic City, NJ, 1990.
- [5] Rondel, N. and G. Burel, "Cooperation of Multi-Layer Perceptrons for the Estimation of Skew Angle in Text Document Images", in Proceedings of the Third International Conference on Document Analysis and Recognition ICDAR95, p. 1141, IEEE, August 1995.
- [6] Smith, R., "A Simple and Efficient Skew Detection Algorithm via Text Row Accumulation", in Proceedings of the Third International Conference on Document Analysis and Recognition ICDAR95, p. 1145, IEEE, August 1995.
- [9] Yu, B. and A. K. Jain, "A Robust and Fast Skew Detection Algorithm for Generic Documents", Pattern Recognition, vol. 29, pp. 1599-1629, 1996. (Manuscript from ftp://ftp.cps.msu.edu/pub/prip/binyu/orient.ps.gz)
- [8] Yu, C. L., Y. Y. Tang, and C. Y. Suen, "Document Skew Detection Based on the Fractal and Least Squares Method", in Proceedings of the Third International Conference on Document Analysis and Recognition ICDAR95, p. 1149, IEEE, August 1995.

Applying Feature Based Word Recognition Approach to Screen Text Recognition

G. Raza, A. Hennig, N. Sherkat, R. J. Whitrow
Department of Computing, The Nottingham Trent University,
Burton Street, Nottingham NG1 4BU, UK
{ghr,amr,ns,rjw}@doc.ntu.ac.uk

Abstract

In earlier work we described a feature based word recognition method for the recognition of poor quality words taken from fax messages. Various independent and robust features are used to identify alternatives of every object of the word without attempting to segment touching objects. Later, a lexical lookup method is used to verify the alternatives. In this paper, the developed method is applied to screen text of different fonts and sizes in order to observe its performance on screen image. It has been observed that the developed method is capable of recognising screen text of varying fonts and sizes whilst avoiding segmentation of touching characters.

1 Introduction

Character segmentation is a key step in most conventional Optical Character Recognition (OCR) systems. This segmentation based approach is possible for good quality prints, where characters are clearly separated from their neighbours. However, such an approach is unsuitable for the documents containing characters touching their neighbouring characters. The performance of existing OCR systems is not acceptable for such documents. Documents of this type come from many different sources such as facsimile messages, low quality prints, photocopies etc. Furthermore, existing OCR systems are unable to deal with screen images[1][2].

A feature based word recognition method [3] for the recognition of poor quality word images has been developed. For this paper, this method has been modified and applied to screen text of different fonts and sizes. The results obtained show the ability of the method to cope with screen images.

In this method, different independent features of each object (which could be a single character or several touching ones) of a word are extracted from the input sample word. Different alternatives for each object are

found by comparing the features with a database of ideal features for each object. These alternatives are ranked according to the number of features matched. The alternative with the most features matched is considered to be the most likely candidate.

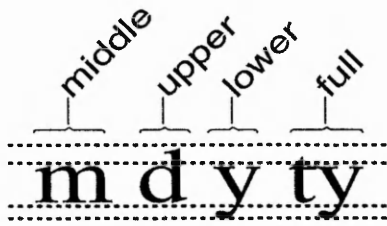
These alternatives are combined to form different words using a dictionary lookup step. These words are also ranked according to the total number of features matched. The word with most features matched is deemed to be the recognised sample word. If there is more than one word, then higher level linguistic information, such as language syntax and semantics, can be used to identify the correct word [4][5][6].

2 Feature Extraction

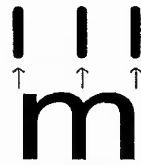
Feature extraction is an important stage in the recognition of characters particularly in the case of poor quality documents. In order to recognise a character, different features are extracted, which will (hopefully) exhibit the distinctive characteristics of the character[7]. Ideally, the features should enable the recognizer to discriminate correctly between distinct classes of characters.

In the literature different commonly used feature extraction methods have been described, e.g. global features, distribution of points, geometric and topological features, linguistic descriptions, use of context and fuzzy sets. There is no general technique for the design of feature extraction which utilises the designer's a priori knowledge of the recognition problem. Geometrical and topological features are commonly used by human beings in the recognition of patterns because such features can easily be detected by the human eye.

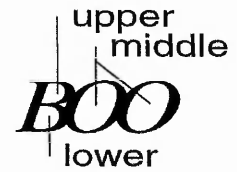
In the current research, various independent features of every object of the sample words are extracted and compared against the ideal form of the object (see Figure 1). Features are: the zones in which the objects are found (middle zone only, upper or lower zones and full, i.e. both upper and lower zone), vertical bars, holes (in the upper,



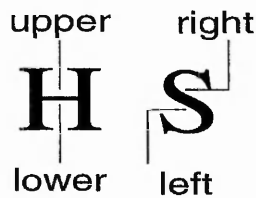
a) zones



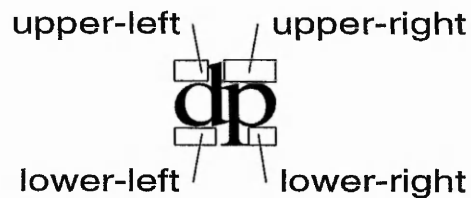
b) vertical bars



c) holes



d) opens



e) corners

Fig. 1: Features used in the recognition system

middle or lower part of the object), and openings in any of the eight principal directions (i.e. upper, left, lower right side opens as well as upper left, lower left, lower right, upper right corners)

These features have been selected in order to achieve the following goals:

(a) We are primarily dealing with poor quality documents. Therefore, consideration has been given to define features, which are generally not affected by the image quality. As this is not always possible, at least the majority of features are designed to be unaffected by additional or missing parts of the object.



Fig. 2: Objects width missing or additional parts

In Figure 2, the first letter has an unwanted extra part in its top half. This extra part leads to converting top side open feature of this object to the upper hole. But this change has not affected the other features of this object, e.g. bottom side open, left side open, right side open and vertical bars etc. Similarly, in Figure 2, some part in the upper half of the second object is missing. Therefore, instead of upper hole of that object, a top side open feature

may now be extracted. Nevertheless other features of this object can be extracted correctly and are sufficient for its classification.

(b) We are dealing with multi-size documents i.e. sample text may be of any size from 6 point to 48 point size. Therefore attempts have been made to find features, which are generally the same for different sizes of a particular object. As an example feature, we consider upper holes as present in the letter 'A'. This upper hole (as well as the lower open) should be found in letters 'A' of any size. Similar applies to the upper and lower open of the letter 'H', as shown in Table 1.

	10pts	12pts	18pts	24pts
'H'	H	H	H	H
'A'	A	A	A	A

Table 1: Letters with different sizes having same features

(c) We are also dealing with multi-font documents. Therefore, attempts have been made to find features, which remain unchanged for several fonts. For example, the upper open feature in the letter 'U' should be present different fonts, e.g. Helvetica, Courier, Arial and Times Roman etc. Similar applies to the lower hole of the letter 'a', as can be seen in Table 2.

	Courier	Helvetica	Times Roman
'a'	a	a	a
'U'	U	U	U

Table 2: Letters with different fonts having same features

Our experiments have shown that these features are reasonable tolerant to noise such as broken, touching and degraded characters.

3 Database Development

The presented approach aims to recognise each object within word as a whole, without trying to segment touching characters. Hence the proposed method requires a database of single letter features (a-z, A-Z), as well as a database of touching letters

Development of a single letter database is trivial. In order to develop two letter and three letter databases, all possible combinations of two letters and three letters in the words of a 4k dictionary were found. Later, all these combinations are stored in a file along with their features. Hence as a whole, we have used three different kinds of databases namely database 1 containing single letters (a, b, c, etc.), database 2 containing two letter combinations (ab, ac, al, ol, etc.) and database 3 containing three letter combinations (aba, ack, arn, etc.) along with their features. The ideal features of each object are currently defined manually. Work into automatic creation of the database is currently being carried out.

4 Algorithm Description

The algorithm for word recognition is illustrated in Figure 3.

The image of a document is first separated into lines.

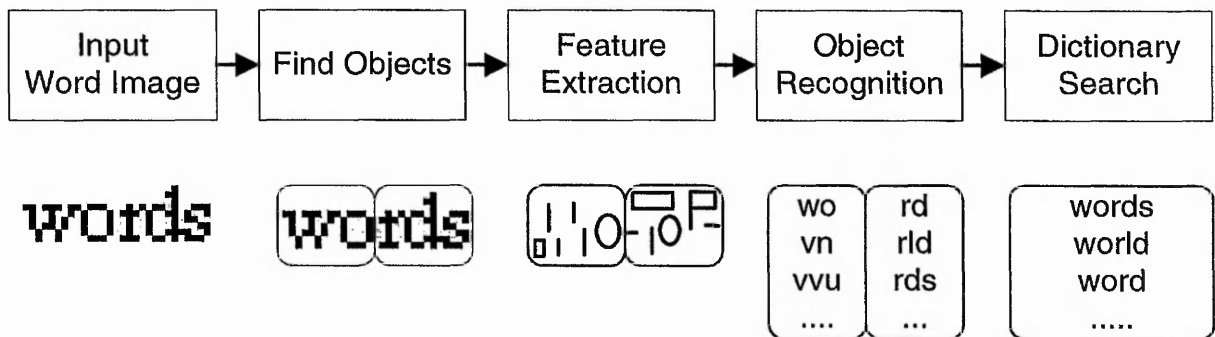


Fig. 3: Word recognition algorithm: Data flow and example.

The word images are then identified within a given line. Each word image is passed on to the object separator, where it is separated into different objects. Features of every object of the word are extracted next. Comparing the features with the database yields a list of alternative interpretations for each object. Words from a given dictionary are then sought that can be built from the object alternative lists

The two major parts of the recognition system are described in more detail below, namely finding object alternatives and searching for valid words

4.1 Finding Object Alternatives

Having found the features of an object, its height and width is calculated. Then, the minimum (L_{min}) and maximum (L_{max}) number of letters in the object are estimated, using the following heuristics.

$$L_{min} = \frac{height}{width} + 1, \quad L_{max} = \frac{height}{width} + 2 \quad (EQ1)$$

We then compare the features of this object with the corresponding features of database objects containing objects of length L_{min} and L_{max} . For every feature f the matching number (M_n) is found using the following equation.

$$M_f = 1 - \left| \frac{features_{db} - features_{sample}}{features_{db}} \right| \quad (EQ2)$$

The sum of all M_f yields the total matches T_m of the sample object with database object. This database object is recorded in a table of object alternatives together with its T_m . The table is kept sorted with the best alternative on top and restricted in size to a given maximum, currently 100 entries.

The above procedure is repeated for every object in the word. The alternatives of every object are stored in a separate table of object alternatives.

4.2 Search Dictionary Words

Object alternative may represent single characters or a combinations of characters. For each table, we find the minimum and maximum number of characters per object. We use these figures to calculate the longest and shortest possible words, thus giving us the range $[W_{min}, W_{max}]$ of the length of words that may be formed using these tables of alternatives, i.e.

$$W_{min} = \sum_{i=1}^{nrObjects} \text{shortest object length}_i \quad (\text{EQ3})$$

$$W_{max} = \sum_{i=1}^{nrObjects} \text{longest object length}_i$$

For each dictionary word, whose word length is within $[W_{min}, W_{max}]$, it is determined whether this word can be constructed from the tables of object alternatives using a recursive search. If the word can be constructed, the total number of matching features for the word (W_m) is calculated as the total of number of matching features (T_m) for each object alternative used to form the word.

$$W_m = \sum_{i=1}^{nrObjects} T_{m,i} \quad (\text{EQ4})$$

The word formed from the tables of alternatives is stored in a further table along with its total number of matching features. This table is also kept sorted by the total number of matching features maintaining the best word alternative at its top.

The word(s) with the highest total number of matching features in the resulting list are considered to be the recognised word. If several words have identical high W_m higher level linguistic information such as language syntax and semantics may be used to identify the correct word. We can thus further improve the performance of the recognition system.

5 Experimental Results

Nine different sample documents consisting of screen text were created in order to evaluate the performance of the recognizer. Each document contained the same 315 words, one of which was not present in the dictionary. These documents were written in three different fonts (Helvetica, Courier and Times Roman) in three different font sizes (12, 14 and 16 points).

These sample documents were initially written and displayed in 'Word Perfect' and their image captured into 'Paint Brush' in order to obtain their bitmap. The 'Graphics Workshop' program was then used to convert the bitmaps into TIFF format as required by the recognizer. The results obtained from screen text are given in Table 3.

The recognition behaviour of the screen text in the Helvetica font in 12, 14 & 16 point size appears strange, as the highest recognition rate has been observed for the smallest i.e. 12 point size text and the least recognition rate is noticed for 14 point size text. However, in other two cases, i.e. the Courier and Times Roman screen text tests, the relationship between point size and recognition rate is consistent. This clearly means that recognition rate increases with an increase in the point size.

Sample No	Font	Point size	Total nr of Words	Invalid Words	Words recognised	Recognition Rate	Average Recognition Rate
1	Helvetica	12	315	1	232	73.9%	53.0%
2		14	315	1	89	28.3%	
3		16	315	1	178	56.7%	
4	Courier	12	315	1	303	96.5%	97.8%
5		14	315	1	304	96.8%	
6		16	315	1	314	100%	
7	Times Roman	12	315	1	138	45.0%	66.3%
8		14	315	1	215	68.5%	
9		16	315	1	268	85.3%	

Table 3: Screen text recognition results

6 Discussion

The main difficulty appeared in the case of Helvetica font text documents where no linear relationship between recognition rate and the text point size was noticed, see Figure 3a. The reason behind this observation could be a severe limitation in extracting a variety of character features, including top left side open feature (in the letter 'd'), top right side open feature (in b and h), bottom left side open feature (q), bottom right side open feature (p). Those characters were almost without ascenders and descenders, all characters therefore seemed to be in middle zone. This rendered the zone-feature virtually useless.

However similar to printed text (Courier and Times Roman fonts) recognition, screen text in Courier and Times Roman fonts (see Figure 3b and 3c) were recognised according to the human capabilities. The highest recognition rate was observed for the largest point size screen text and the recognition rate kept on decreasing with reducing the text point size. This is not surprising, considering that the size of a character determines the efficiency of the features extraction. More accurate feature extraction might therefore yield better result even for smaller characters.

7 Conclusion and Future Work

A word recognition algorithm for the recognition of screen word images is presented. The method has been applied to screen text of different fonts and sizes. The results obtained using this method are encouraging. This method avoids segmentation of touching characters and hence bypasses errors incurred during the segmentation stage of a typical OCR system. It tries to identify each object of the word based on its features. This method is capable of recognising poor quality words of different fonts and sizes. The method is robust and particularly suited for the recognition of poor quality documents

which need a small dictionary.

The work described in this paper is currently in progress. Modifications involve the modification and improvement of the method, in order to enable it to cope with a larger variety of fonts and sizes and higher degrees of noise in the image. Investigations are carried out into additional features, improved matching of database and sample features, investigation into automatic creation of the databases used as well as the introduction of artificial noise during databases creation.

8 References

- [1] N. Sherkat, R. J. Whitrow, K. Pugh and S. Harness, Fast icon and character recognition for providing universal access to a WIMP environment for the blind, *Studies in Health Technology and Informatics*, IOS Press, Amsterdam, (1993), pp. 19-23.
- [2] S. Harness, K. Pugh, N. Sherkat and R. J. Whitrow, Enabling the use of windows environment by the blind partially sighted, *IEE Colloquium*, London, (1993)
- [3] G. Raza, N. Sherkat and R. J. Whitrow, Word recognition using multiple independent features, *International Conference on Natural language Processing and Industrial Applications*, Vol. 2, (1996), pp. 233-236.
- [4] F. G. Keenan and L. J. Evett, Applying syntactic information to text recognition, in L. J. Evett and T. G. Rose (Eds.) *Computational Linguistics for speech and Handwriting Recognition*, AISB Workshop, 1994
- [5] A. C. Jobbins, G. Raza, L. J. Evett and N. Sherkat, Postprocessing for OCR: Correcting errors using semantic relations. in L. J. Evett and T. G. Rose (Eds.) *Language Engineering for Document Analysis and Recognition (LEDAR)*, AISB96 Workshop, Sussex, England, 1996, pp. 154-161.
- [6] T. G. Rose and L. J. Evett, The use of context in cursive script recognition, *Machine Vision and Applications*, Vol. 8, 1995, pp. 241-248.
- [7] C. Y. Suen, Distinctive features in automatic recognition of handprinted characters, *Signal Process*, Vol. 4, (1982), pp. 193-207.

POSTPROCESSING FOR OCR: CORRECTING ERRORS USING SEMANTIC RELATIONS

A. C. Jobbins, G. Raza, L.J. Evett¹ & N. Sherkat

Department of Computing, Nottingham Trent University
Burton Street, Nottingham NG1 4BU, England
e-mail: lje@doc.ntu.ac.uk

Abstract

Semantic relations between words can be used to aid selection from alternative candidate words, output from an Optical Character Recognition (OCR) system, in order to improve the overall recognition rate. One method of automatically identifying the semantic relations between words is by using an existing knowledge source, such as Roget's Thesaurus. A technique has been developed which exploits the lexical organisation of the thesaurus and identifies semantic relations. The development of this technique is outlined and the results from its application to OCR output are presented and discussed.

1. Introduction

Most work in OCR has been concerned with physical pattern recognition. The output of an OCR system consists of recognised words. In some cases there are alternative candidate words for a particular word position. In these cases, higher-level linguistic information can be used to help determine which is the correct word.

It would be expected that many words within a text would be related to the subject area(s) of that text and therefore would be related in meaning to each other. When an OCR system produces alternative candidate words at the same word position semantic information can be applied to determine which of these alternative words is most likely to be correct. For example, consider the following phrase which is output from an OCR system:

... many employers assume that women in general have lower income needs ...

incise

There are two word alternatives given at the tenth word position (*income* and *incise*). In this example, the word *income* is related to the words *employers* and *needs*; *incise* has no obvious relationship to other words in the phrase. Semantic information can be used to bias words for recognition. However, to apply semantic information to the output of an OCR system some source of such information is required.

1. To whom correspondence should be addressed.

2. Automatic Identification of Semantic Relations

To automatically identify semantic relationships between words an existing electronic lexical knowledge source can be used. For example, Chodorow has used on-line dictionaries [1], Rose et al. utilised text corpora [2] and Amsler used lexical knowledge-bases [3]. A source of lexical information about semantic relations that has so far not been exploited in depth is the electronic version of Roget's Thesaurus. The thesaurus contains explicit links between words, unlike dictionaries and corpora, and is publicly available. It is a well used and well established source of information about semantic associations between words.

2.1 Roget's Thesaurus

The third edition electronic version of Roget's Thesaurus is composed of 990 sequentially numbered and named categories. There is a hierarchical structure both above and below this category level. There are two structure levels above the category level and under each of the 990 categories there are groups of words that are associated with the category heading given. The words under the categories are grouped under five possible grammatical classifications: noun, verb, adjective, adverb and preposition. These classifications are further subdivided into more closely related groups of words. Some groups of words have cross-references associated with them that point to other closely related groups of words. Figure 1 gives an example of an extract within category 373 of the Thesaurus. The cross-references are given by a numerical reference to the category number followed by the title given in brackets:

```
H00373.03.03.04092.00.00.%H Female  
P00373.03.03.04093.01.00.%P N.  
100373.03.03.04094.02.00.%T female,feminine gender,she,her,-ess;  
femineity,feminality,muliebrity;femininity,feminineness,the eternal  
feminine;womanhood 134 (adulthood);womanliness,girliness;
```

Figure 1: Roget's Thesaurus Category Extract

The thesaurus contains a collection of words that are grouped together according to their similarity of meaning. Those words grouped together have a semantic relationship with each other and this information can be used to identify semantic relations between words. For example, a semantic relationship between two words could be assumed if they occurred within the same category in the thesaurus.

The work of Sedelow and Sedelow supports the use of Roget's Thesaurus, where they claimed it to be an adequate representation of human knowledge and of English semantic space [4]. They considered the issue of multi-locality of words in the thesaurus and the disambiguation of homographs by the application of a general mathematical model of thesauri. They demonstrated that it is possible to develop algorithms that can elicit semantic structures from the thesaurus and from manual experimentation tested the semantic organisation. From these results they concluded:

"...any assertions that the *Thesaurus* is a poor representation of English semantic organization would be ill-founded and, given the depth of analysis, would have to be regarded as counterfactual."

3 Developing a Post-Processing Technique

3.1 Thesaural Connections

The application of the thesaurus for the identification of semantic relations between words requires a means of determining what constitutes a valid semantic connection between two words. For example, given words w^1 and w^2 how could the lexical organisation of the thesaurus be exploited to establish whether a semantic relation $\{w^1, w^2\}$ exists between them? Morris and Hirst identified five types of thesaural relations between words based on the index entries of Roget's Thesaurus [5]. In the current work, four types of possible connections between words in the thesaurus were identified by considering the actual thesaural entries. This ensured the inclusion of all words located in the thesaurus; those words that form part of a multi-word thesaurus entry may not be represented in an index entry. The connections between pairs of words that have been identified are as follows:

(1) **Same category connection** is defined as a pair of words both occurring under the same category. Figure 2 gives an example of this connection type.

word [1]: *river*

word [2]: *tributary*

Figure 2: Same Category Connection

The words would be considered to be semantically related because they were found within the same category, where a category contains a group of associated words. This connection represents the strongest connection type of the four presented.

(2) **Category to cross-reference connection** occurs when a word has an associated cross-reference that points to the category number of another word. Figure 3 illustrates this connection type.

word [1]: *tide*

word [2]: *river*

Figure 3: Category to Cross-Reference Connection

Cross-references occur at the end of semi-colon groups and point to other categories that closely

relate to the current group of words.

(3) **Cross-reference to category connection** can be described as the inverse of the previous connection type. The cross-references associated with a word could be matched with the categories another word occurs under.

(4) **Same cross-reference connection** is defined as the cross-references of two words pointing to the same category number. Figure 4 gives an example of this connection type.

word [1]: *tide*
word [2]: *flood*

Figure 4: Same Cross-Reference Connection

The association of a cross-reference with a group of words indicates that the category the cross-reference is pointing to contains words that are related to the current group. Therefore, if two groups of words both have the same cross-references associated with them this implies that the words within these two groups could also be related.

3.2 Strength of Relations

A semantic relation between two words can be identified by the satisfaction of one or more of the four connection types identified in Roget's Thesaurus. The number of matches found between a pair of words for each of these connection types could be cumulated to provide an indication of the degree of connectivity or semantic relatedness between the two words. However, the number of matches found between a pair of words would be influenced by the number of times those words appear in the thesaurus. The probability of finding matches between words of a high occurrence rate would be greater than those of words of a low occurrence rate, due to the increased number of possible matches that could be made between these words. This could effect the accuracy of the assessment of the semantic relatedness between words. Consequently, the number of matches found for each connection type between a pair of words was normalised according to the frequency of occurrence of the words in the thesaurus. In this way, some indication of the strength of relations between words can also be used to select between candidate words.

3.3 Relations Algorithm

The following algorithm, hereafter referred to as the Relations Algorithm, locates semantic relations between words across a text and for each word in that text an associated score of its degree of relatedness to the rest of that text is calculated.

(1) Filter out the function words from the text¹;

1. For every document processed the function words are removed leaving the remaining content word set. The function word set includes words such as *the*, *and*, *there*, etc., these words would be limited for the identification of semantic relations between words because of their generality of usage.

(2) For each word in the text locate it in Roget's Thesaurus and extract the related information about categories and cross-references;

(3) Compare each word in the text to all the other words in the text and for each of these word pairs obtain the normalised number of matches found;

(4) For each word cumulate the total number of matches found and then calculate the average number of matches found for that word.

The average number of matches given for each word is used as an indication of the overall level of relatedness that word had with the rest of the words in the text.

4 Experiment

4.1 Method

Five files, each of at least 500 words in length, were selected at random from the Lancaster/Oslo/Bergen corpus [6]. These were scanned in using DeskScan software at 300 dots per inch and stored as tiff files. The words in these files were recognised using a word shape recogniser which uses multiple independent features and dictionary look-up [7]. For every sample word a number of alternative words from the dictionary were found based on features. These words were ranked in descending order according to the number of features matched. Those words with the highest number of features matched were biased for recognition. The output from this OCR system produced zero (i.e. word was not recognised), one or several alternative words at each word position. Table 1 gives the statistics for the types of OCR errors that occurred for the content words within each of the five test files. In some cases the original word was not recognised by the OCR system and in other cases the original word was recognised but there were other alternative words suggested at the same word position. For all other cases the original word was found and there were no alternative words suggested; that is, the OCR system successfully recognised the original word.

	Number of Words Not Recognised by OCR System	Number of Words with Alternatives	Total Number of OCR Errors
file #1	4	7	11
file #2	3	12	15
file #3	4	3	7
file #4	4	10	14
file #5	10	11	21

Table 1: OCR Errors for Content Words

The output from the OCR system for each of the five test files was input to the Relations Algorithm¹. This produced a score for each word indicating its measure of relatedness to the rest

1. Output was put in lower case which removed instances of alternative candidate words which arose due to case differences, for example *women* and *Women*

of the text. For those word positions where there were alternative words, the word with the highest score was biased for recognition. Figure 7 shows an example of the output from the application of this post-processing technique, where the words are given followed by their score attained (the presence of function words is indicated by an F).

```
women 0.6142
F
general 4.4714
F
```

Figure 7: OCR Output with Post-Processing

A text can be defined as being a piece of coherent language of any size and the comparison between word pairs could be done across an entire document or in smaller units within that document. The Relations Algorithm was applied at two levels of analysis, at the sentence level and the document level. For the sentence level semantic relations were considered between words within the same sentence and for the document level semantic relations were considered between words across an entire document.

4.2 Results

Table 2 gives the results of applying the Relations Algorithm to the content words of the OCR output for both the sentence level and the document level of analysis. These results are based on the number of words correctly selected for those words that had alternative candidates and the correct word was present. For example, at the document level of analysis for test file number 4 the correct words were selected for all the words that had alternative candidates (i.e. a result of 100%). The average recognition rate for the sentence level of analysis was 78.6%. The average recognition rate at the document level was 82.5%.

	Sentence Level	Document Level
file #1	71.4%	71.4%
file #2	83.3%	83.3%
file #3	66.7%	66.7%
file #4	90%	100%
file #5	81.8%	90.9%
Average	78.6%	82.5%

Table 2: Percentage of Correct Content Words Found

Table 3 gives the overall results for the OCR system and the results following the application of

the post-processing technique where the document level of analysis was employed. These results are given for all the content words in the test files, regardless of whether the original word was recognised.

	OCR Result	Result with Post-Processing	Maximum Result Possible
file #1	95.6%	97.6%	98.4%
file #2	94.9%	98.5%	98.9%
file #3	97.4%	98.1%	98.5%
file #4	94.9%	98.6%	98.6%
file #5	91.9%	95.8%	96.1%
Average	94.9%	97.7%	98.1%

Table 3: Recognition Rates for Content Words

4.3 Discussion

Overall the OCR system attained an average recognition rate of 94.9% for the five test files, for the content words. The application of the post-processing technique improved this recognition rate by a further 2.8% to 97.7%. For each of the five test files, the application of the post-processing technique improved upon the recognition rate of the OCR system alone.

For each of the five test files there were words that the OCR system failed to recognise. These words could not be recovered using the present method. This error rate prevents any post-processing of the OCR output producing a 100% recognition rate. For each of the five test files the maximum result possible was calculated (i.e. by considering the number of words that were not recognised by the OCR system). The results of the post-processing technique nearly attain the maximum possible result in every case and for test file number 4 the best recognition rate possible, of 98.6%, is achieved. Overall the post-processing technique was only 0.4% short of attaining the maximum possible result, whereas the output from the OCR system was 3.2% short of this target.

5 Conclusions

Those words which the post-processing technique failed to recognise may be able to be recognised by an alternative technique which employs higher-level information. Analysis of these instances revealed that when an incorrect word was selected it tended to have a low score compared to the next best score, whereas when a correct word was selected its score tended to be comparatively higher than the next best scoring word. This points to a possible method of error detection where the results of this post-processing technique could be deemed correct if a sufficiently reliable score (a threshold measure would have to be applied here) was attained. For those words where such a score was not attained another post-processing technique could then be applied (e.g., [8], [9], [10])

Taking into consideration the error rate of the OCR system any post-processing techniques have only a small potential for improvement. However, any reductions in the error rate are important for the usability of the technology. The present technique provides a computationally simple method

of removing a good proportion of errors. In cases where the correct word is not one of the candidates produced by the OCR system, the present technique offers no assistance. However, there may be possibilities for the development of a technique that could predict words, based on semantic, and other, information, and attempt to fill any 'gaps' in recognition.

References

- [1] M.S. Chodorow, R.J. Byrd & G.E. Heidorn (1985) 'Extracting semantic hierarchies from a large on-line dictionary', *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics*, Chicago, Illinois, pp. 299-304
- [2] T.G. Rose, L.J. Evett and A.C. Jobbins (1994) 'A context-based approach to text recognition', *Proceedings of the 3rd Annual Symposium on Document Analysis and Information Retrieval*, Las Vegas, Nevada, pp. 219-227
- [3] R.A. Amsler (1989) 'Research towards the development of a lexical knowledge base for natural language processing', *Proc. 1989 SIGIR Conf. Assoc. for Computing Machinery*, pp. 242-249
- [4] S.Y. Sedelow & A. Sedelow (1986) 'Thesaural knowledge representation', *Proceedings, 2nd Annual Conference of the University of Waterloo Centre for the New Oxford English Dictionary: Advances in Lexicography*, University of Waterloo
- [5] J. Morris & G. Hirst (1991) 'Lexical cohesion computed by thesaural relations as an indicator of the structure of text', *Computational Linguistics*, Vol. 17, No. 1, pp. 21-48
- [6] S. Johansson (1980) 'The LOB corpus of British-English texts: presentation and comments', *ALLC Journal*, 1
- [7] G. Raza (1995) 'Algorithms for the Recognition of Poor Quality Documents', *Unpublished Transfer Report*, Nottingham Trent University
- [8] T. G. Rose & L. J. Evett (1992) 'A large vocabulary semantic analyser for handwriting recognition', *AISB Quarterly*, 80, pp. 34-39
- [9] T. G. Rose & L. J. Evett (1995) 'The use of context in cursive script recognition', *Machine Vision and Applications*, 8, pp. 241-248
- [10] F. G. Keenan & L. J. Evett (1994) 'Applying syntactic information to text recognition', In L. J. Evett & T. G. Rose (Eds.), *Computational Linguistics for Speech and Handwriting Recognition*, AISB94 Workshop Proceedings, Leeds, England

Recognition of poor quality words without segmentation

G. Raza, N. Sherkat and R. J. Whitrow

Department of Computing, The Nottingham Trent University
Burton Street, Nottingham, NG1 4BU, England
Telephone: +44 (0)115 9418418
Fax: +44 (0)115 9486518
E-mail: ghr@doc.ntu.ac.uk

ABSTRACT

In this paper a method for recognition of poor quality documents is presented. The method is based on extraction of independent and robust features of 'objects' within a word without segmenting touching objects. A number of alternatives for each object are found. A lexical lookup method is used to verify the alternatives. The method has been applied to seven different poor quality facsimile messages in order to observe its effectiveness. The facsimile messages were first processed using a commercial OCR software and only the unrecognized words were selected and processed using the developed method. Different improvement rates were observed for the facsimile messages. The improvement ranged from 21.57% to 100%. The results obtained from these facsimile messages suggest that the developed method is capable of recognizing poor quality documents whilst avoiding segmentation of touching characters.

Keywords: Word recognition, Feature extraction, Segmentation, Lexical lookup

1. INTRODUCTION

During the past thirty years, substantial research effort has been devoted to Optical Character Recognition (OCR) [1]. As a result of this, many algorithms and systems have been developed. These algorithms and systems are shown to perform well for the recognition of clean text with characters well formed and separated from their neighbours [2]. The performance of such systems degrades as the quality of the input image degrades (poor quality documents). Poor quality documents contain a large number of touching, broken, or part-missing characters. Poor quality documents appear frequently in every day life. They come from different sources such as multiple generation photocopies, facsimile messages, low quality prints and newspapers, etc.

In most existing OCR systems, character segmentation is a key process. In this process, segmentation techniques are applied in order to segment word images into constituent characters. The segmented characters are then recognized. This segmentation and then recognition method is unsatisfactory for poor quality documents.

The segmentation of touching characters is one of the main problems in OCR and poses difficulties for recognition algorithms based on segmentation [3] [4]. As a result of segmentation, two or more characters may be grouped as one character, or one character, may be segmented into two characters.

In order to improve recognition rate, it is essential to have correct segmentation points, but this is difficult in the case of poor quality documents. It has been observed that half the errors in character recognition are due to incorrect segmentation [5]. It has been reported that the method of segmenting touching characters appears to be ad hoc in nature [6], and thus not particularly useful for poor quality documents (Figure 1).

Since segmentation of touching characters is not feasible for poor quality documents, we have attempted to avoid this step. In this paper, we present a new method for recognition of poor quality documents. This method attempts to recognize whole word using lexical lookup without segmenting touching characters into single letters. In this way it is intended to bypass errors caused by incorrect segmentation.

In this method, different independent features of each object of a word are extracted from the input sample word. *An object is a black pixel or group of black pixels completely surrounded by white pixels. In our case, an object is generally either a single character, two or more touching characters or a punctuation mark.* Different alternatives for each object are found by comparing its features with a database of ideal features for each object.

Thank you for your fax of 8 February 1995. I am pleased to learn that you and the BTEC moderator, Mr Hind will be coming to the college. The dates of your visit namely, 28 February and 1 March are suitable for us. I have got in touch with Tunku Abdul Rahman College Principal, Dr Lim Khaik Leang regarding your visit to the college for one day. He has kindly agreed to the visit on 1 March 1995. I informed the Principal, Dr Lim that both of you would like to get to know TAR College better and to familiarise with the Computer Science programmes at TAR College. In particular, you would like to learn more about the Certificate in Computer Studies and the Certificate in Computing and Accounting.

Thank you for your fax note of 20th February, 1995. The week starting 27th February, 1995, WIT is closed for vacation because of the Ramadhan festival on the 3rd and 4th of March, 1995 therefore all staff will be away to their respective home towns to celebrate the festival on the 3rd of March, 1995 and hence I regret that we may not be able to meet you on Thursday, 3rd of March, 1995 as requested. However, I look forward to another visit of yours where we may be able to take up the matter referred in your fax for discussion.

Figure 1: A sample poor quality document collected from facsimile 3 and facsimile 7

These alternatives are ranked according to the number of features matched. The alternative with the highest number of features matched is considered to be the most likely candidate.

These alternatives are combined to form different words using dictionary lookup. These words are ranked according to the total number of features matched. The word with the highest number of features matched is taken as the recognized sample word. Higher level linguistics information such as language syntax and semantics can be used to identify the correct word, if more than one word having the same number of matches is obtained as an output [7] [8] [9] [10].

2. FEATURE DEFINITION

Feature extraction is an important stage in pattern recognition and one of the crucial steps in character recognition. Correct feature extraction is essential for effective recognition of characters. In this approach, different independent features of every object of the sample word, which represent the ideal form of characters, are extracted. The features extracted are zones (upper (u), middle (m), lower (l) and full (f)), holes (upper, middle, lower), upper left side open, upper right side open, upper middle open, lower left side open, lower right side open, lower middle open, left side open, right side open and vertical bars. Our experiments have shown that these features are tolerant to noise such as broken,

touching and degraded characters. Additionally these features are expected to be consistent in a character of different fonts/sizes. Hence the proposed approach considers a database of single letter features (a-z, A-Z). This provides an efficient means of coping with poor quality documents. However, separate databases are required for finding different alternatives for touching objects as explained below.

3. DATABASE DEVELOPMENT

The developed approach aims to recognize each object within a word as a whole, without trying to segment touching characters, therefore we need databases of touching letters as well as single letters.

Development of a single letter database is trivial. In order to develop the two letter and three letter databases, all possible combinations of two letters and three letters in the words of a 4k dictionary are found. This dictionary consists of words frequently used in every day life. All these combinations are stored along with their features. Hence as a whole, we have used three different types of databases namely database 1 containing single letters (a, b, c, etc.), database 2 containing two letter combinations (ab, ac, al, ol, etc.) and database 3 containing three letter combinations (aba, ack, am, etc.) along with their features.

4. ALGORITHM DESCRIPTION

The basic algorithm for word recognition is illustrated in Figure 2.

First of all, the input image is separated into words. Then each word is passed to a feature extractor, where features of every object of the word (without segmenting touching characters) are extracted as explained in the previous section. We then find different alternatives for every object using the appropriate databases. Finally an algorithm tries to build dictionary words using the alternatives found. Finding object alternatives and the word building algorithm are explained below.

4.1 Finding object alternatives

Having found the features of an object, its vertical length (height) and horizontal length (width) are calculated. The approximate number of characters (N_{app}) in the object is calculated using EQ 1.

$$N_{app} = \text{width/height} + 1 \quad (\text{EQ } 1)$$

We must then compare the features of this object with the corresponding features of the database objects of length N_{app} .

EQ 1 has been found to be accurate for *most* objects but certain objects will give an incorrect result. For example, 'm' might give two characters, and 'fi' might give one character. Therefore we also compare the features of the object with database objects of lengths ($N_{app} - 1$) and ($N_{app} + 1$).

For every feature the matching number (M_n) is found using EQ 2.

$$M_n = 1 - \text{distance} \quad (\text{EQ } 2)$$

where

$\text{distance} = |\text{database object feature} - \text{sample object feature}|$

If $M_n = 1$, then we have a perfect match.

Finally, M_n of every feature are added to give Total matches (T_m) of the sample object with that of the database object. This database object is recorded in a table of alternatives with its T_m . After comparing features of the sample object with every database object, a maximum of fifty different alternatives based on best features matched are selected and noted. It has been observed that a maximum of fifty alternatives for every object are enough to contain the correct object. The table of alternatives is kept sorted in descending order according to T_m , so that the alternatives with highest T_m are on the top of the table.

The above procedure is repeated in order to find different alternatives for each object of the sample word. The

alternatives for every object are stored in a separate table of alternatives.

4.2 Building words

Each dictionary word is searched for in the table of alternatives using a recursive search. If a dictionary word can be found using tables of alternatives, then the corresponding features of each alternative, which were joined together to make a dictionary word, are added in order to find the total matching features of the word. The found dictionary word and its total matching features are stored in a `word_table`. The `word_table` is kept sorted in descending order according to the total matching features. This is done so that words with maximum number of total matching features are on top of the `word_table`.

The word with the highest total number of matching features in the `match_table` is considered to be the recognized word. If there are more than one word having the same number of features or if the sample word is in the `match_table`, but at a lower rank, then the higher level linguistic information such as language syntax and semantics may be used to identify the correct word [9]. In this way we can further improve the performance of the recognition system.

5. EXPERIMENTAL RESULTS

Seven different actual facsimile messages were collected to observe the performance of the developed recognizer using a commercial recognition software. These facsimile messages contained text of two different fonts namely Helvetica and Times Roman with a point size of 12 containing Plain, Bold and Underlined words.

The facsimile messages were then scanned using a Hewlett Packard Scanjet Plus scanner. The software used to scan these documents was Deskscan. The documents were scanned at 300 dots per inch.

The facsimile messages were initially processed using the commercial software. All the words which the software failed to recognize correctly were marked. The words which were present in the 4000 word dictionary, used for the present research, were separated from the unrecognized words. This separation was done so that we attempt to recognize only those unrecognized words which were present in the dictionary used. These words were then tested using the developed recognizer in order to assess any improvements. The results obtained using the commercial OCR software and the developed recognizer together with the percentage improvements are presented in Table 1.

It is clear from the table that the recognizer gave varied improvements for all facsimile messages ranging from

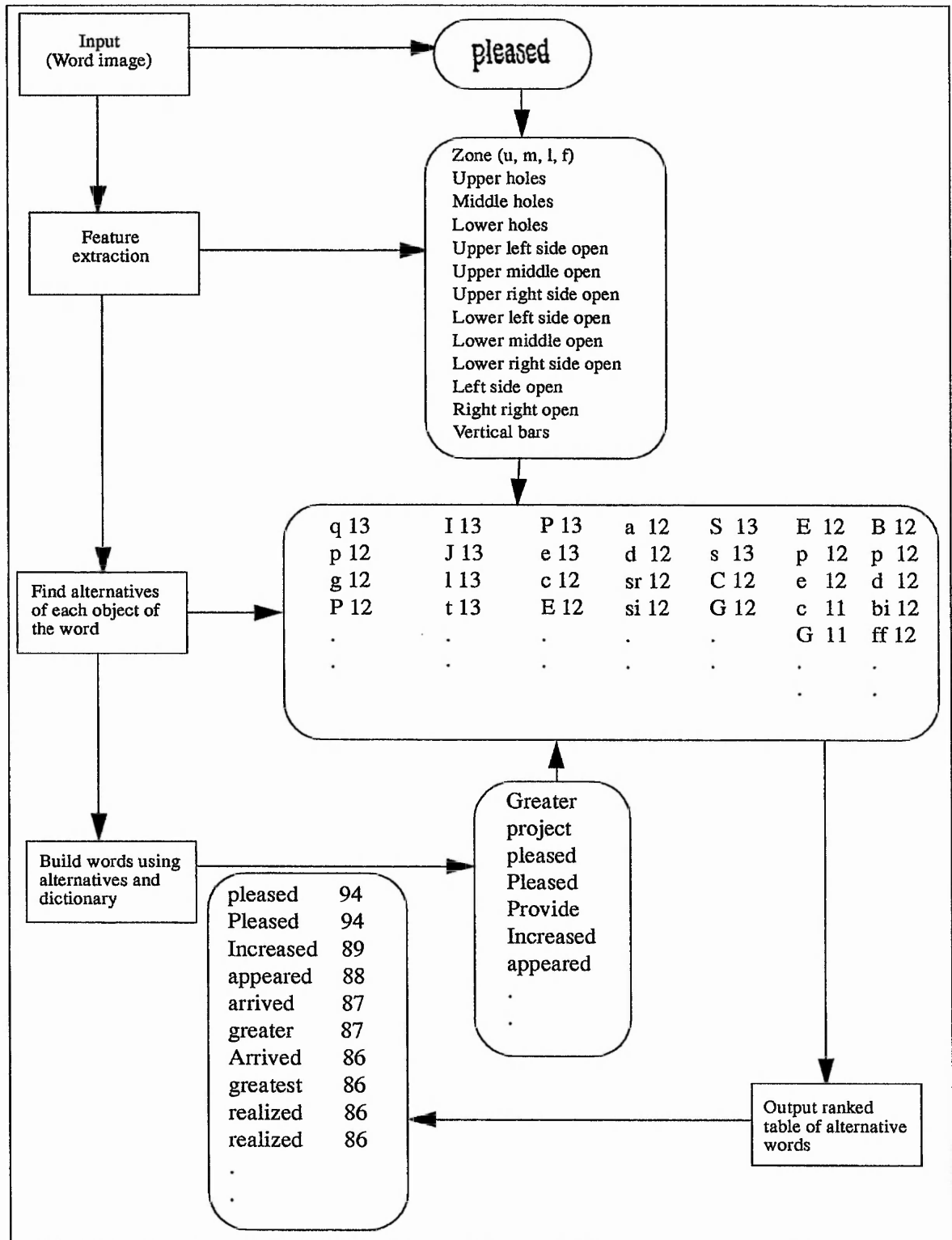


Figure 2: An outline of the word recognition algorithm

21.57% to 100%. The main reasons for these variations in different facsimile messages are the number of touching characters and underlined words. The developed recognizer can successfully recognize words containing maximum of three touching characters. It has been observed that the underlined words have nearly all of their characters touching. Therefore a slight improvement is observed for facsimile messages having more words containing greater than three touching characters and underlined words.

Although we acknowledge the fact that the developed recognizer uses a dictionary, it has been observed that the commercial software does not provide sufficient output for most of the unrecognized words (see Figure 3). Therefore a dictionary search (spell check) is highly unlikely to improve the results.

6. CONCLUSION AND FUTURE WORK

A new method for the recognition of printed text containing poor quality word images is presented. The method is based on extraction of multiple independent features. This method avoids segmentation of touching characters and hence bypasses errors incurred during the segmentation stage of a typical OCR system. It tries to identify each object of the word based on features. The method has been proved useful for the recognition of poor quality words of different fonts and sizes. This method has been observed to be particularly good and robust for the recognition of poor quality documents needing a limited lexicon. Work is in progress to extend the method so that it can cope with word images containing any number of touching characters.

7. ACKNOWLEDGEMENTS

HP scanjet and Deskscan are products of Hewlett Packard.

8. REFERENCES

- [1] S. Impedovo, L. Ottaviano and S. Occinegro, Optical character recognition - a survey, *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 5, NO. 1 & 2, 1991, pp. 1-24.
- [2] H. S. Baird, Feature identification for hybrid structural/statistical pattern classification, *Computer Vision, Graphics, and Image Processing*, Vol. 42, No. 3, pp. 318-333.
- [3] R. Casey and K. Wong, Document-analysis system and techniques, In R. Kasturi and M. Trivedi, editors, *Image Analysis and Applications*, New York, 1990, pp. 1-35.
- [4] G. Nagy, At the frontiers of OCR, *Proceedings of the IEEE*, Vol. 80, No. 7, 1992.
- [5] C. H. Chen, J. L. DeCurtins, Word recognition in a segmentation-free approach to OCR. *IEEE*, 1993, pp. 573-576.
- [6] D. G. Elliman, I. T. Lancaster, A review on segmentation and contextual analysis techniques for text recognition. *Pattern Recognition*, 23, No. 3/4, 1990, pp. 337-346.
- [7] F. G. Keenan and L. J. Evett, Applying syntactic information to text recognition, in L. J. Evett and T. G. Rose (Eds.) *Computational Linguistics for speech and Handwriting Recognition*, AISB Workshop, 1994.
- [8] J. J. Hull, Feature selection and language syntax in text recognition. in *From Pixels to Features*, J. C. Simon (editor), North Holland, 1989, pp. 249-260.
- [9] A. C. Jobbins, G. Raza, L. J. Evett and N. Sherkat, Postprocessing for OCR: Correcting errors using semantic relations. in L. J. Evett and T. G. Rose (Eds.) *Language Engineering for Document Analysis and Recognition (LEDAR)*, AISB96 Workshop, Sussex, England, 1996.
- [10] T. G. Rose and L. J. Evett, The use of context in cursive script recognition, *Machine Vision and Applications*, Vol. 8, 1995, pp. 241-248.

Sample words	Commercial software output	Recognizer output
ATTENTION	~IINIION	ATTENTION
NOTTINGHAM	NOJJINGH~M	NOTTINGHAM
UNIVERSITY	UNIUIR~IY	UNIVERSITY
approximately	~cproxirna'ely	approximately
every	e.~ry	every
February	Feb.l.l.l.aty	February
you	yo~i	YOU
yours	yolil.S	virtual
matter	n~attef	matter
requested.	l-aqu~xtd	requested
Thursday	I~ursday	initially

Figure 3: A commercial software and recognizer output for some sample words

Sample no	No. of words not recognized using software	No. of words not in Dict. among unrecognized words	No. of words in Dict. among unrecognized words	No. of words not recognized using recognizer	No. of words recognized using recognizer	Percent improvement (%)
Fax1	33	6	27	3	24	88.89
Fax2	19	10	9	1	8	88.89
Fax3	64	13	51	40	11	21.57
Fax4	12	2	10	0	10	100
Fax5	44	9	35	23	12	34.28
Fax6	20	6	14	6	8	57.14
Fax7	27	5	22	9	13	59.09

Table 1: Improvements using the developed recognizer

WORD RECOGNITION USING MULTIPLE INDEPENDENT FEATURES

G. Raza, N. Sherkat and R. J. Whitrow

Department of Computing, The Nottingham Trent University
Burton Street, Nottingham, NG1 4BU, England
Telephone: +44 (0)115 9418418
Fax: +44 (0)115 9486518
E-mail: ghr@doc.ntu.ac.uk

Abstract

A method for the recognition of poor quality documents containing many touching characters is presented. The method is based on extraction of independent and robust features without segmentation. A lexical lookup method is used to verify the alternatives. It has been demonstrated that the developed method is effective for the recognition of poor quality documents whilst avoiding segmentation of touching characters.

Keywords: Word recognition, Segmentation, Feature extraction, Database development, Lexical lookup

1 Introduction

Much work has been done in the area of Optical Character Recognition (OCR) (Govindan 90). The work has led to the development of many algorithms and systems. The performance of these algorithms and systems for good quality documents is acceptable. However, these systems do not perform particularly well on poor quality documents such as facsimile messages, low quality prints, photocopies etc. Humans can still read very poor quality documents.

Poor quality documents have a large number of touching, broken and part-missing or unwanted objects. Most conventional OCR systems rely on segmentation of touching characters. This is one of the main problems in recognition of poor quality documents. In order to obtain a greater recognition rate, it is very important to have correct segmentation points. It has been observed that half of the errors in character recognition are due to incorrect segmentation (Chen & DeCurtins 93). The method of segmenting touching characters appears to be ad hoc in nature (Elliman & Lancaster 90), and thus not particularly useful for poor quality documents (Figure 1).

Looking at the difficulties faced by the segmentation problem in character recognition, we have attempted to overcome this problem by avoiding it as much as possible. In this paper, we present a new approach for recognition of poor quality documents. This approach attempts to recognize whole words using lexicon lookup without segmenting touching characters into single letters. In this way it is intended to bypass errors caused by incorrect segmentation.

In this approach, different independent features of each object of a word are extracted from the input sample word. An object is a black pixel or group of black pixels completely surrounded by white pixels. In our case, an object is generally either a single character, two or more touching characters or a punctuation mark. Different alternatives for each object are found by comparing its features with a database of ideal features for each object. These alternatives are ranked according to the number of features matched. The alternative with the highest number of features matched is considered to be the most likely candidate.

These alternatives are combined to form different words using dictionary lookup. These words are also ranked according to the total number of features matched. The word with most features matched is taken as the recognized sample word. If there is more than one word, higher level linguistics information such as language syntax and semantics can be used to identify the correct word. For example, the semantics of a constrained domain (chess games) has been used to correct character recognition errors (Baird & Thompson 90). Language level syntax has been used to improve word recognition by reducing the number of alternatives for a word's identity based on the hypothesized syntactic categories for two adjacent words (Hull 89).

I am pleased to learn from you that your department is able to admit more than 20 students into the B Eng (Hons) in Electronics and Computing. That being the case and if the initial figure can be increased to 40 with possibility of further increase in the future, the college is interested to begin planning for the commencement of this course in April 1995. I understand that this course has been accredited by IEE, UK but not by BCS.

Figure 1: A sample poor quality document

2 Feature definition

Feature extraction is an important stage in pattern recognition and one of the crucial steps in character recognition. Correct feature extraction is essential for effective recognition of characters. In this approach, different independent features of every object of the sample word, which represent the ideal form of the characters, are extracted. The features extracted are zones (upper, middle, lower and full), holes (upper, middle, lower), upper left side open, upper right side open, upper middle open, lower left side open, lower right side open, lower middle open, left side open, right side open and vertical bars. Our experiments have shown that these features are tolerant to noise such as broken, touching and degraded characters. Additionally these features are expected to be consistent in a character of different fonts/sizes. Hence the proposed approach considers a database of single letter features (a-z, A-Z). This provides an efficient means of coping with poor quality documents. However, separate databases are required for finding different alternatives for touching objects as explained below.

3 Database development

The developed approach aims to recognize each object within a word as a whole, without trying to segment touching characters, therefore we need databases of touching letters as well as single letters.

Development of a single letter database is trivial. In order to develop two letter and three letter databases, all possible combinations of two letters and three letters in the words of a 4k dictionary are found. This dictionary contains words used frequently in every day life. All these combinations are stored along with their features. Hence as a whole, we have used three different kinds of databases namely database 1 containing single letters (a, b, c, etc.), database 2 containing two letter combinations (ab, ac, al, ol, etc.) and database 3 containing three letter combinations (aba, ack, arn, etc.) along with their features.

4 Algorithm description

The basic algorithm for word recognition is illustrated in Figure 2.

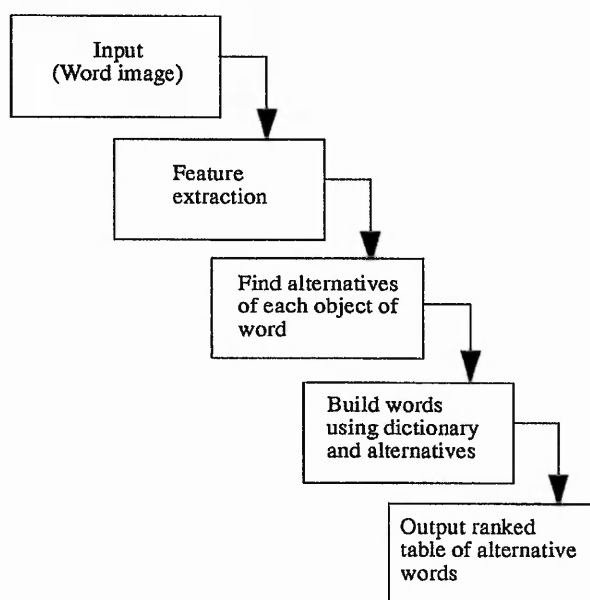


Figure 2: An outline of the word recognition algorithm

In this approach, the first step is to separate the input image into words. Then each word is passed to a feature extractor, where features of every object of the word (without segmenting touching characters) are extracted as explained in the previous section. We then find different alternatives for every object using the appropriate databases. Finally an algorithm tries to build dictionary words using alternatives found. Finding object alternatives and the word building algorithm are explained below.

4.1 Finding object alternatives

Having found the features of an object, its vertical length (height) and horizontal length (width) are calculated. The approximate number of characters (N_{app}) in the object is calculated using EQ 1.

$$N_{app} = \text{width/height} + 1 \quad (\text{EQ 1})$$

We must then compare the features of this object with the corresponding features of the database objects of length N_{app} .

EQ 1 has been found to be accurate for *most* objects but certain objects will give an incorrect result. For example, 'm' might give two characters, and 'fi' might give one character. Therefore we also compare the features of the object with database objects of lengths ($N_{app} - 1$) and ($N_{app} + 1$).

For every feature the matching number (M_n) is found using EQ 2.

$$M_n = 1 - \text{distance} \quad (\text{EQ 2})$$

where

distance = |database object feature - sample object feature|

If $M_n = 1$, then we have a perfect match.

At the end, M_n of every feature is added to give Total matches (T_m) of the sample object with database object. This database object is recorded in a table of alternatives with its T_m . After comparing features of the sample object with every database object, fifty different alternatives based on best features matched are selected and noted. The table of alternatives is kept sorted in descending order according to T_m , so that the alternatives with highest T_m are on the top of the table.

The above procedure is repeated in order to find different alternatives for each object of the sample word. The alternatives of every object are stored in a separate table of alternatives.

4.2 Building words

Words are selected from the dictionary used and are searched in the table of alternatives using a recursive search. If a dictionary word can be found using tables of alternatives, then the corresponding features of each alternative which joined together to make a dictionary word are added in order to find the total matching features of the word. The found dictionary word and its total matching features are stored in a word_table. The word_table is kept sorted in descending order according to the total matching features. This is done so that words with maximum number of total matching features are on top of the word_table.

The word with the highest total number of matching features in the match_table is considered as the recognized word. If there are more than one word having the same number of features or if the sample word is in the match_table, but at a lower rank, then the higher level linguistic information such as language syntax and semantics may be used to identify the correct word (Jobbins et al. 96). In this way we can further improve the performance of the recognition system.

5 Experimental results

In order to evaluate the recognizer's performance with fax messages, five different Fax messages were used. These Fax messages were written in Times Roman font, 12 point size. A large number of touching and broken characters may be found in these documents. Some words had part missing letters and some contained unwanted extra parts. Different parts of these Fax messages are shown in Figure 1 and Figure 3.

Department I am pleased to learn from
lecturers January Looking forward
wishes February Grand
Burton Street more than 20 students
will sincerely department able
further With best wishes

Figure 3: Sample words selected from different fax messages