G60/97

Table of Errata

Reference	Erratum	Correction
Page 3-17, paragraph 2, line 1	"...much of the errors..."	"...many of the errors..."
Page 3-17, paragraph 2, line 7	"...to some extend..."	"...to some extent..."
Page 4-24, paragraph 2, line 1	"The is a..."	"There is a..."
Page 4-25, paragraph 9, line 3	"...(refereed to..."	"...(referred to..."
Page 6-4, paragraph 4, line 2	"...remain parameters..."	"...remaining parameters..."
Page 6-4, paragraph 5, line 2	"...remain parameters..."	"...remaining parameters..."
Page 6-6, paragraph 2, line 1	"...the multiple of..."	"...the product of..."
Page 7-3, paragraph 3, line 3	"...discreet document processing..."	"...discrete document processing..."
Page 7-5, Figure 7.1	"Real World Constriant..."	"Real World Constraint..."
Page 7-16, paragraph 2, line 8	"...have only direction..."	"...have only one direction..."

DIGITAL IMAGE PROCESSING IN A HIGH VOLUME DOCUMENT ENVIRONMENT

A thesis
presented as a requirement for admission to the degree of

DOCTOR OF PHILOSOPHY

by

BRIAN MAXWELL GRIFFIN
B.Eng. [Monash University.]
M. I.E.Aust., M. A.P.E.S.M.A.

at the

Digital Imaging Applications Centre
Gippsland School of Engineering,
Monash University, Victoria, Australia.

MARCH 1997

ABSTRACT

This thesis is concerned with the processing of digital images in the high volume document environment. Such environments are typified by business requirements to digitise printed documents such as company papers and archive them in an on-line referable format. The Australian Securities Commission's National Information Processing Centre was used as the principal case study for this work.

The first specific objective of the research was to analyse and model the effects of variables which govern the processing performance of digital images in the high volume document environment. The characteristic variables and performance classes which pertain to the high volume document environment are defined by the research work. The effects of the characteristic variables on the performances classes are established through experimental work and models are developed relating the characteristic variables to the performance classes.

The second objective of the research work was to model the performance of digital image processing in the high volume environment thus establishing a methodology for optimising that performance. The generic model was evolved through a series of increasingly sophisticated interim models to a specific model for the high volume environment. This final model, which incorporates the first objective's modelling work, is then used to establish the methodology for optimising digital image processing in the high volume document environment.

The research makes a number of original contributions to the body of knowledge. The research reports a comprehensive set of results which define the effects of certain characteristic variables on the performance classes of OCR systems. An original model is developed for predicting the performance of OCR systems in terms of their characteristic variables. Another original model is developed to describe and optimise the performance of digital image processing systems in the high volume document environment. The research work presented in this thesis also describes a series of innovative tools for analysing OCR systems and digital image document processing systems.

STATEMENT OF AUTHORSHIP

I hereby certify that this thesis does not incorporate without acknowledgment any material previously submitted for a degree or diploma in any university; and that to the best of my knowledge and belief it does not contain any material previously published or written by another person where due reference is not made in the text.

[redacted]

Author: B. M. Griffin

ACKNOWLEDGMENTS

The author would like to acknowledge the assistance and support of the following people and organisations in the preparation of this thesis:

- Associate Professor Ken Spriggs, principal supervisor.
- Dr. Yousef Ibrahim, associate supervisor
- Mr. Geoff Vains
- Mr. Waran Nageswaran
- Mrs. Jolanta Parker
- Ms. Maria Creighton
- Mr. Mark Anderson
- My Family
- Digital Imaging Applications Centre
- School of Engineering, Monash University Gippsland Campus
- National Information Processing Centre, Australian Securities Commission

SUPPORTING PUBLICATIONS

Published To Date:

- B. Griffin, K. Spriggs, G. Vains, W. Nageswaran, "OCR performance in a high volume commercial environment," *Proceedings of the Digital Image Computing: Techniques and Applications 1995 Conference*, Macquarie University, vol. 2, pp. 525-532, 1993.
- B. Griffin, K. Spriggs, Y. Ibrahim, G. Vains, "OCR performance optimisation in a high volume commercial environment," *Proceedings of the Digital Image Computing: Techniques and Applications 1995 Conference*, University of Queensland, pp. 485-490, 1995.
- B. Griffin, K. Spriggs, G. Vains, W. Nageswaran, "Optical character recognition document processing system design using a high level visual language," *presented at the Seventh Australian Software Engineering Conference*, Sydney, 1993.
- B. Griffin, "Keyfield data extraction for document imaging systems," *presented at the Fourth Digitial Imaging Applications Centre Industrial Seminar*, Monash University, May 1994.

Publications in Preparation:

- B. Griffin, K. Spriggs, "Modeling OCR systems in the high volume commercial environment," to be submitted to the Digital Image Computing: Techniques and Applications 1997 Conference.
- B. Griffin, K. Spriggs, Y. Ibrahim, "Application of system models to high volume commercial OCR systems," to be submitted to the Digital Image Computing: Techniques and Applications 1997 Conference.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1. INTRODUCTION

This chapter introduces the research area of the thesis and covers some of the background to the research. The evolution of the research area is described before examining the contribution of the research work reported to the wider body of knowledge. The collaboration between the Digital Imaging Applications Centre (DIAC) research center and the companies supporting the research work is discussed. The organisation and presentation of the rest of the thesis is then described.

1.1.BACKGROUND

This section describes some of the background to the research work. The objectives and scope of the research work are defined and the motivation for the research work is discussed.

With recent reports that the computer driven office has resulted in a 400 per cent increase in paper use [1] in Australia alone over the last 10 years, the fallacy of the paperless office and the need for efficient text recognition systems has become increasingly apparent. From the viewpoint of the rapidly increasing volume of paper based information, the research of efficient means of processing that information is well justified.

Objectives of the research reported in the thesis include analysis of the performance of OCR and digital document processing systems and development of a series of tools for analysing and optimising those systems. These objectives form original contributions of the research to the knowledge in digital image document processing and are discussed in further detail in Section 1.3.

As the title of the thesis implies, the research work examines digital image processing in the high volume document environment. The research work falls within the broad domain of character recognition, but encompasses areas such as optical character recognition, high volume document processing and image processing performance analysis. In particular, the research work deals with the development of apparatus for analysing digital image document processing systems and the development of models for optimising those systems. The research also reports a comprehensive set of results which determine the effects of certain characteristic variables on the performance classes of OCR systems.

To further define the scope of the research work, Figure 1.1 depicts the breakdown of the character recognition domain into smaller and more specific sub-domains. Figure 1.1 is an adaptation of similar figures presented by Doermann [2] and Impedovo et al. [3]. The area of interest diagram is not intended as an exhaustive description of all the subdomains of character recognition, but rather shows the relationship of the various domains and subdomains which are relevant to the research.

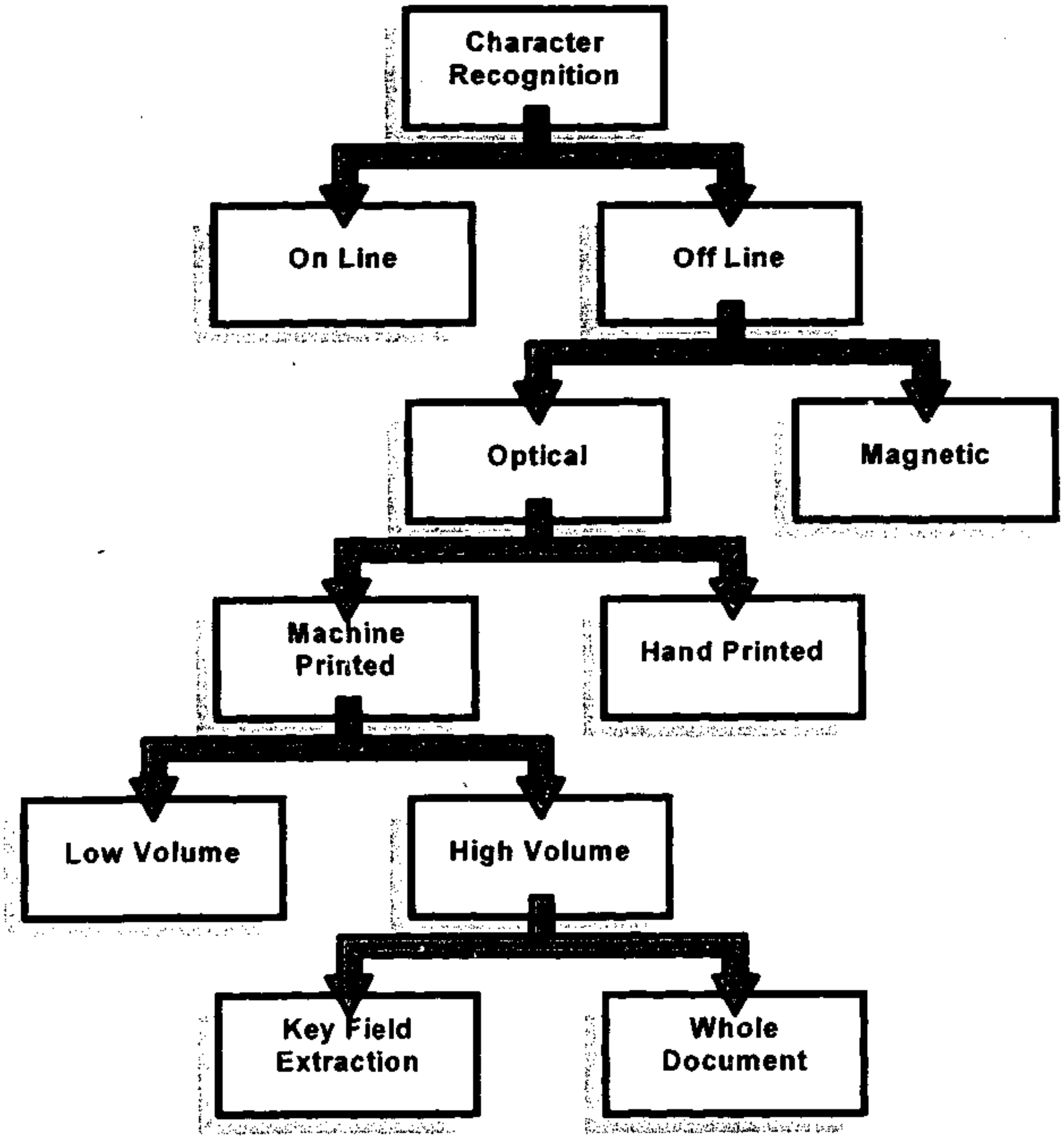
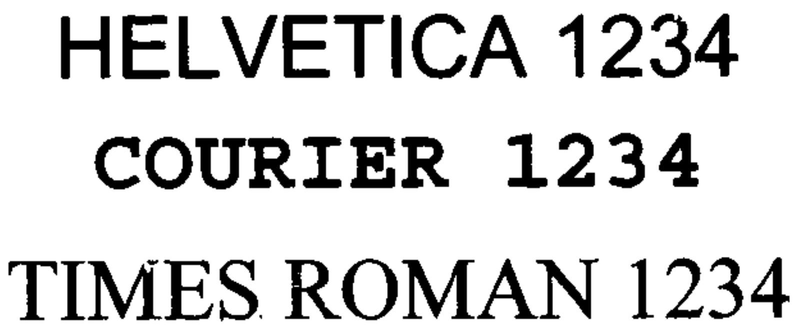


Figure 1.1 Areas of interest in the character recognition domain
This figure shows the division of areas of interest in the character recognition domain into successively smaller areas of interest.

The scope of the research reported in this thesis encompasses optical character recognition of machine printed documents in the high volume environment. Figure 1.1 shows the high volume machine printed domain including two relevant sub-domains, those of key field extraction and whole document recognition. By focusing the scope of the research on these areas it does not exclude those peripheral areas which may impact upon the research, but allows the research results to be more detailed and specific.

Even by focusing the scope of the research on machine printed documents in the high volume environment, there is still a diverse range of characters to be recognised. An example of the range of characters to be recognised is given in Figure 1.2. The example shows three contemporary machine printed typefaces; Helvetica, Courier and Times Roman. Figure 1.2 also shows two characteristics of typefaces which can affect recognition; viz. serifs and spacing.



HELVETICA 1234

COURIER 1234

TIMES ROMAN 1234

Figure 1.2 Examples of contemporary machine printed typefaces
Three common examples of contemporary typefaces are shown. The top typeface is a proportional, sans-serif typeface named Helvetica. The middle typeface is a non-proportional, sans-serif typeface named Courier. The bottom typeface is a proportional, serif typeface called Times Roman.

Where reference is made to digital image document processing within the thesis, it is implied that it is machine printed and optically scanned documents which are being considered unless stated otherwise.

There are two main objectives for researching digital image processing in the high volume document environment:

- The first objective is to be able to accurately measure and predict the performance of digital image processing systems in the high volume document environment for a given set of characteristic variables. By then controlling the characteristic variables it is possible to manipulate the performance of the digital image processing system.

- The second objective is to be able to accurately optimise the performance of digital image processing systems in the high volume document environment. Optimisation of document processing system performance is particularly important in the high volume environment where the greater the volume, the greater the potential savings in resources and time that can be achieved from a performance gain.

1.2. EVOLUTION OF THE RESEARCH

The research reported in this thesis has evolved from the works of several notable international researchers who have pioneered the digital image document processing field.

The literature survey in Chapter 2 draws upon the work reported in the survey papers by Nagy [4], Mori et al. [5], and Impedovo et al. [3]. Much of the theory which is overviewed in Section 2.5 is based on these survey papers and the work of Bokser [6] on omnidocument technologies.

The work reported by Baird on the skew angle of printed documents [7], document image defect models [8], and their uses [9], forms the basis which the preliminary experimental work reported in Chapter 4 is built upon. Later work by Griffin et al. [10] extends Baird's previous work and further supports the results reported in Chapter 4.

Work by Ho et al. [11] on the evaluation of OCR accuracy using synthetic data, and Doermann and Yao [12] on generating synthetic data for text analysis systems confirms many of the findings reported in Chapter 5. Further work by Griffin et al. [13] complements the findings reported in Chapter 5.

The modelling work reported in Chapter 7 and Chapter 8 evolves from the works of Casey et al. [14] on intelligent forms processing systems and Srihari [15] on high performance reading machines. The OCR system models presented by Doermann [2] and Impedovo et al. [3] were also used as a basis for the more sophisticated models reported in Chapter 7. Further research work in the area of digital image processing in the high volume document environment is reported in Section 2.6.2.

1.3. CONTRIBUTION OF THE RESEARCH

The research work carried out and reported in this thesis is shown to make several significant contributions to the knowledge of digital image document processing. The work is significant because it contributes several solutions to problems which exist in the knowledge area. The work is original in that it addresses a void in the area of knowledge which has only been partly contributed to by the current literature. While all of the research work reported is original in some respect, the original contribution of the research work reported is focused on four major areas:

- A comprehensive set of experimental results which define the effects of certain characteristic variables on the performance classes of OCR systems.
- The development of an original model for predicting the performance of OCR systems in terms of the system's characteristic variables.
- The development of another original model for describing and optimising the performance of digital image processing systems in the high volume document environment which incorporates the previous OCR model.
- Also, through the work of this thesis, a series of innovative tools were developed. These new tools are essential not only in analysing an OCR system in terms of its performance classes and characteristic variables, but also in its optimisation.

These areas of contribution of the research represent a significant advancement in the published body of knowledge of digital image document processing.

1.4. INDUSTRIAL COLLABORATION

Major sections of the research work presented in this thesis were made possible only by collaboration between DIAC and the relevant industrial organisations. The collaboration of the relevant industries is important because it illustrates the practical nature of the research and the applicability of the research results to industrial systems. The industrial organisations which assisted the research work at DIAC included the Australian Securities Commission (ASC), KODAK Australia and the Collaborative Information Technology Research Institute (CITRI).

DIAC provided the laboratory space, equipment and other facilities for most of the experimental work reported in the thesis. The expertise of the DIAC researchers in the digital imaging field assisted the direction of the research and development work. DIAC also served as an administrative body through which collaboration with the other external industries was organised.

The ASC provided access to the high volume document processing environment which was necessary for collecting the data upon which much of the research work is based. The ASC's National Information Processing Centre (NIPC) provided experience with digital image processing systems in an commercial environment. The ASC assisted the development of the DIAC laboratories which were used for the preliminary experimental work described in Chapter 4.

KODAK Australia provided information and expertise regarding the high volume document processing systems which were used by the ASC. KODAK supplied software modules for image viewing and manipulation which were incorporated into the prototypes developed as part of the research.

The Collaborative Information Technology Research Institute conducted research work for the ASC in the area of document image transmission. The areas of research for CITRI and DIAC overlapped to some extent and regular seminars facilitated the exchange of research data. CITRI provided information on the ASC's national document transmission network which was relevant to the practical application of the research.

1.5. ORGANISATION OF THE THESIS

This section describes the general organisation of the thesis. It outlines each of the chapters and provides an overview or the research work.

A survey of the current research literature is reported in Chapter 2. It introduces the basic principles of digital image processing which lead into a historical perspective of digital image processing from the high volume document environment viewpoint. State of the art OCR systems, current research areas and key enabling technologies are examined. An overview is presented of the theories applicable to digital image document processing and the high volume document environment in particular.

The research work progresses from the introduction in two distinct phases. The first phase concentrates on OCR performance at a elemental process level, while the second focuses on digital image document processing at an overall system level.

An analysis of the literature is conducted in Chapter 3 using LitBase which is a literature database tool developed specifically for analysing the research literature. The development of LitBase is examined both as a literature analysis tool and as a test bed for OCR experimentation.

The preliminary experimental work with LitBase led to the results reported in Chapter 4. The experimental work investigates the relationship between the OCR performance classes and the characteristic variables of an OCR system. An analysis of the these preliminary experimental results is conducted in Chapter 5. The analysis defines and quantifies the relationship between the OCR performance classes and characteristic variables. Chapter 5 also develops the accuracy-resolution-text size (ART) and resolution-text size-speed (RTS) curves. The ART and RTS curves visualise the relationships between the OCR performance classes and characteristic variables. The elementary models derived from the ART and RTS curves are shown to predict OCR system performance. The elementary models are developed into an OCR system performance optimiser (OSPO) tool which is shown to be able to tune the characteristic variables of an OCR system to optimise the systems performance.

Chapter 6 reports the development and implementation of two prototype OCR systems developed using the previously established performance class, characteristic variable relationship. The prototypes substantiate the analysis of the experimental results which were reported in Chapter 4 and show the reliability of the tools and models based on the reported analysis in Chapter 5.

The work reported in Chapters 2 through 6 represents the first phase of the research work. It describes the research and development of an original technique and model for optimising OCR performance in the high volume environment. The second phase of the research work is reported in Chapters 7 and Chapter 8. It extends the optimisation and modelling of the first phase to the whole digital image document processing system.

The modelling of the digital image document processing system is developed in Chapter 7. A series of five increasingly sophisticated models are developed which incorporate the OCR optimisation model developed in Chapter 4 and Chapter 5. Experimentation and analysis of the models is reported in Chapter 8. It shows the ability of the models to optimise digital image document processing system performance under conditions typical of the high volume environment. Chapter 9 presents the conclusion to the research work in terms of its general achievement and original contributions to the area of knowledge. It also lists several avenues of future research work and apparatus development.

Following the conclusion are appendices, a bibliography, a glossary, and an index. The appendices cover detailed apparatus specifications, software source code listing for the models and tools developed for the research work and tables of summarised experimental data.

CHAPTER 2

LITERATURE SURVEY

CHAPTER CONTENTS **PAGE**

2. LITERATURE SURVEY

This chapter introduces digital imaging and its application to document processing. A historical perspective of OCR systems is presented, highlighting significant milestones and developments. Systems using the current state of the art OCR techniques are described in further detail. Key enabling technologies and their impact on digital imaging are discussed. An overview of digital imaging theories is presented to cover the various aspects applicable to document processing. Particular focus is then placed upon the application of digital image processing to the high volume document environment and is the main focus of this thesis.

2.1. INTRODUCTION

This section introduces digital image processing by looking at document processing systems prior to the digital era. Some of the early applications of digital imaging to document processing are examined to show how the technology began to develop. An elementary OCR model is used to describe the basic principles of digital image document processing. Once these basic principles are established, the criteria for evaluating OCR system performance are defined.

2.1.1. Pre-Digital Document Processing

Digital image document processing did not emerge until computational technology had advanced sufficiently to allow machines to process document images. Even before that time, however, research work in Germany and the U.S. was being conducted in optical character recognition [5]. Patents were obtained on OCR as early as 1929 [5].

This research work into OCR progressed to the point where several early applications were developed using digital imaging techniques to process documents.

2.1.2.Early Applications of Digital Imaging to Document Processing

There are a number of early applications of digital imaging to document processing e.g. [3], [5], [16], [17]. Mark sensing and magnetic ink character recognition are two such examples which characterise these early applications.

2.1.2.1.Mark Sensing

Mark sensing [17] was a relatively simple form of digital image document processing. It involved the detection of an object on a document but did not include the recognition of that object. Because recognition of the object was not required, the process was simple and required low computational power. Mark sensing was therefore implemented before the other more complex forms of document image processing such as magnetic ink character recognition.

The mark which was sensed could be a hole punched in the document and was read by mechanical or optical means. The mark could also be made by pen or pencil and read optically. Weaving looms used a mechanical system [18] similar to mark sensing to achieve patterns in the weaving thus representing an earlier example of the technology.

Punch cards [17] were used for mark sensing by programmers to input program code into computers. These punch cards could be created manually by the programmer and fed into a punch card reader for input into the computer. Paper tape [17] was another form of mark sensing which allowed information to be stored on a continuous reel of paper for later retrieval. Mark sensing was also used for processing forms which required limited responses.

Mark sensing required information to exist at pre-determined positions on the document. Information at positions other than pre-determined positions was ignored. This requirement for prior knowledge of a document's mark sense positions is another limitation of mark sensing.

Because of these limitations imposed by mark sensing, the range of document processing tasks to which it could be applied was restricted. More refined forms of digital image processing had to be created to distinguish not only whether an object existed on the document, but also to determine what the object represented. One such form of digital image processing is magnetic ink character recognition.

2.1.2.2. Magnetic Ink Character Recognition

One of the ways in which Magnetic Ink Character Recognition (MICR) [5] differs from mark sensing is that MICR can distinguish one mark or character from another. MICR can therefore overcome some of the limitations imposed by mark sensing on digital image document processing.

MICR involved passing a vertical slit over the characters and summing the magnetic information at each slit position. Each character was reduced into a one dimensional array. Comparing the array with a list of values allowed the character to be recognised. This process is depicted in Figure 2.1 for the number '5'.

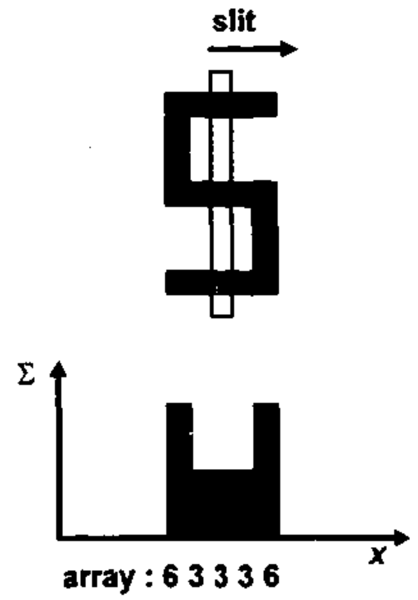


Figure 2.1 Magnetic ink character recognition process
A vertical slit passes over the character (top image) and sums the magnetic information at each *x* position. The sums (bottom image) are shown as a function of *x*. The corresponding array is used to classify the character.

MICR required a special typeface which when reduced to one dimension gave unique arrays for each character. MICR also required a magnetic ink. These were restrictions which limited the applications to which MICR could be applied. An example of the MICR typeface is shown in Figure 2.2.

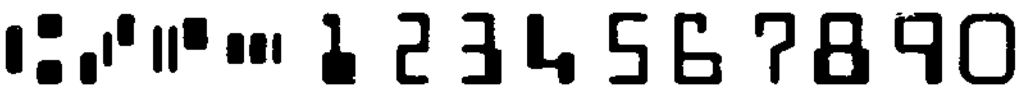


Figure 2.2 MICR typeface example
The typeface shown is used for Magnetic Ink Character Recognition. Apart from being printed in magnetic ink, the typeface is specially designed for reduction to one dimension.

One common example of the application of MICR is in the banking industry for cheque processing [5].

MICR can be seen as the next step towards the more powerful OCR systems available today. To introduce these more powerful OCR systems, it is prudent to examine the basic principles of digital image document processing on which these OCR systems are founded.

2.1.3.Basic Principles of Digital Image Document Processing

The basic principles of digital image document processing can be illustrated by examining the processes which occur in a typical contemporary OCR system.

A typical OCR system can be represented by an elementary OCR model. Figure 2.3 shows the process flow of an elementary OCR model. The elementary OCR model shown is an adaptation of the models presented by Impedovo [3] and by Doermann [2]. This generic model can be used to illustrate the basic processes which occur in one form or another in OCR systems: digitisation, segmentation and classification.

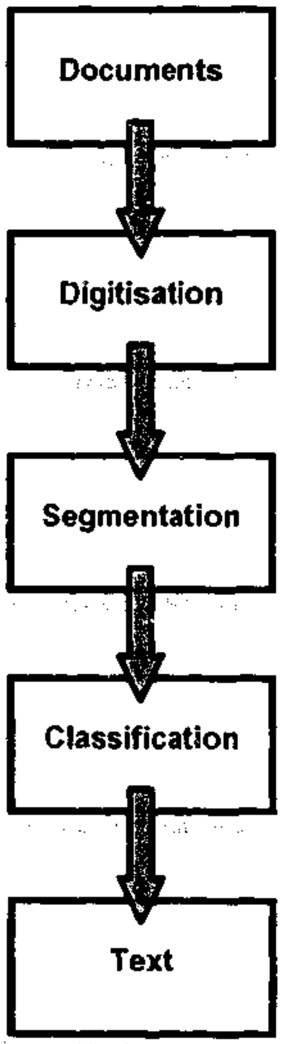


Figure 2.3 Elementary OCR model
The model shows the basic processes of an OCR system, and the order in which they occur as the OCR system transforms physical documents to electronic text.

The physical documents are digitised into an electronic form. Once in a digital form, the documents can be segmented into components (e.g. paragraphs) and sub-components (e.g. words). Segmentation continues until the smallest component, usually a character, is reached. At this stage the components are recognised and converted into text.

The digitisation, segmentation and classification processes are covered briefly in the following sections. Further details of these processes are given in Section 2.5.

2.1.3.1.Digitisation

One of the first processes to occur in a typical OCR system is digitisation. The digitisation process involves the conversion of the document from its physical form into an electronic form suitable for computer processing. Scanning is a typical method of digitising documents which involves passing a document over an array of optical sensors. The facsimile machine is an example of a low resolution bi-level scanner. Further detail about scanners and digitisers are given in Section 2.4.1.

2.1.3.2.Segmentation

Once the document has been digitised into a document image, it is then divided up into areas representing individual character images. This segmentation process typically occurs at different levels. A page might first be divided up into text regions, then divided into columns, paragraphs, lines, words, and finally characters. Where segments such as characters are joined, the decision as to where to break the characters apart will have a significant impact upon the classification stage.

2.1.3.3.Classification

The classification, or recognition, of the character images is one of the last processes to occur in a typical OCR system. The classification process involves comparing the attributes of the character image with a table of character attributes to determine which letter, number or symbol the character image best represents.

Once the basic principles of digital image document processing are understood, it is possible to begin comparing the performance of OCR systems. However, the criteria upon which to base these comparisons have to be established first.

2.1.4.Criteria for Evaluating OCR System Performance

There are several measures for assessing an OCR system's performance [19]. Accuracy, speed and sensitivity are three of the common criteria, or performance classes, used for evaluating OCR systems [10]. There may be other relevant performance classes depending upon the particular OCR system being evaluated. Improvements to a particular performance class of an OCR system can also affect other performance classes [13]. This trade-off between performance in one class for another means that for accurate comparisons of OCR systems, all of the systems performance classes have to be examined. The manner in which these performance classes affect one another is examined in further detail in Chapter 4 and Chapter 5. The first and perhaps foremost criterion for evaluating OCR system performance is the system's accuracy.

2.1.4.1.Accuracy

The accuracy of an OCR system is typically expressed as the percentage of characters recognised correctly [11]. Another method for expressing the accuracy of an OCR system is the percentage of words recognised correctly. The equation for calculating the character accuracy, A, of an OCR system as used by the Information Science Research Institute [20] and others [21] is given in Equation 2.1, where n is the number of characters in the sample and *#errors* is the number of erroneously recognised characters. The character accuracy is a dimensionless ratio expressed as a percentage.

$$A = \frac{n - (\#errors)}{n} \tag{2.1}$$

The accuracy performance class is generally regarded as the most significant measure of an OCR system's performance. It must be remembered though, that an OCR system's accuracy is dependent upon the quality of the documents input to the system. With high quality documents as input, it is quite possible to claim almost 100% accuracy for a particular OCR system. It is therefore important to identify the document quality used to obtain the claimed OCR system accuracy. Another criterion for evaluating OCR system performance is speed.

2.1.4.2.Speed

The speed of an OCR system can be expressed in a number of ways depending on the units of measurement which are chosen. Pages per minute (ppm) or words per minute (wpm) may be useful when comparing OCR system speed with that of a typist. Characters per second (cps) is the unit of measurement used when making comparisons between OCR systems. The equation for calculating the speed, S, of an OCR system (in characters per second) is given in Equation 2.2 [20] where n is the number of characters in the sample, *#errors* is the number of erroneously recognised characters and t is the time taken to process the sample (in seconds).

$$S = \frac{n - (\# errors)}{t} \qquad (2.2)$$

As with the accuracy performance class, the speed performance class can also be dependent upon the quality of the documents. It is therefore important to identify the document quality used to obtain the claimed OCR system speed

2.1.4.3.Sensitivity

The sensitivity of an OCR system represents the degree to which the system can cope with non-ideal documents. Ideal documents can become degraded in a number of physical and electronic ways, such as printing, reproduction, transmission and decay [8], [9]. The sensitivity of an OCR system can be expressed in terms of how well the system can deal with these types of degradations [22], [23]. An OCR system might, for example, claim a certain accuracy and speed when processing a specified type of document.

2.2.HISTORICAL PERSPECTIVE OF OCR SYSTEMS

This section presents an historical perspective of OCR systems from the high volume document environment point of view. Milestones and historically significant developments are discussed and the progression of OCR system performance is examined.

2.2.1.Milestones and Historically Significant Developments

There are many milestones and historically significant developments in the OCR field [5]. From the point of view of high volume document imaging there are three particularly significant developments: OCR character set standards, omnifont OCR, and grey scale OCR.

2.2.1.1.OCR Character Set Standards

The standardisation of OCR character sets is a significant milestone in the history of OCR development because it represented an attempt to alter the way people wrote so that computers could more readily recognise the writing - making people more 'computer-friendly' rather than making computers more 'people-friendly.'

In 1966, two OCR character sets were standardised [6]. OCR A was an American standardised font and OCR B was a European standardised font. These two character sets were designed specifically for use with OCR systems. The OCR fonts were kept simple by avoiding serifs and extraneous detail and had uniform line thickness. They were also designed to miminise similarities between characters, such as '5' and 'S'. Examples of the OCR A and OCR B typefaces are given in Figure 2.4. The differances between the '5' and 'S' in both typefaces can clearly be seen.

ABCabcQRS12345

ABCabcQRS12345

Figure 2.4 Standard OCR character set typefaces
The top typeface shown is the OCR A typeface. The bottom typeface shown is the OCR B typeface. Both typefaces are designed specifically for improved OCR performance.

By using these fonts, the performance of OCR systems can be better than when using fonts designed for human recognition or for aesthetic purposes.

2.2.1.2.Omnifont OCR

The advent of omnifont OCR systems represents another significant development in the OCR field [6]. Omnifont OCR systems refer to OCR systems which can recognise a wide variety of fonts and typefaces. Previous OCR systems required the system to be trained with a particular font before that font could be accurately recognised. This restricted the application of the OCR system to documents whose font types could be controlled.

By developing a recognition system which looked at the features of characters which were font independent, it was possible to develop an omnifont OCR system. An 'o' for instance is represented by a completely enclosed ellipse in most font types. This font independent feature of the letter 'o' can be used to distinguish it from other characters.

2.2.1.3.Grey-Scale OCR

Grey-scale OCR systems represent an important development in the OCR field. These systems used grey-scale images of documents as opposed to previous OCR systems which worked with binary images. Previous OCR systems images were digitised as grey scale and thresholded to binary images, or were simply digitised as binary images. In the process of thresholding the images, potentially useful data about the characters may be lost [24]. Grey-scale OCR systems use this extra data which binary OCR systems discard to improve the recognition accuracy of the grey-scale OCR system [25].

2.2.2.Progression of OCR System Performance

The development of OCR systems over time can be measured in terms of their performance. OCR system performance is limited to a large extent by the computational device used to implement it. It therefore follows that the progress of OCR systems tends to match the progress of computer performance. Table 2.1 is a summary of OCR system performance over approximately 40 years, showing the name of the OCR system, the year it was developed or tested, and its claimed or measured performance in terms of accuracy, speed and sensitivity.

Table 2.1 Progress of OCR performance
This table lists several OCR systems in chronological order including performance ratings from several sources [5], [6], [20], [26], and the year the system was able to achieve those ratings.

OCR System	Year	Claimed or Measured Performance
Solatron ERA	1957	120 cps on typed numerals
Hitachi H8959	Early 70s	95.7% accuracy on constrained handwritten characters
NEC NAS50	1974	93% accuracy on constrained handwritten postcodes
NTT DT-OCR100C	Early 80s	97.1% accuracy on constrained handwritten Katakana characters
Toshiba OCR-V595	1983	99.5% accuracy and 70-100 cps on typed Kanji characters
Caere OmniPage	1988	40 cps on multifont typed characters
Xerox Kurzweil Discover	Late 80s	10-40 cps on multifont typed characters
Fuji Electric Co. XP-70S	Early 90s	30 cps on multifont multilingual typed characters
Sanyo Electric Co. CLL-2000	Early 90s	2 cps on multifont multilingual typed characters
Calera WordScan	1992	99.3% accuracy and 50 cps on multifont typed characters
Expervision RTK	1993	98.1% accuracy on multifont typed characters
Recognita plus DTK	1993	95.57% accuracy on multifont typed characters
Xerox XIS OCR Engine	1994	98.13% accuracy on multifont typed characters

Although the OCR systems of Table 2.1 are listed chronologically, there does not appear to be much discernible progress in OCR system performance over the time span shown. All the accuracy claims, for example, are 93% or better. This appearance of little progress can be partly attributed to the quality of the documents used to measure the OCR systems accuracy. As the quality of the documents is often not stated when quoting performance figures for an OCR system, it can be difficult to compare different OCR systems performance without using the same sample of documents. It is possible for developers to increase the quality of the documents until the OCR systems measured accuracy is acceptable.

2.3. THE STATE OF THE ART

This section examines the state of the art in OCR, as at the time of the writing of this thesis. It gives examples of several commercially available OCR systems and developers and briefly describes their OCR systems. Current areas of research in the OCR field are identified and some centres conducting research in the field are listed.

2.3.1. Commercially Available Systems and Developers

There are a number of commercially available OCR systems and OCR system developers. Caere Corporation and Calera Recognition Systems, Inc. are two prominent examples of companies which develop commercially available OCR systems.

2.3.1.1. Caere

Caere Corporation was founded in 1976 by Dr. Robert Noyce. Its areas of expertise include optical character recognition, intelligent document management and image recognition techniques. In 1988 Caere Corporation produced a PC based OCR software package called OmniPage. Since then the OmniPage package has been revised and improved several times. The OmniPage OCR engine can also be found as part of other vendors' document imaging systems such as those used in facsimile and scanning packages. Caere Corporation also produce a variety of other document imaging equipment including electronic forms processing systems, document storage and retrieval systems, automated data entry equipment and high-end OCR developers kits. Much of the experimental work reported on in this thesis was carried out using various versions of Caere's Omnipage OCR software.

Caere Corporation[i] maintains an Internet site (http:/www.caere.com) which, apart from company and product information, contains several technical publications. These publications include technical papers and theses in the OCR field contributed by Ceare's engineers.

2.3.1.2.Calera

Calera Recognition Systems, Inc. produced a PC based software OCR package called WordScan. The experimental results which were obtained for this thesis, using Caere's Omnipage OCR software, are supported by the results obtained using Calera's WordScan OCR software. WordScan Plus version 4.0 incorporates Predictive Optical Word Recognition (POWR). POWR uses Hidden Markov Models (HMMs) [27-31] to improve recognition accuracy by analysing whole words prior to segmentation into characters.

Calera Recognition Systems, Inc. has since merged with Caere Corporation and are now a wholly owned subsidiary of Caere Corporation. Information about Calera Recognition Systems, Inc. and its products can be obtained by contacting Caere Corporation (refer to section 2.3.1.1).

[i] Caere can be contacted at: Caere Corporation, 100 Cooper Court, Los Gatos, California 95030 U.S.A., Voice: (408) 395-7000, Fax: (408) 354-2743.

2.3.2.Current Research Areas

There are several areas of research in the OCR field which are currently undergoing study and development. Among these are highly degraded document recognition, [6], [29], [31-33], recognition error reduction, [34-38], cursive script recognition, [39-45], and other language recognition, [46-50].

2.3.2.1.Highly Degraded Document Recognition

Currently there is a particular interest in the research community in the area of highly degraded document recognition [29]. While good quality documents can be accurately recognised, the lower recognition accuracy of poor quality documents can be seen as offering considerable scope for improvement [31].

The more a document departs from its ideal form the more difficult it can become to recognise that document. There are several ways for a document to become degraded. Most document printing, transmission and reproduction results in some form of degradation in document quality. Many different methods are employed to cope with the various types of document degradation [25].

When documents are too highly degraded such that no method is capable of compensating sufficiently for the degradation, recognition errors are likely to occur. These recognition errors could benefit from error reduction techniques.

2.3.2.2.Recognition Error Reduction

Recognition error reduction is another area currently attracting research work [51], [52]. In every OCR system there exists the possibility of incorrectly recognising characters in a document. Recognition error reduction techniques seek to reduce the number of incorrectly recognised characters which occur. Simple dictionary based spelling correction is one method which can be used to reduce these recognition errors. Grammatical rules and other *a priori* knowledge methods can also be applied to further reduce recognition errors.

2.3.2.3.Cursive Script Recognition

Cursive script recognition considerably extends the scope of documents which are recognisable by computer and is another area currently attracting research work [41]. There are two significant differences between the recognition of handwritten cursive script characters and the recognition of typed characters [53]. The first difference is that cursive script characters are joined to adjacent characters. The second difference is that there is a greater degree of variation in the way handwritten cursive script characters can be written.

Cursive script recognition provides an extra level of difficulty during the segmentation process because characters touch the other characters on either side [39], [40]. A decision has to be made as to where one character ends and another begins. If the wrong choice is made as to where to segment the touching characters, then the characters are likely to be incorrectly classified.

An alternative to segmenting cursive script is word recognition [39], [54]. Words are not segmented into characters but are, instead, treated as whole units for recognition purposes [55], [56]. One consequence of word recognition is the greater number of objects which have to be classified. Character recognition of English documents has to deal with 26 letters, while word recognition of English documents has to deal with many thousands of words.

The variance of handwritten cursive script requires a whole new set of problems to be solved [41], [44], [53], [57], [58]. Training of a classifier has been shown to be an effective method of recognising a particular individual's writing style. To date, the accuracy of omnifont recognition of handwritten cursive script, without classifier training, has limited widespread application of the method. However, it is a rapidly developing technique.

2.3.2.4.Other Language Recognition

Particular attention is being paid to recognising languages based on more complex character sets [36], [48], [49], [50], [59]. The Japanese and Chinese languages make use of character sets which are more numerous and more complex than languages based upon the English alphabet. There are numerous other differences between these languages and English which also increase the difficulty of recognising these languages.

2.3.3.Research Centres

There are a number of research centres around the world conducting research into document imaging. Two research centres in the USA provide Internet access and contain a wealth of readily downloadable information including technical reports, research papers and even complete doctoral theses. Another research center in Australia which focuses on research application is also described.

2.3.3.1.Information Science Research Institute

The Information Science Research Institute (ISRI) was established by the USA Department of Energy at the University of Nevada, Lass Vegas in 1990. ISRI's stated mission is to foster the improvement of automated technologies for understanding machine-printed documents. Each year ISRI sponsors a Symposium on Document Analysis and Information Retrieval (SDAIR) and produces an annual research report on its OCR Technology Assessment program. ISRI's web page (http://www.isri.unlv.edu) contains ISRI publications, technical reports, and annual research reports as well as other information about ISRI[i].

2.3.3.2. Center of Excellence for Document Analysis and Recognition

The Center of Excellence for Document Analysis and Recognition (CEDAR) was established by the United States Postal Service at the State university of New York at Buffalo in 1991. Initially, CEDAR focused on scanned postal-relevant documents but has since diversified to include the reading of fax documents, forms and cheques and printed documents with complex layouts. CEDAR's web page (http://www.cedar.buffalo.edu/index.html) contains information on its current and past projects, CEDAR publications and resources and CEDAR[ii] personnel.

2.3.3.3. Digital Imaging Applications Centre

The Digital Imaging Applications Center (DIAC) was established by the School of Engineering at Monash University Gippsland Campus in 1991. DIAC is a multidisciplinary, application oriented research and development center focusing on infra-red imaging, image processing, knowledge

[i] ISRI can be contacted at: Information Science Research Institute, University of Nevada, Lass Vegas, 405 Maryland Parkway, Box 454021, Lass Vegas, Nevada 89154-4021 USA Voice: (702) 895-3338, Fax: (702) 895-1183, Email: isri-info@isri.unlv.edu.

[ii] CEDAR can be contacted at: Center for Excellence for Document Analysis and Recognition, UB Commons, 520 Lee Entrance, Suite 202, Amherst, NY 14228-2567 (USA), Voice: (716) 645-6162. Fax: (716) 645-6176.

engineering, computer graphics electronic data exchange and software engineering. DIAC[i] maintains a web page (http://giaeb.cc.monash.edu.au:80/~briangr/public.html) which contains information on DIAC research and development projects, DIAC publications, and DIAC personnel.

2.4.KEY ENABLING TECHNOLOGIES

This section covers the key enabling technologies without which digital image processing could not be implemented. Digitising devices, high capacity optical storage and high speed image processors are described in terms of their impact on digital image processing in the high volume document environment.

2.4.1.Digitisers

The ability to transform a document from its physical form into an electronic form is a key enabling technology for document imaging. This transformation process is called digitising or scanning and is described in Section 2.1.3.1.

There are several digitising methods, including divided slit scan, laser beam scan, photocell matrix scan and mechanical scan. The divided slit scanner is the digitising method most commonly used for OCR systems [3]. It involves the document being passed over an array of photoelectric devices by a transport mechanism or the array being passed over the document.

One of the important components of digitisers is the document transport mechanism. The document transport mechanism's ability to quickly and accurately move the documents over the scanning head is crucial to consistently digitised documents.

[i] DIAC can be contacted at: Digital Imaging Applications Centre, Monash University Gippsland Campus, Churchill, Victoria 3842 (Australia), Voice: +61 (051) 22-6461, Fax: +61 (051) 22-6500.

Hand and wand scanners are examples of smaller divided slit scanners which can also be used to digitise parts of documents for OCR systems. Because these scanners rely on the operator to manually move the photoelectric array over the document, they are not suitable for the high speed, high volume environment.

Digitisers and their performance are important to digital image processing in the high volume document environment because the following stages of processing and their performance are reliant on the accuracy of the digitisation process. Digitisation errors such as image skew and image offset reduce performance and must be corrected or compensated for in the following processes. Examples of the effects of digitisation error on digital image processing in the high volume document environment is given in Chapter 7 and Chapter 8.

2.4.2.High Capacity Optical Storage

High capacity optical storage devices are a key enabling technology for digital image document processing. Although text storage requirements are fairly modest by today's data storage standards (e.g. 2000 bytes per page [60]) the storage requirements for high definition document images are orders of magnitude greater (e.g. 467,500 bytes for 8.5 x 11 inch page at 200DPI, binary image [60]). Even with compression reducing image sizes to one tenth normal size, the storage capacity of a 635MB CD-ROM is only 13,500 images (at 467,500 bytes per image prior to compression).

High capacity optical storage devices and their performance are important to digital image processing in the high volume document environment because they limit the storage capacity of documents and the time taken to store and access documents. Examples of the effects of high capacity optical storage devices on digital image processing in the high volume document environment is given in Chapter 7 and Chapter 8.

In the high volume environment the storage requirements for document images Optical storage technology offers several advantages over magnetic storage technology and is used in many document imaging systems [60]. Optical storage technology offers potentially greater storage capacity, portability and data security. Three common types of high capacity optical storage drives are write once read many drives, compact disc recordable drives and magneto optical drives. Optical disk jukeboxes allow many optical disks to be available to an optical drive, effectively multiplying the capacity of the optical drive by the number of disks in the jukebox.

2.4.2.1.Write Once Read Many Drives

The Write Once Read Many (WORM) drive is an early form of optical storage. The storage capacity of the WORM disk varies according to the disk diameter. Five and a quarter inch WORM disks can store 940MB, twelve inch disks can store 9GB, and fourteen inch disks can store 10.2GB [60]. The fact that data could not be altered once written provides security for the data. Access times and transfer rates for WORM drives are slightly slower than for magnetic storage media.

2.4.2.2.Compact Disc - Recordable

The Compact Disc - Recordable (CDR) is a more evolved and standardised form of optical drive than the WORM Drive. The CDR is identical in most respects to the standard audio compact disc played on home stereo systems. Like the WORM drive, once data is written to the CDR, it cannot be altered. The nominal capacity of a typical 4.7 inch CDR is 635MB [60]. Smaller 3.5 inch CD-ROM disks can hold up to 180MB [60]. CDR disks are read by CD-ROM drives, with typical access times of 300ms [60] and transfer rates from 150Kb/s (single speed drive) to 1800Kb/s (twelve speed drive).

New disc based ROM technology such as the Digital Versatile Disc (DVD) [61] sets even higher standards for data storage. A DVD can store 4.7GB of data and has an access time of 142.3ms.

2.4.2.3.Magneto Optical Drives

By combining magnetic and optical technology, a Magneto Optical (MO) drive enables data to be re-written to the disk many times. These are hybrid drives and have the high capacity storage of optical drives as well as the re-writablity of magnetic drives. The storage capacity of an MO disk varies according to the disk diameter and data density. Three and a half inch MO disks can store between 128MB and 256MB [60], 5.25 inch disks can store between 256MB and 650MB. Typical access times for MO drives are about 40ms [60] and transfer rates are slightly slower than WORM drives.

2.4.3.High Speed Image Processors

High speed image processors are another key enabling technology for digital image document processing. The performance of an OCR system is limited to a large extent by the computational processors used to implement the OCR system. These high speed image processors can be either dedicated image processors or general purpose processors.

2.4.3.1.Dedicated Image Processors

Dedicated image processors can implement a number of the OCR systems functions in hardware. While this may result in better OCR performance initially, the OCR system may be difficult to upgrade or alter.

One example of a dedicated image processor is a wand scanner which recognises single lines of text and outputs the data as a stream of keyboard codes. This effectively replaces or supplements keyboard data entry.

Another, more specific, example of a dedicated image processor is Calera's Truescan. Truescan is a PC board containing customised chips and a Motorola 68020 processor and is capable of OCR at 100 cps.

2.4.3.2. General Purpose Processors

For this discussion I shall define general purpose processors as those used in personal computers running DOS, Windows, Unix or Apple operating systems.

High speed general purpose processors can be used to implement OCR system functions in software. The software based OCR system can be more easily updated and modified than the hardware based OCR system. It can therefore be adapted to a wider variety of applications.

There are many examples of OCR systems based on general purpose processors, such as Omnipage and Wordscan which are designed for use with IBM compatible PCs.

2.5. OVERVIEW OF THEORY

This section presents an overview of the theories applicable to digital image document processing. Theories covering digitisation, thresholding, normalisation and segmentation are covered, as are morphological operations and noise filtering. Different approaches to error correction are also included in the overview of the theories. Examples of applications of the theories to document images are included throughout the overview.

2.5.1.Digitisation

As digitisation is one of the first processes to occur in a digital image document processing system, as previously described in Section 2.1.3.1 and Section 2.4.1, it is the first theory to be overviewed. A documents physical image can be expressed as a continuous illumination function *f(x,y)* of two co-ordinates in the image plane [62]. Digitisation samples the continuous function *f(x,y)* into a discrete sampled function *s(x,y)*. The discrete sampled function can be represented by a pixel matrix of *M* rows by *N* columns. The size of the pixel matrix compared to the physical image is referred to as the sampling resolution. The continuous range of the image function *f(x,y)* is quantised into *K* intervals of image intensity.

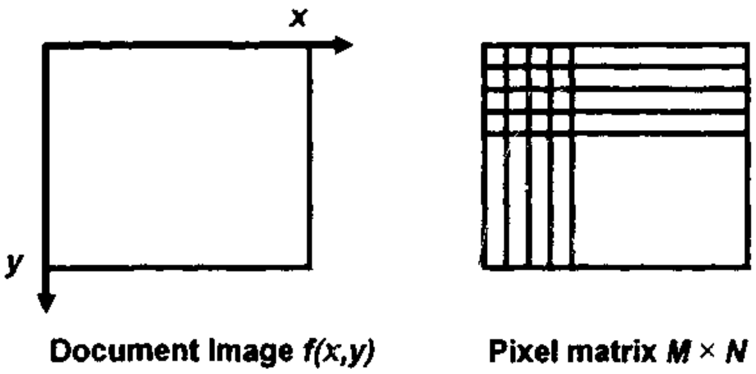


Figure 2.5 Continuous and sampled image maps
The continuous image map (represented by a document image function *f(x,y)*) on the left is digitised into the sampled image map (represented by a pixel matrix *M* × *N*) on the right.

The relationship between the document's physical and digital image is illustrated in Figure 2.5. The pixel matrix on the right approximates the document image on the left. An example of the digitisation process is shown in Figure 2.6 for an image of the letter 'E' in a Times Roman typeface. The image of the left represents a continuous function and the image on the right represents the digitised image. Both images are shown enlarged to illustrate the difference between them. The effects of discrete sampling and quantisation can clearly be seen in Figure 2.6 and are explained in Section 2.5.1.1. and Section 2.5.1.2.

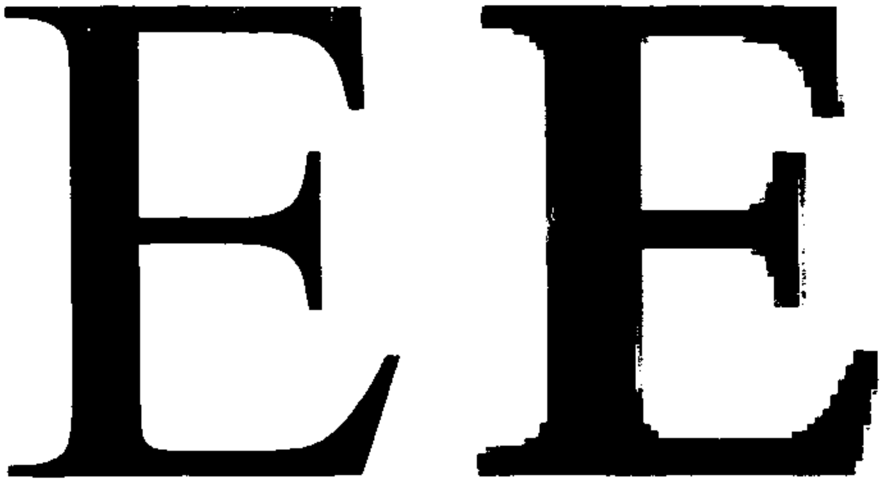


Figure 2.6 Continuous and sampled image example

The continuous image (on the left) of the letter 'E' in a Times Roman typeface is digitised into the sampled image (on the right).

2.5.1.1.Sampling Resolution

The sampling resolution is the ratio of the size of the pixel matrix to the size of the physical image, and can be expressed as pixels or dots per inch. The finer the sampling resolution (the larger the values of M and N) the better the pixel matrix approximates the physical image [63]. Typical sampling resolutions for document image processing range from 200 to 400 DPI (Dots Per Inch) [64].

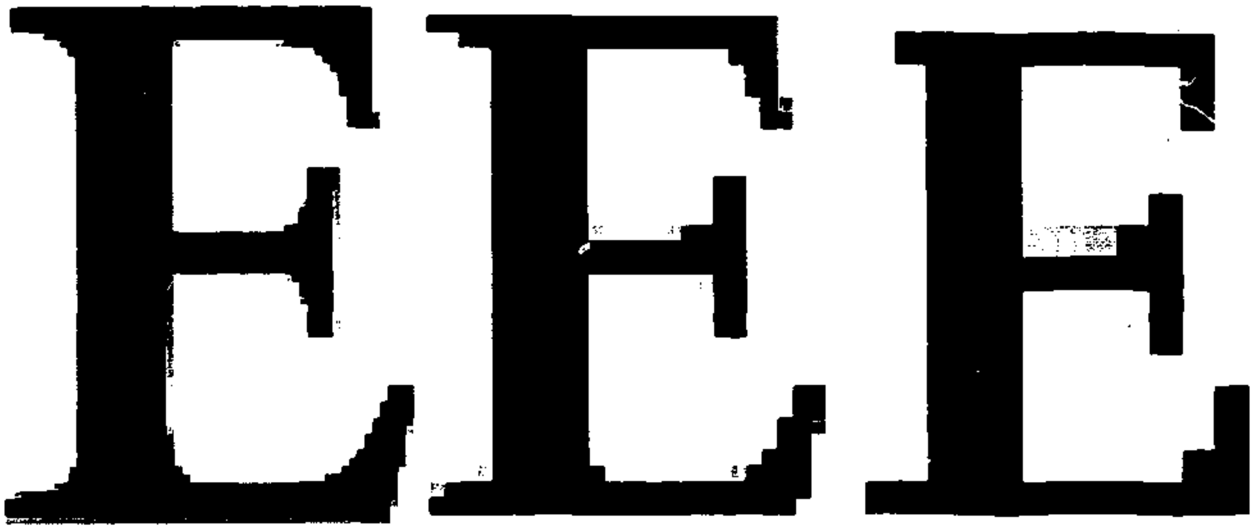


Figure 2.7 Sampling resolution example

These three images (from left to right) show the letter 'E' sampled at 400, 200, and 100 DPI respectively, and are enlarged to show detail. The loss of detail which occurs with lower sampling resolutions is clearly shown.

An example of the effects of different sampling resolutions upon character images is given in Figure 2.7. The left image is sampled at 400 DPI, the center image at 200 DPI and the right image at 100 DPI. All images are shown enlarged to illustrate the differences between them. The reduction in image quality as the sampling resolution decreases is clearly shown. Any further reduction in sampling resolution could reduce the image quality so that the character is unrecognisable.

2.5.1.2.Quantisation

The process of rounding the image function *f(x,y)* range into *K* integer values is called quantisation [63]. The finer the quantisation (the larger the value of *K*) the better the pixel matrix approximates the physical image. Typical quantisation ranges for document image processing are from 2 levels to 256 levels [65]. The quantisation error *q* which occurs due to the rounding process is given in Equation 2.3 [62].

$$q \leq \frac{1}{2K} \tag{2.3}$$

An example of the effects of different quantisation values K upon character images is given in Figure 2.8. The left image has been quantised with a K value of 16. The center image has been quantised with a K value of 8. The right image has been quantised with a K value of 4. The character is clearly visible in all three images.

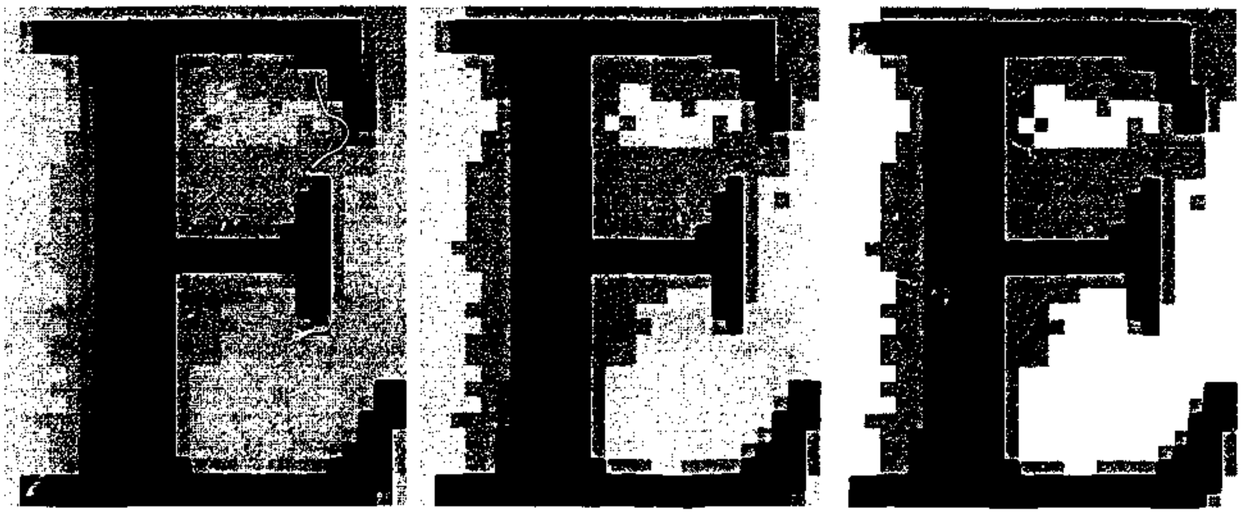


Figure 2.8 Quantisation example
These three images (from left to right) show the letter 'E' quantised to 16, 8 and 4 levels respectively. There is little loss of apparent detail in the character as a result of lower quantisation level.

2.5.2.Thresholding

Thresholding reduces the quantisation range K to 2 levels. It changes the document's digital image into foreground (text) and background. Thresholding is the transformation of an input pixel matrix $s(x,y)$ to an output pixel matrix $t(x,y)$, as shown in Equation 2.4 [63].

$$\begin{aligned} t(x,y) &= 1 \quad \text{for} \quad s(x,y) \geq T \\ t(x,y) &= 0 \quad \text{for} \quad s(x,y) < T \end{aligned} \tag{2.4}$$

The selection of the threshold value T effects which parts of the image are labelled as foreground, and which parts are labelled as background. If the threshold value T is too high, character images in the output pixel matrix $t(x,y)$ may start to merge with each other. If the threshold value is too low, gaps may start to appear in the character images.

An example of the effects of different threshold values T upon character images is given in Figure 2.9. The left image has been thresholded with a too high a value of T. Parts of the image have merged to obscure the character. The center image has been thresholded with a correct value of T. The whole character is clearly recognisable. The right image has been thresholded with too low a value of T. Parts of the character are no longer visible as a consequence.

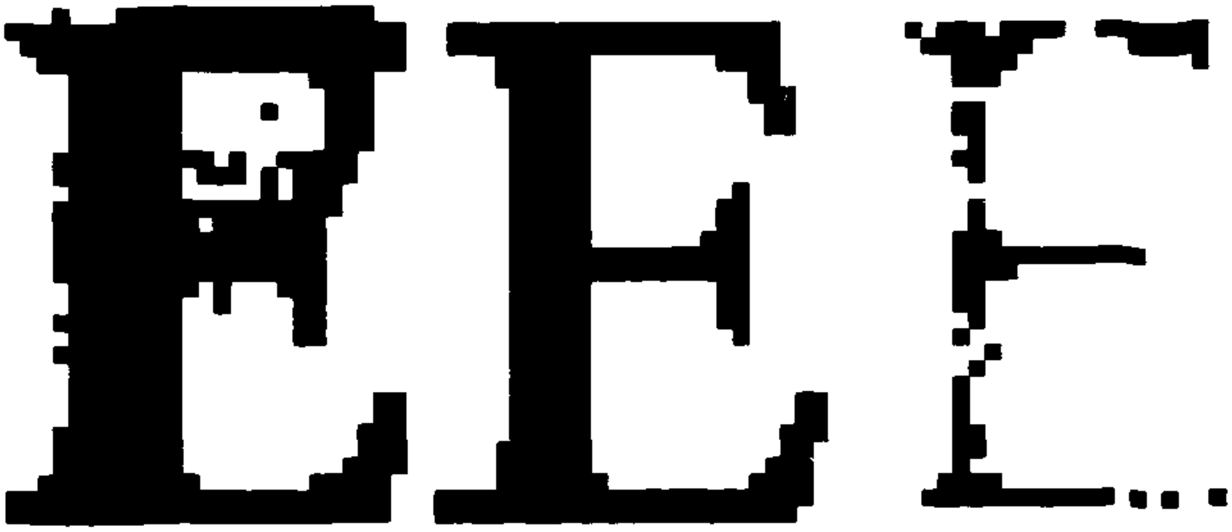


Figure 2.9 Thresholding example
These three images (from left to right) show the letter 'E' thresholded at high, medium and low levels respectively. The joining effect of a high thresholding level is clearly evident, as is the breaking effect of a low thresholding level.

2.5.3.Normalisation

In the context of OCR processing, normalisation refers to the reduction of the character image to a uniform size and slant [66]. Its purpose is to standardise the input for the processes that follow. An example of the normalisation process is shown in Figure 2.10. The image on the left is a slanted character image of larger than standard size character. When the character image is normalised, the slant is corrected and the character image is reduced to a standard for further processing. In this case the standard size is about half the size of the original character image. The image on the right shows the slant corrected and size adjusted character image.

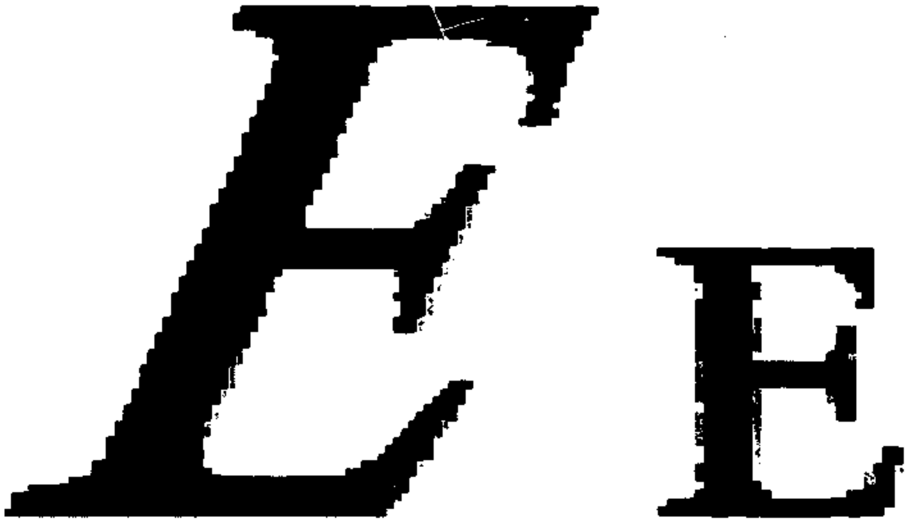


Figure 2.10 Normalisation example
The image on the left represents a character image which is slanted. The image on the right shows the character image after slant correction and resampling to a uniform size.

A comparison between the normalised character (Figure 2.10 right image) and a character not requiring normalisation (Figure 2.7 center image) reveals degradation in the normalised character image. The degradation of the image on the right is a result of the slant correction and resampling process.

2.5.4.Segmentation

Segmentation is the process that determines the components of the document image [67], [68]. The segmentation process is illustrated in Figure 2.11 which shows a page being segmented into characters. Although only two divisions are shown per stage, many divisions are possible, eg. several characters per word.

Segmentation is used firstly to distinguish areas of text from non-text areas and, secondly, to break down the text areas into columns, lines, words, and characters [69]. It is necessary to make the distinction between the text and non-text areas so that the second segmentation phase can operate exclusively on text regions [70]. Non-text regions can comprise graphics, diagrams, pictures and symbols such as mathematical equations.

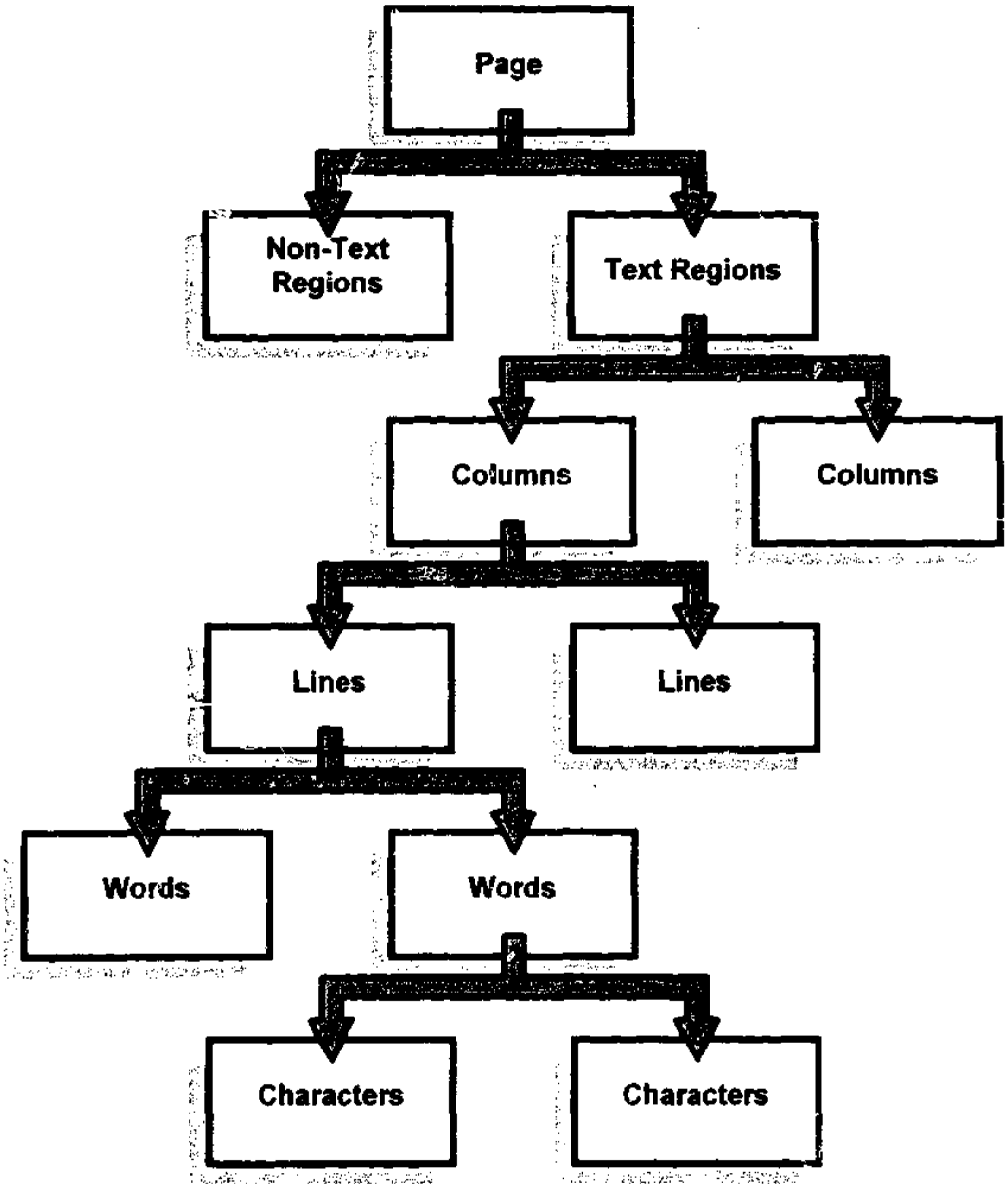


Figure 2.11 The segmentation process
The segmentation process begins with the page being separated into text and non-text regions. The text regions are successively broken down into columns, lines, words, and finally characters.

Any region determined to be non-text does not proceed to further processing stages. If a text region is incorrectly identified as non-text, its OCR accuracy will, by default, be zero per cent because it is not passed to the classification stage. It is therefore important from an OCR accuracy performance viewpoint to correctly identify text regions. If a non-text region is incorrectly identified as a text region, it will be passed to the classification stage. Because the region contains non-text, the classification stage will use processing time trying to classify characters that are not there. It is therefore important from an OCR speed performance viewpoint to correctly identify non-text regions. An example of the first segmentation phase is given in Figure 2.12. The newsletter is shown thresholded but unsegmented on the left. On the right the newsletter has been segmented into text and non-text regions. The lines above and below the title and the two photos have been identified as non-text regions by the segmenting process.

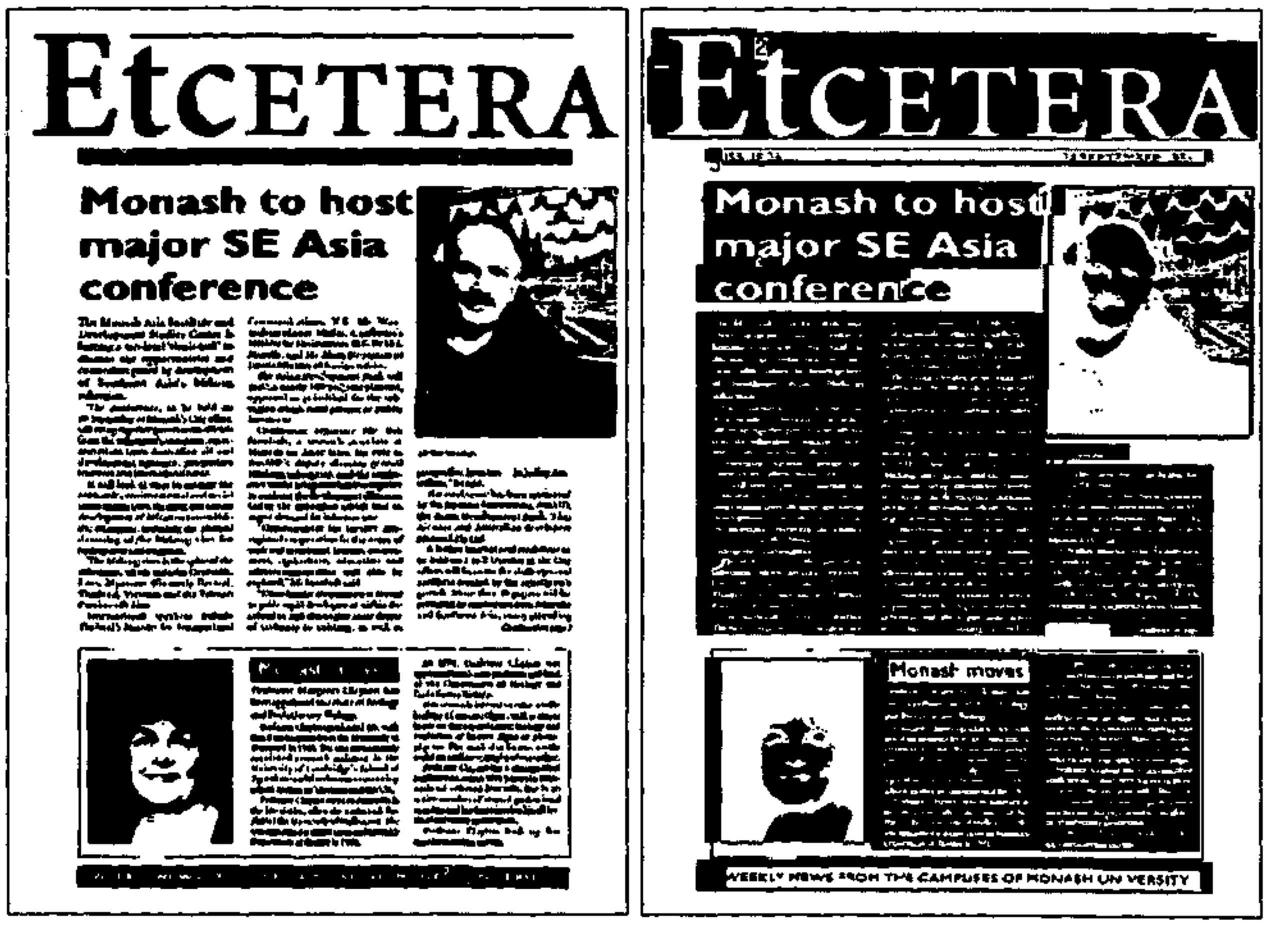

EtCETERA
Monash to host major SE Asia conference
EtCETERA
Monash to host major SE Asia conference
Monash moves
WEEKLY NEWS FROM THE CAMPUSES OF MONASH UNIVERSITY

Figure 2.12 First segmentation phase example
The newsletter on the left is segmented to produce the series of numbered text and non-text regions on the right.

The second segmentation phase breaks down the text regions into columns, lines, words and finally characters. The success of this segmentation phase affects the success of the following classification stage. Incorrect segmentation of words into characters can lead to partial or joined characters being passed to the classifier. The success of this segmentation phase is affected by a number of factors, including font type, thresholding and noise.

If the font is detailed or tightly kerned such that characters touch or character regions overlap, it can increase the difficulty of the second segmentation phase. Poor thresholding can also lead to touching characters or breaks in thin lined characters which make segmentation difficult. Noise, which has not been properly filtered from the image, can also hinder segmentation.

2.5.5.Noise Filtering

Noise can be introduced to a document by a number of means, such as reproduction and transmission [71]. Noise can be partially or wholly removed from document image by applying a noise filter to the image. The noise filter is designed in such a way as to reduce as much noise as possible while retaining all of the original signal. Morphological operations and convolution are two methods of filtering noise from document images.

2.5.5.1.Morphological Operations

Morphological processing methods can be used to filter noise in binary digital images [72], [73]. They can also be used to compensate for poor thresholding. The two basic morphological operations are erosion and dilation. Erosion reduces the size of foreground regions in the binary image, effectively peeling off the outer layer of foreground pixels. Dilation increases the size of the foreground regions, effectively growing an outer layer of foreground pixels.

These two basic morphological operations can be combined into more complex opening and closing transforms. The effect of the opening and closing transforms is determined by the number of erosion and dilation iterations performed and their order. The opening transform can be used to smooth character image boundaries, break thin joins between character images and remove noise dots around the character image. An example of the opening operation is given in Figure 2.13. In this example, the character image is the highly thresholded image from Figure 2.8. The image on the left is eroded to produce the center image. The center image is dilated to produce the image on the right. The merging effects of high thresholding have been corrected to some extent in this example to produce a more easily recognisable character image. The smoothing of the image boundary and elimination of the noise can also be seen in the example.

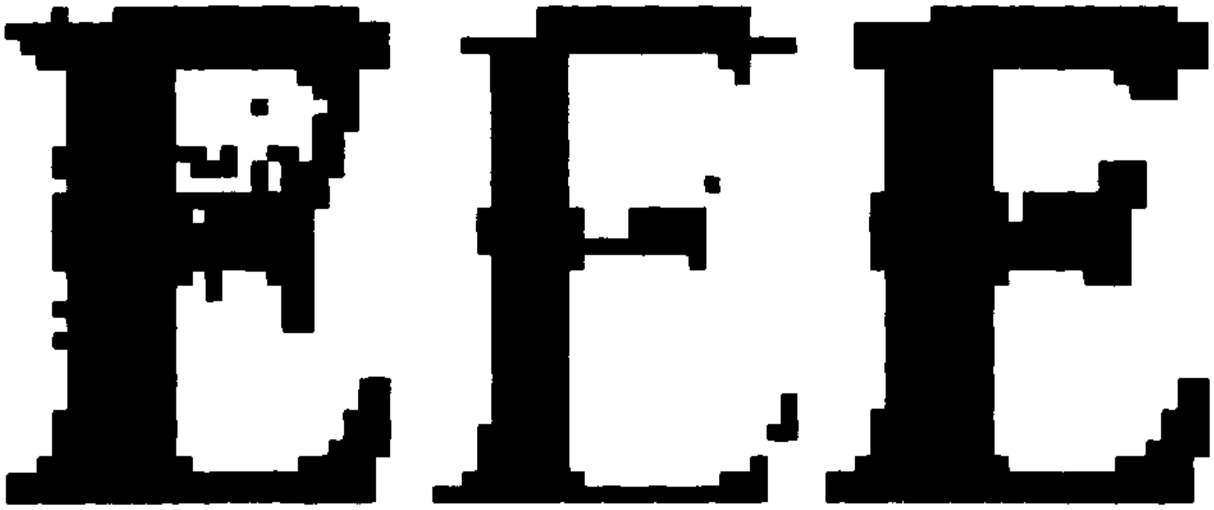


Figure 2.13 Opening operation example

The image of the letter 'E' on the left has a joined section, in this case as a result of a high threshold level. By performing an erosion operation (producing the center image) followed by a dilation operation (producing the image on the right), the joined section is opened. The erosion operation followed by the dilation operation combine to form an opening transform.

The closing transform can be used to smooth character image boundaries, join gaps in a character image and fill noise holes in a character image. An example of the closing transform is given in Figure 2.14. In this example, the character image is the lowly thresholded image from Figure 2.8. The image on the left is dilated to produce the center image. The center image is eroded to produce the image on the right. The breaking effects of low thresholding have been corrected to some extent in this example to produce a more easily recognisable character image.

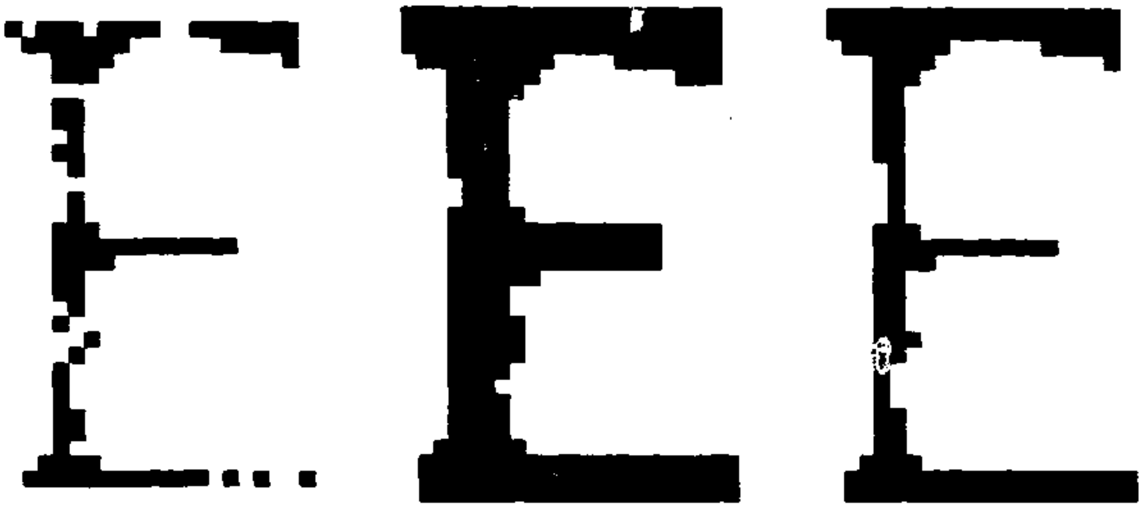


Figure 2.14 Closing operation example
The image of the letter 'E' on the left has several broken sections, in this case as a result of a low threshold level. By performing a dilation operation (producing the center image) followed by an erosion operation (producing the image on the right), the broken section is closed. The dilation operation followed by the erosion operation combine to form a closing transform.

2.5.5.2. Convolution

Convolution methods can be applied to binary digital images to filter noise from the image. By passing a *m*×*n* window or kernel over the document image pixel matrix *t(x,y),* various types of noise can be filtered, depending on the kernel contents. A simple averaging filter with a 3×3 kernel h_1 is shown in Equation 2.5 [63].

$$h_1 = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{2.5}$$

A filter designed to remove Gaussian noise might use a kernel h_2 as shown in Equation 2.6 [63].

$$h_2 = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \tag{2.6}$$

An example of the effects of applying the noise removing filter with kernel h_2 is shown in Figure 2.15. The image on the left shows a character image with noise added. After applying the noise removing filter the character image is transformed to the image on the right. From the image on the right it can be seen that all the noise has been removed from the character image.

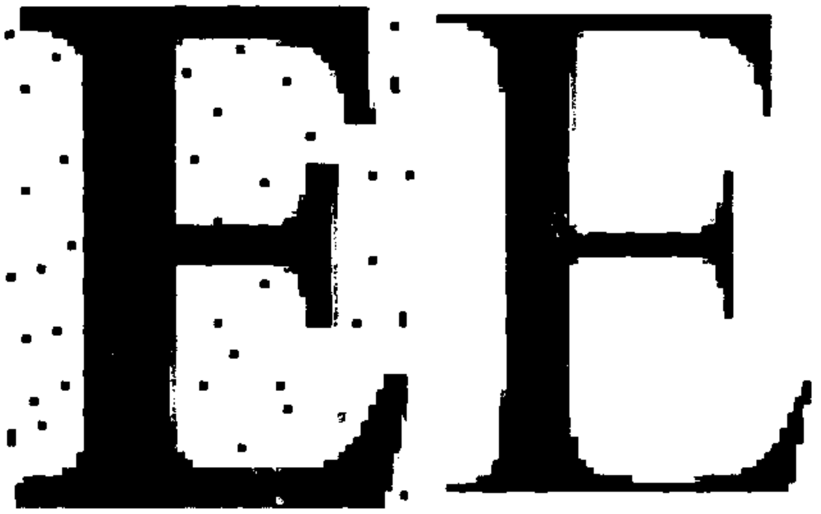


Figure 2.15 Noise removing filter example

The left image shows a character image with noise. After applying a noise filter the image on the right is produced.

2.5.6.Classification

The classification of character images as particular letters, numbers or symbols is a critical process for any OCR system [67]. The performance of an OCR system is perhaps more dependant upon the classification stage than any other stage in the OCR process. As a result of this dependence, numerous methods have been developed to improve the classification stage and thereby increase the performance of the OCR system.

Some classifications methods have been designed for specific types of characters such as handwritten characters. These methods do not perform as well on machine printed documents as those methods designed specifically for classifying machine printed characters [74].

The classification methods for machine printed characters were developed prior to those for handwritten characters. This is because the variation in shape of machine printed characters is less than that for handwritten characters. The lower shape variation allowed simpler classification methods and less powerful hardware to be used.

2.5.6.1. Template Matching Method

The template matching method is one classification method applicable to objects with low shape variation, such as machine printed characters [5]. MICR is an example of the template matching method which is described in section 2.1.2.2 and illustrated in Figure 2.1. MICR reduces the two dimensional information into a one dimensional array. This array can be matched against a table of one dimensional arrays to classify the image as a particular character.

2.5.6.2. Feature Matching Method

Feature based matching is another classification method similar to the template matching of MICR [75]. It uses vertical (as per MICR) as well as horizontal projections of characters. The major features of the characters, such as vertical or horizontal strokes, appear as peaks in the two projections. The position, size, shape and number the peaks in the two projections are used to classify the image as a particular character.

2.5.6.3. Peephole Method

The peephole method is another template matching classification method suitable for machine printed characters [5]. A template with holes is placed over the characters as shown in Figure 2.16. If a hole covers part of the character image, it is classed as a foreground hole, otherwise it is classed as a background hole. The foreground and background holes form an array which can be used to classify the image as a particular character. The placement of the holes in the template is designed to provide maximum distinction between the character images to be classified.

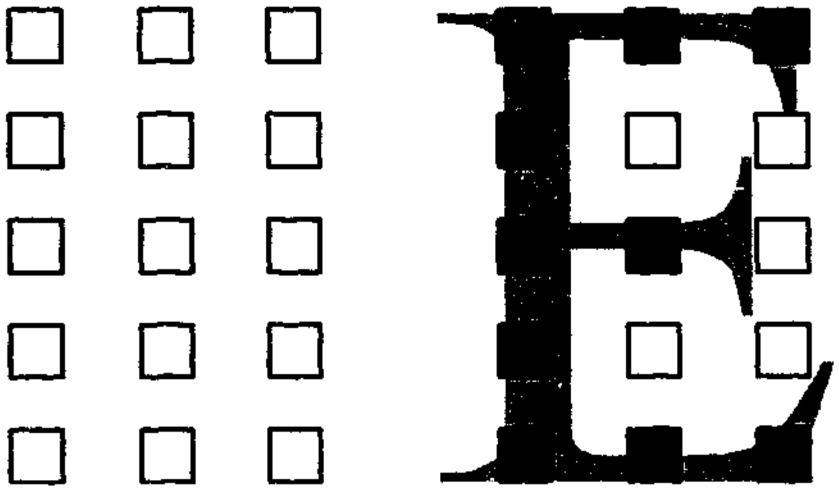


Figure 2.16 Peephole classification method example
When the template of peepholes shown on the left is placed over the character image of the letter 'E' on the right, some of the holes show the letter beneath (black foreground holes) and some do not (white background holes). This array of black and white holes can be used to classify the character image.

2.5.7.Error Correction

The possibility of errors occurring in a document OCR system are always present [48]. The more the input document quality departs from the ideal document quality, the more likely recognition errors are to occur. To improve the overall accuracy of an OCR system, some sort of error correction is desirable to detect and reduce the percentage of recognition errors. The use of contextual information within the document, and other *a priori* knowledge of the document can be used as a basis for error correction.

By examining the context in which a particular character occurs, it is possible to detect whether the character has been erroneously recognised, and to correct that character. Dictionary methods and more complex grammatical methods can use the contextual information within a document to correct recognition errors which occur in that document.

2.5.7.1.Dictionary Methods

The use of dictionary methods for OCR error correction has been shown to be simple and effective [76]. As with the use of dictionaries for spelling correction in word processors, words which are erroneously transformed into other valid words will not be detected. Once an incorrectly recongised word has been detected there are several possibilities. The word can be automatically changed to the most similar word in the dictionary or the word and accompanying word image can be brought to the attention of the human operator for correction. There is no guarantee that the most similar word in the dictionary will be the correct interpretation of the word image. More grammatical information is needed to correct these errors.

2.5.7.2.Other *A Priori* Knowledge Methods

A priori knowledge of the document can be used to detect and correct recognition errors in the document. If, for example, the document is a form where fields have certain attributes, then these attributes can be used to assist recognition error correction. A company name field and company number field might be compared to an on-line company database to verify the correct recognition of both fields. Even knowing whether a field contains only alpha or only numeric text can assist error correction. As with other error correction methods, the errors detected may be corrected automatically or brought to the attention of the human operator for correction.

2.6. DIGITAL IMAGE PROCESSING IN THE HIGH VOLUME DOCUMENT ENVIRONMENT

This section places particular emphasis on the application of digital image processing to the high volume document environment, which is the main subject of this thesis. The unique aspects of this area of digital image processing are examined, and attention is given to previous work in this field.

2.6.1. Unique Aspects

There are a number of unique aspects of digital image processing in the high volume document environment. One is the way in which system performance and optimisation affects the digital imaging processes. The significance of the OCR system modelling and quality control are also unique aspects of digital image processing in the high volume document environment.

2.6.1.1. Performance Optimisation

Particular emphasis is placed upon the performance of the digital image processing system in the high volume document environment and the optimisation of that performance. The system performance may be described in terms of three classes - accuracy, speed and sensitivity. The connection between the three performance classes is an important aspect of the research work in the high volume environment. Understanding the connection between the accuracy, speed and sensitivity performance classes is critical to the optimisation of the OCR system in the high volume document environment.

2.6.1.2. System Modelling

The modelling of the document processing system is another aspect of the work which is unique to the high volume environment, and is described in further detail in Chapter 7.

The models are necessary for performance optimisation and show the significance of quality control to the high volume document environment. The control and analysis of document and data quality leads to a confidence in the information system fed by the high volume document processing system.

2.6.1.3.Quality Control

Quality control is another unique aspect of document processing systems in the high volume environment. Control of document quality occurs at three distinct stages in the document processing system - as physical documents, document images and document text. The quality control procedures are incorporated into the OCR system models reported in Chapter 7.

2.6.2.Previous Work in the Area

There are a number of examples of previous work carried out in the area of digital image processing in the high volume document environment. All of the experimental work and modelling work undertaken for this thesis is based on, and adds to, the research reported in the available literature. It should be noted that because this area is often linked to commercial interests, there appears to be some reluctance to openly publish findings.

The preliminary experimental work reported on characteristic variables in Chapter 4 is based upon Baird's work on document image defect models [8] and their uses [9]. Baird's work on the effects of image resolution and image skew are extended by Griffin's work on OCR performance [10] and performance optimisation [13] in the high volume document environment.

The development of the OCR system model in Chapter 7 draws its origins from the works of Srihari [15] on high performance reading machines and Casey [14] on intelligent forms processing systems.

Other OCR system models presented by Impedovo [3] and by Doermann [2] have also influenced the development of some of the more sophisticated, high volume, OCR system models in Chapter 7.

Further analysis of previous work in the research area is conducted in Chapter 3 using a literature analysis database package called LitBase.

CHAPTER 3

LITERATURE ANALYSIS USING "LITBASE"

CHAPTER CONTENTS PAGE

3. LITERATURE ANALYSIS USING LITBASE

This chapter analyses the research literature using the custom built LitBase literature analysis package. The LitBase program is introduced and followed by a detailed description of the development of the LitBase system. An analysis of the literature as performed by LitBase is presented, highlighting interesting results. An analysis of the LitBase package itself is also reported. Observations of the performance of the LitBase package support those made in the literature analysis, and lead to the experimental work in Chapter 4.

3.1.INTRODUCTION

This section introduces LitBase by looking at the motivation for developing the LitBase system and examining the modular design of the system.

The LitBase program is a literature database analysis package developed by the author for two main purposes. The primary purpose of the LitBase program was to store and analyse the research literature upon which the thesis is based. The secondary purpose was to provide the author with a test-bed OCR system to form the basis for experimental analysis of OCR system architecture. Both of these purposes for developing LitBase led to a modular design which was quick to build and easily upgraded.

The hardware and software necessary to implement the LitBase system are described in detail in Section 3.1.3.

3.1.1.Development Motivation

This section examines the motivation for development of the LitBase system. The majority of the existing electronic document storage and retrieval systems which were examined were found to be fairly general in nature and would have needed modification to suit the specific requirements of the proposed literature database and analysis system. Rather than modifying an existing system and work within its limits, the decision was made to develop a new literature database and analysis system, called LitBase, which could meet all the specific requirements and be more easily modified than an existing system. Another motivation for developing LitBase was to provide the author with experience in developing a complete OCR-capable electronic document storage and retrieval system. The insight gained into OCR systems by developing LitBase easily justified the effort spent developing the LitBase system.

3.1.2.Modular Design

The LitBase package is composed of a number of modules which are all controlled by a database system developed using Microsoft Access on a Pentium based personal computer running Microsoft Windows. These modules consist of: an image viewer, scanning and OCR module, indexing and search system, and a text display program. LitBase enables the operator to store and analyse research literature by several methods:

- LitBase can be used to scan, store and index research papers as a series of images stored on disk.
- LitBase can perform OCR on these images to convert the images into ASCII text which is also stored to disk and indexed.
- Once converted to ASCII text and indexed on disk, the research papers can be searched and analysed using LitBase.
- The text or images of the research papers can be displayed on screen and printed or copied to other applications.

If, for instance, a reference to a particular topic is required, LitBase can search the stored research literature for all entries referring to that particular topic, and present the user with a detailed list of references.

3.1.3.LitBase Hardware and Software

LitBase was designed to operate on hardware and operating systems which were readily available to the local research community. It should operate on any system with specifications similar to those provided below.

LitBase was designed to run under the Windows 3.x operating system and Workgroups for Windows local area network on IBM compatible PC's. LitBase used a Hewlett Packard flatbed scanner with automatic document feeder to digitise the research articles. Image storage was made by a Phillips compact disc recorder to recordable compact discs (CD-R). CD-R image access was made via a Pioneer compact disk jukebox. PCs used to develop and implement LitBase included Intel 80486 and Pentium processor equipped machines. Document image display was made on NEC high resolution monitors. The hardware and software apparatus reference codes for these items are: H2, H4, H5, H6, H7, H9, H10, H11, H16, S1, S2 and S8, found in Appendix D.

LitBase makes use of several software packages to implement some of its features. OmniPage Professional provides scanning and OCR functions. Write, a Windows 3.x accessory, provides the document text display. ImagePac enables the display of document images on screen. ISYS provides some of the full text search and analysis features. The software apparatus reference codes for these items are: S3, S12, S13 and S15, found in Appendix D. The modular design of LitBase allows any of these packages to be replaced with a similar package with only slight modification to the LitBase system.

3.2.LITBASE DEVELOPMENT

This section examines the development of the LitBase system from its simple database prototype to its final integrated package. The development of the LitBase user interface, reference database, image storage and retrieval, text recognition and literature analysis modules are described and examples given. The integration of the system into a user friendly and homogeneous package is also described.

3.2.1.User Interface

The LitBase system and user interface evolved from a simple, textual database to a comprehensive system for processing and analysis of research literature. There were, however, several developmental stages between the first and current versions, each of which introduced new features and greater complexity. One aspect of LitBase common to all the versions is the graphical and intuitive user interface, as shown by the LitBase main menu in Figure 3.1.

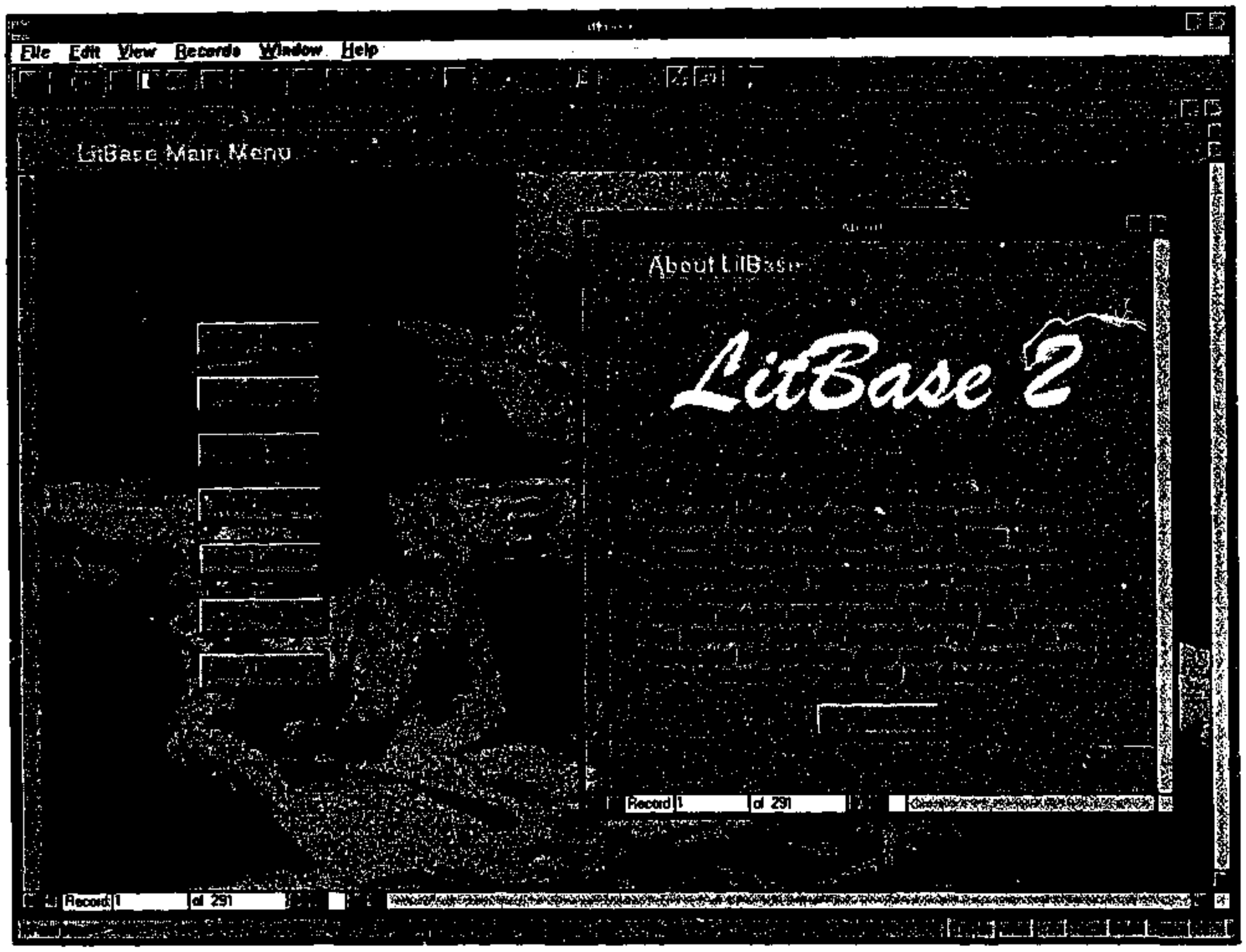
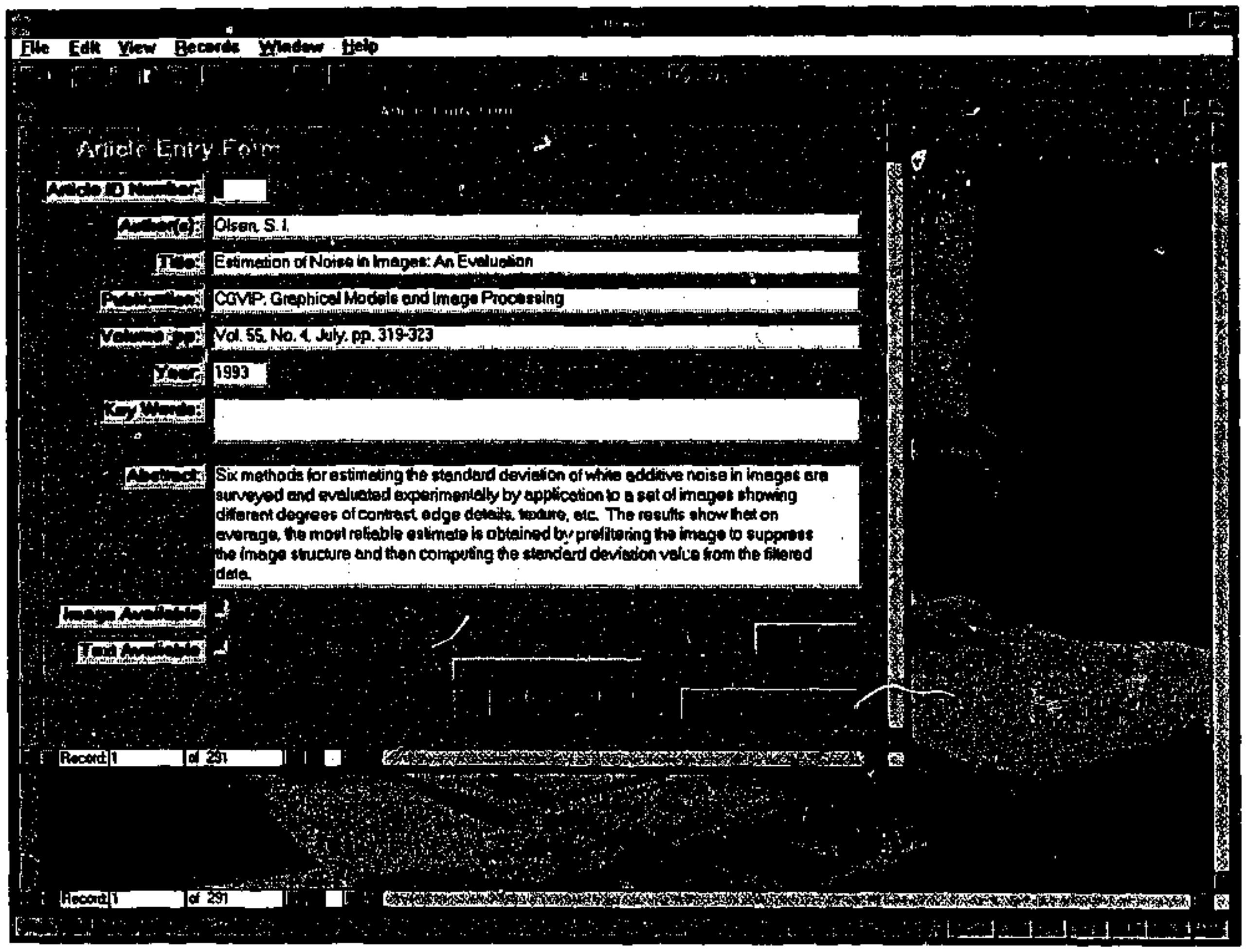


Figure 3.1 LitBase main menu and about screen.
This screen view of LitBase illustrates the intuitive graphical interface of the LitBase program. The menu system and selection buttons to access program features are shown, as is the About window describing the program.

3.2.2.Reference Database

The early versions of LitBase simply provided a reference database for storage and retrieval of research paper particulars such as author, title, publication etc., and an index number corresponding to the location of a photocopy of the research paper in a filing system. This proved adequate for a small number of research papers, but as the collection grew, so did the need for a more detailed description of the papers. An abstract field was added to the database so that more detailed searches of the collected literature could be conducted. An example of the article entry screen is shown in Figure 3.2. The research papers particulars are shown in the appropriate fields.



Article ID Number:

Author(s): Olsen, S. I.

Title: Estimation of Noise in Images: An Evaluation

Publication: CGVIP: Graphical Models and Image Processing

Volume / pp: Vol. 55, No. 4, July. pp. 319-323

Year: 1993

Key Words:

Abstract: Six methods for estimating the standard deviation of white additive noise in images are surveyed and evaluated experimentally by application to a set of images showing different degrees of contrast, edge details, texture, etc. The results show that on average, the most reliable estimate is obtained by prefiltering the image to suppress the image structure and then computing the standard deviation value from the filtered data.

Image Available: ✓

Text Available: ✓

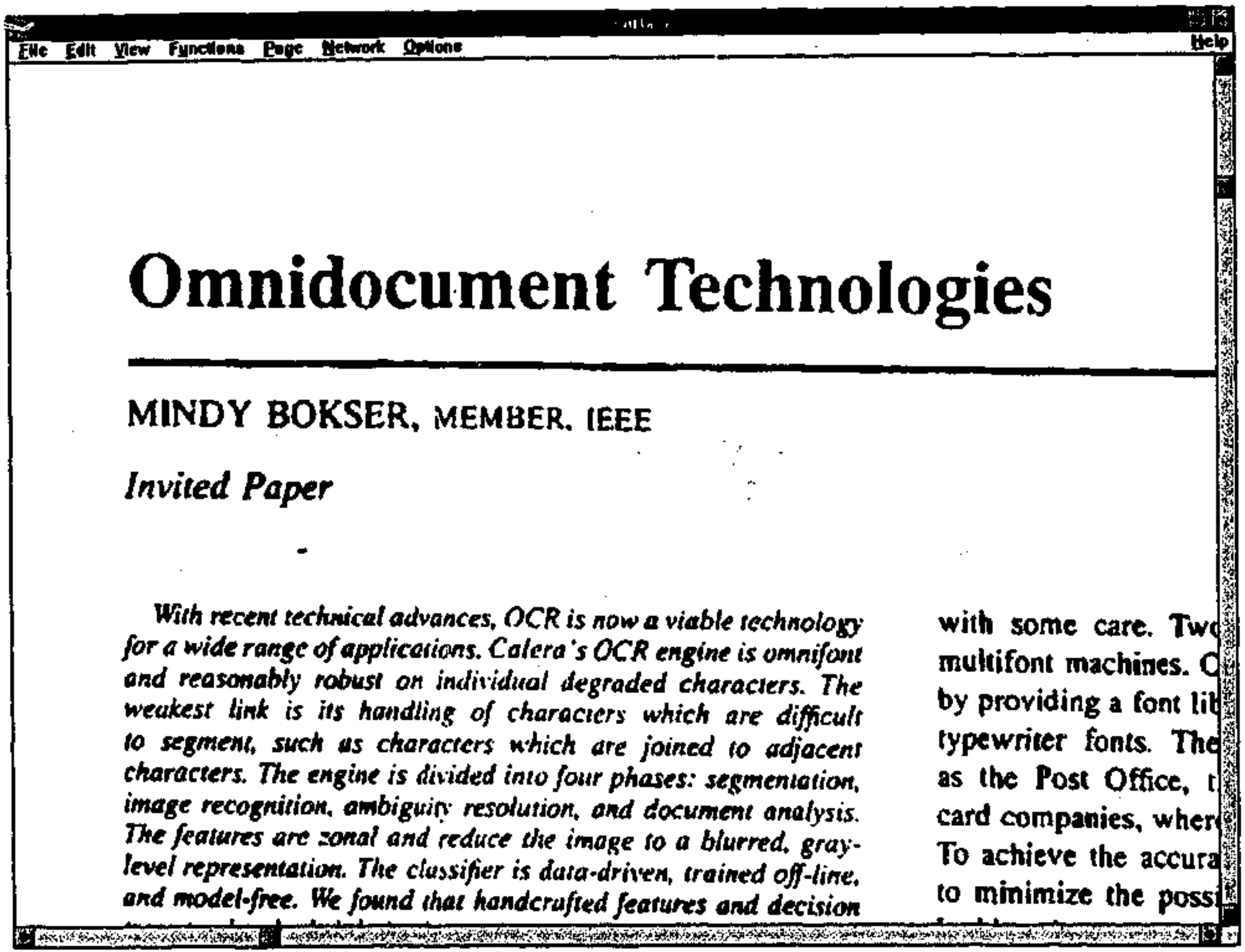
Record: 1 of 291

Figure 3.2 LitBase article entry screen

This screen view of LitBase illustrates the article entry form. The particulars of the research literature are entered into the fields of the article entry form.

3.2.3.Image Storage and Retrieval

The next versions of LitBase introduced the ability to store and display digitised images of the stored research papers. This involved linking LitBase with an external image viewing module. This permitted the user to search and display the research papers on screen, without the need to locate and refer to the paper copy of the research paper. An example of the image viewing module is shown in Figure 3.3. The front page of a research paper is shown zoomed in to display the detail of the text and graphics.



Omnidocument Technologies

MINDY BOKSER, MEMBER, IEEE

Invited Paper

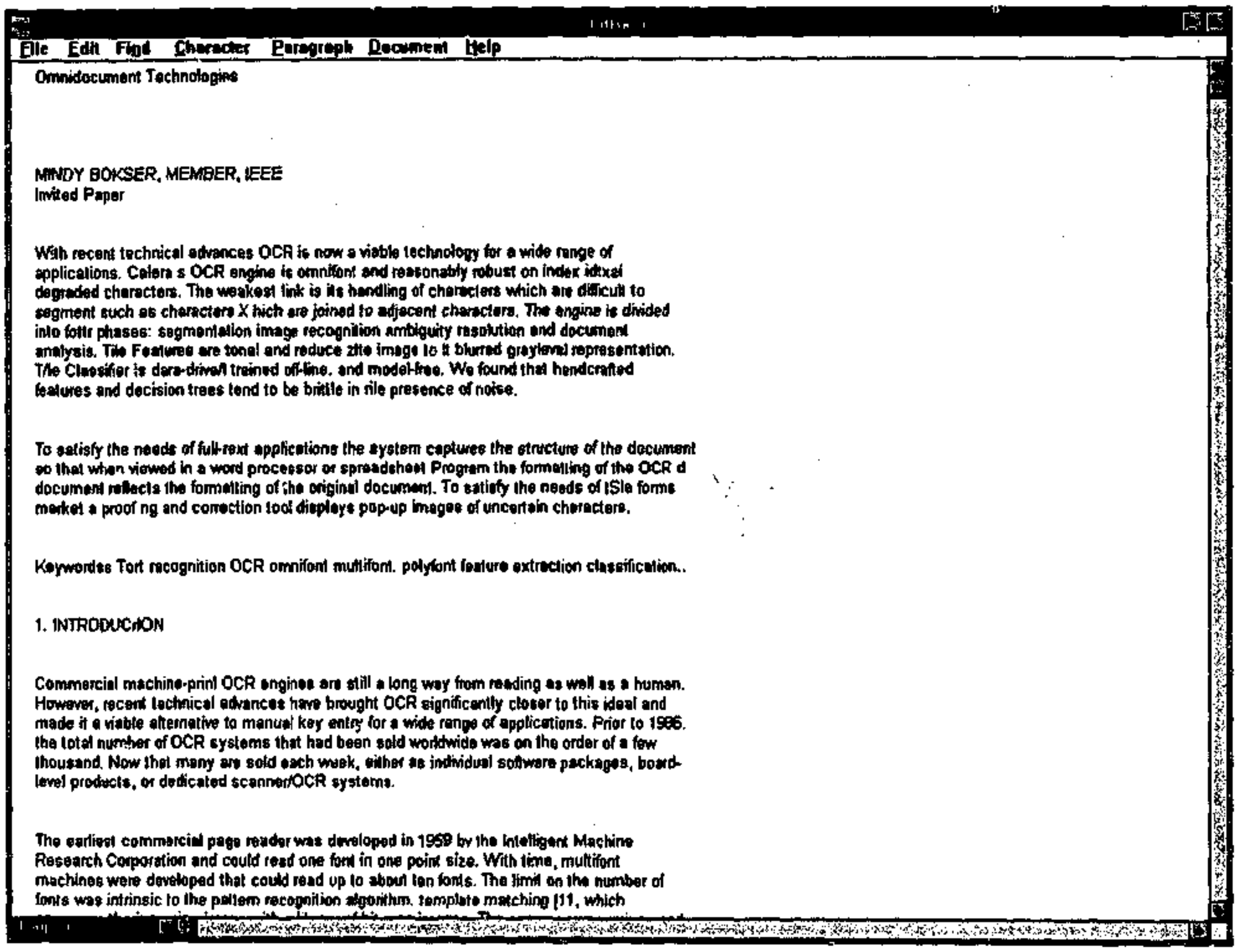
With recent technical advances, OCR is now a viable technology for a wide range of applications. Calera's OCR engine is omnifont and reasonably robust on individual degraded characters. The weakest link is its handling of characters which are difficult to segment, such as characters which are joined to adjacent characters. The engine is divided into four phases: segmentation, image recognition, ambiguity resolution, and document analysis. The features are zonal and reduce the image to a blurred, gray-level representation. The classifier is data-driven, trained off-line, and model-free. We found that handcrafted features and decision

with some care. Tw
multifont machines. C
by providing a font lil
typewriter fonts. The
as the Post Office, t
card companies, wher
To achieve the accura
to minimize the possi

Figure 3.3 LitBase image viewing module
This screen view of LitBase illustrates the image viewing module. The research article can be viewed as is or enlarged to display fine detail. Information can also be copied to other applications.

3.2.4.Text Recognition

Once the research papers were stored as images, the OmniPage OCR package was used to convert these images to text. A text display system was then added to LitBase so that the whole text body of the research paper could be displayed on screen. This involved linking LitBase with an external text viewing module. Users could now take the relevant portion of a research paper which had been located and then place the text straight into a word processor or other application for their own use. An example of the text viewing module is shown in Figurc 3.4. Note the text recognition errors which have occurred in this example as a result of using OCR on a poor quality photocopy of the research paper.



Omnidocument Technologies

MINDY BOKSER, MEMBER, IEEE
Invited Paper

With recent technical advances OCR is now a viable technology for a wide range of applications. Calera s OCR engine is omnifont and reasonably robust on index idtxal degraded characters. The weakest link is its handling of characters which are difficult to segment such as characters X hich are joined to adjacent characters. The engine is divided into fottr phases: segmentation image recognition ambiguity resolution and document analysis. Tile Features are tonal and reduce zite image to it blurred graylevel representation. T/le Classifier is dara-drivell trained off-line, and model-free. We found that handcrafted features and decision trees tend to be brittle in rile presence of noise.

To satisfy the needs of full-rext applications the system captures the structure of the document so that when viewed in a word processor or spreadsheet Program the formatting of the OCR d document reflects the formatting of the original document. To satisfy the needs of tSle forms market a proof ng and correction tool displays pop-up images of uncertain characters.

Keywordes Tort recognition OCR omnifont multifont. polyfont feature extraction classification..

1. INTRODUCrlON

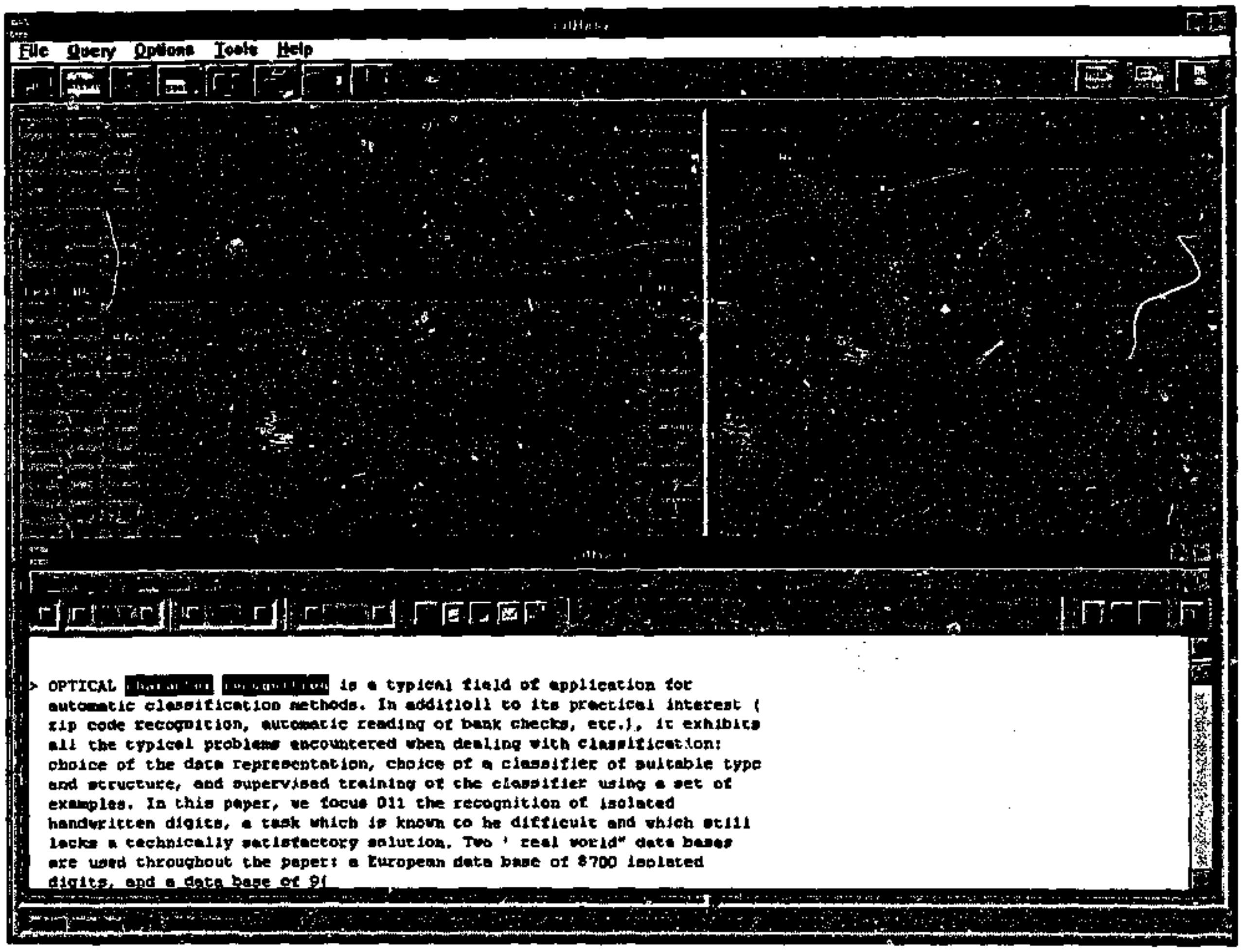
Commercial machine-print OCR engines are still a long way from reading as well as a human. However, recent technical advances have brought OCR significantly closer to this ideal and made it a viable alternative to manual key entry for a wide range of applications. Prior to 1986. the total number of OCR systems that had been sold worldwide was on the order of a few thousand. Now that many are sold each week, either as individual software packages, board-level products, or dedicated scanner/OCR systems.

The earliest commercial page reader was developed in 1959 by the Intelligent Machine Research Corporation and could read one font in one point size. With time, multifont machines were developed that could read up to about ten fonts. The limit on the number of fonts was intrinsic to the pattern recognition algorithm, template matching [11, which

Figure 3.4 LitBase text viewing module
This screen view of LitBase illustrates the text viewing module. The research article text can be viewed and edited. Information can also be copied to other applications.

3.2.5.Analysis System

This section describes the text analysis system used by LitBase. Although this version of LitBase provided the ability to search the paper's particulars and abstract, the user could not automatically search the body of the research paper. A link to an external indexing and analysis module by LitBase was introduced to complete the search engine. Now the user could search the entire research literature collection for references to any group of terms chosen. An example of the indexing and analysis module is shown in Figure 3.5. In this example several references to the search term have been found.



> OPTICAL character recognition is a typical field of application for automatic classification methods. In addifloll to its practical interest (zip code recognition, automatic reading of bank checks, etc.), it exhibits all the typical problems encountered when dealing with classification: choice of the data representation, choice of a classifier of suitable type and structure, and supervised training of the classifier using a set of examples. In this paper, we focus 011 the recognition of isolated handwritten digits, a task which is known to he difficult and which still lacks a technically satisfactory solution. Two ' real world" data bases are used throughout the paper: a European data base of 8700 isolated digits, and a data base of 9(

Figure 3.5 LitBase indexing and analysis module

This screen view of LitBase illustrates the indexing and analysis module. In this example a search has been conducted through a portion of the research literature for the term 'character recognition'. Articles containing the term are listed along with the number of times the term occurred in the article. One of the articles is shown in text form with the term highlighted.

3.2.6.System Integration

The final version of LitBase integrated the scanning and OCR module into the LitBase package, creating a seamless and integrated research literature storage and analysis system. Feedback from other academic and research staff using LitBase led to the inclusion of several other features, including user defined fields and improved sorting and printing facilities.

The structure and process flow in the LitBase system is illustrated in Figure 3.6. The photocopied research articles are taken and their reference details are typed into the LitBase article entry form. The reference details for the article are stored in the reference database before the article is passed onto the scanner to be digitised. The images of the articles are stored on the hard drive while the physical articles are stored in a filing system. The images are converted to text by the OCR module and the text is stored on the hard disk to await further analysis.

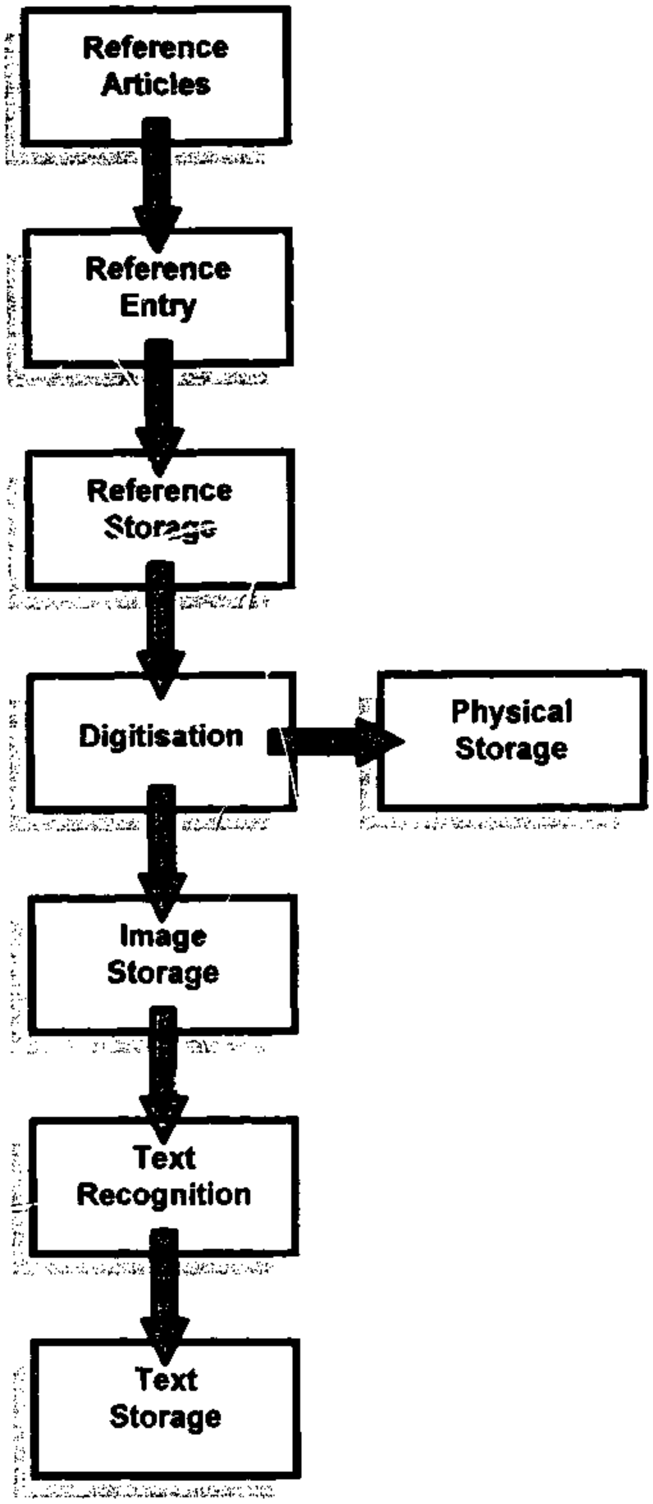


Figure 3.6 LitBase process flow diagram
This diagram shows the flow of the research articles through the LitBase system. The research articles are passed through the various processing stages in order before being stored as text which can later be analysed.

The architecture of the LitBase system is shown in Figure 3.7. The LitBase control software lies at the centre of the system. The operator can communicate with the LitBase control software via the graphical user interface. The LitBase control software is linked to the image viewing, text viewing, text analysis, and scanning and

OCR modules. The operator and the modules can access the reference, article, image and text databases via the LitBase control software.

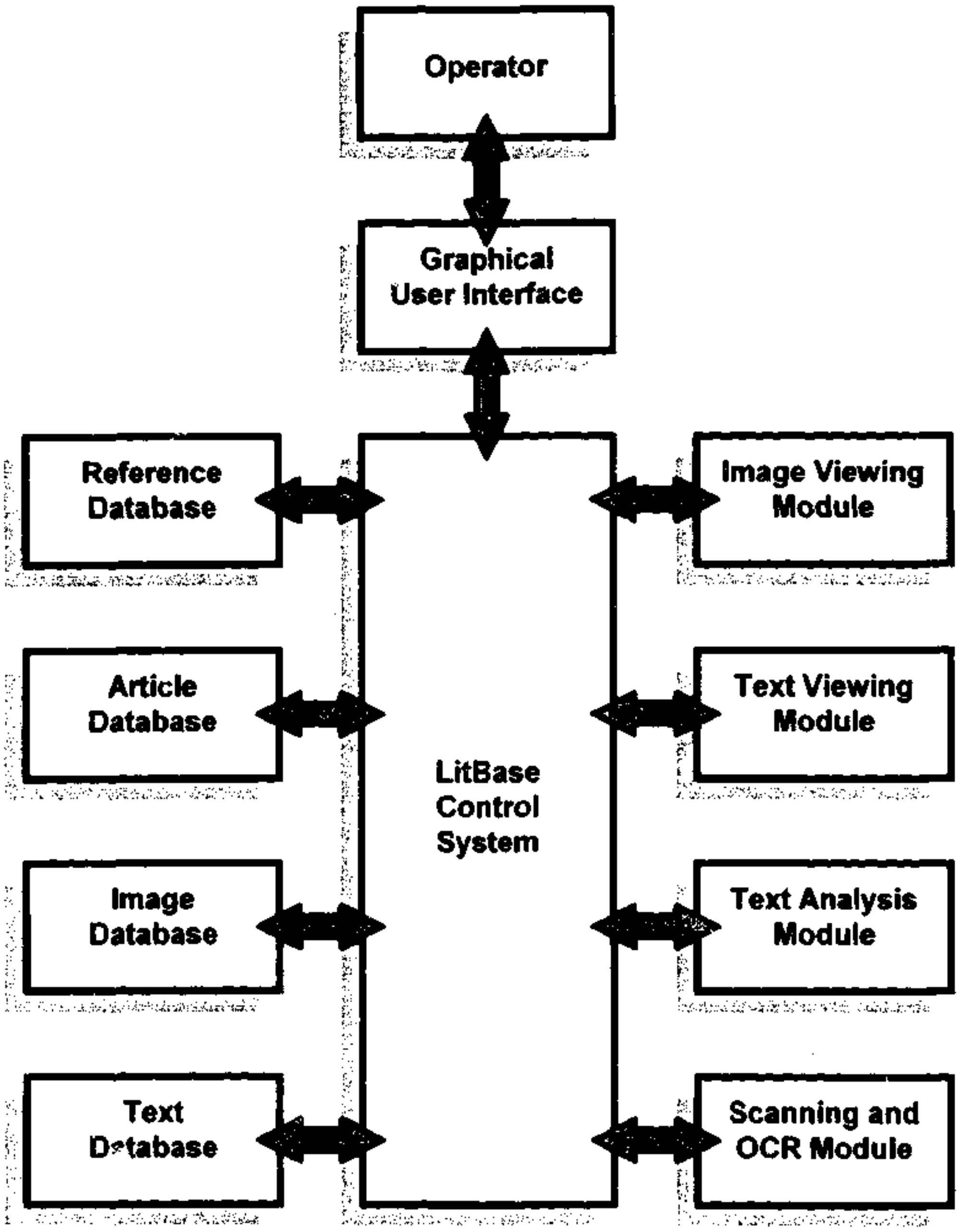


Figure 3.7 LitBase system architecture
This diagram shows the architecture of the LitBase system. The operator communicates with the LitBase control system via the graphical user interface. The external modules on the right can access the databases on the left via the LitBase control system.

It should be noted that the scanning and OCR, text viewing, image viewing, and search and analysis modules are separate programs developed by other companies. These modules were configured for, and integrated with, the LitBase system by the author. The Access source code for LitBase system is included in Appendix C.

3.3. LITERATURE ANALYSIS

This section examines the analysis of the research literature using the LitBase system. It demonstrates the precision of the term search module with a example search and describes the results which highlight a need for an OCR system model for the high volume document environment.

LitBase enabled a high level of precision to be achieved during the literature analysis stage which would have been difficult if not impossible to achieve by more manual means of literature analysis. Using LitBase it was possible to locate each and every reference to specific topics in the collected literature. This term searcn precision highlighted several areas in the literature which required further research including OCR system modeling.

3.3.1. Term Search Precision

An example of the term search precision is given in Figure 3.5. It shows the search results for the term 'character recognition' in a selection of research papers. Every research paper which contains the search words is listed, as well as the number of occurrences of the search words in each paper. In this instance 23 research papers from a selection of 65 contain one or more of the 478 occurrences of the search terms.

The user can navigate through the research papers with occurrences of the search words. The research paper text is displayed, as shown in the lower window of Figure 3.5, with the search words highlighted to assist location by the user. LitBase's ability to search the literature collection is not perfect however. As can be seen from the research paper text from Figure 3.4 and Figure 3.5, there are a number of errors in the text. The recognition errors in the text restrict to some extent the ability of LitBase to search the text containing the errors.

A more extensive example of LitBase's term search precision is given in Table 3.1. The table shows the search results for several LitBase term searches from the entire 301 research paper collection. The terms are listed along with the number of times each term was located by LitBase within the library of research papers. The number of times all terms were located is given in column five (labeled "All Terms").

Table 3.1 shows that in most cases where more than one term is searched for, the number of times all the terms (column five) occurred together is less that the sum of the occurrences of the individual terms. The exception to this is illustrated by the term search for "quantisation (OR) quantization" where the number of occurrences of all terms is the sum of occurrences of the individual terms. Also listed in the table is the number of articles which contain all the search terms (column 6).

The last column in Table 3.1 gives the accuracy of the term search. The accuracy listed is a measurement of the proportion of all terms found from the ones actually occurring in the papers. The term search accuracy ranges from 90.1% to 98.2% with a mean of 94.4%. Examination of the All Terms and Accuracy columns shows that the extremities in accuracy variation occur mainly on term searches which locate fewer terms. The majority of the errors which occured in the term searches can be attributed to errors in the text generated by the OCR software. Further examination of these LitaBase term search accuracy results is conducted in Section 3.4.1.

Table 3.1 Term search results and precision

This table reports several term search results from LitBase on a selection of 301 research papers. The terms are listed along with the number of times each term was located in the library. The number of times all terms occurred is listed, along with the number of articles located containing all the terms. The accuracy is a measurement of the proportion of all terms found from the ones actually occurring in the papers.

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
optical character recognition	8615	11542	11589	1175	142	95.6%
document processing	8529	3163		267	93	94.8%
text recognition	7254	11589		440	61	98.2%
OCR	3752			3752	177	94.5%
normalisation	206			206	72	91.3%
segmentation	3494			3494	239	93.2%
quantisation (OR) quantization	96	52		148	90	95.4%
image processing	10537	3162		1422	236	94.2%
thresholding	185			185	87	90.1%
classification	1881			1881	214	94.6%
error correction	146			246	66	92.5%
					[illegible]	[illegible]

The ability of LitBase to quickly search and display the text and images of research papers enables the user to interactively explore the literature collection in a similar fashion to the many multimedia encyclopedias currently available on CD-ROM.

Interactive analysis of the literature collection and term search analysis like those shown in Table 3.1 highlighted the need for a more detailed understanding of the factors which influence OCR performance. The factors that were highlighted included typeface, font type, text size, image resolution, printing device and image skew. Also highlighted by the literature analysis was the need for a greater understanding of the relationship between the OCR performance classes and the factors or characteristic variables which affect them. The relationship between the OCR performance classes and the characteristic variables are explored in greater detail in Chapter 4 and Chapter 5.

3.3.2.OCR System Model

The high volume document environment which is the focus of the thesis presents a relatively specific field within the digital image processing area. The LitBase literature analysis on the subject revealed significant scope for development including the concept of a global system model. A system model for digital image processing systems in high volume document environments would provide several benefits.

The model could describe in detail the high volume document environment and differentiate that environment from others in the digital image document processing field. The model could incorporate the results obtained from the experimental work into OCR performance classes and characteristic variables. By developing the model to reflect real world constraints, the model could be used to optimise digital image processing systems in real world high volume document environments. The development of the model is reported in Chapter 7 and leads into Chapter 8 which evaluates the performance of the model under various conditions.

3.4.LITBASE ANALYSIS

This section analyses the LitBase system in terms of its performance as an OCR system and as a literature storage, analysis and retrieval system. The accuracy, speed and sensitivity performances are calculated from evaluation tests conducted using a representative sample of documents and data logged during normal LitBase operation.

3.4.1.Accuracy Performance

The OCR accuracy level averaged 94.4 per cent over the 301 research papers converted to text by the LitBase system and is calculated using Equation 2.1. This corresponds to the average accuracy level of LitBase term searches of 94.4 per cent which is reported in Section 3.3.1 and can be interpreted as being the main factor influencing LitBase term search accuracy. The 3.6 percent OCR error rate led to some occurrences of search words not being found in some papers. The fraction of research papers not located by the term searches averaged 1.2 per cent for the term searches presented in Table 3.1. This error level was alleviated to some extent by the search and analysis module's ability to detect similar words.

An investigation into the cause of the OCR error rate attributed much of the errors to poor duplication of the original research papers. Factors such as perspective distortion [77] and photocopier quality were the main causes of the poor duplication. Small size text and low scanning resolution also contributed to the recorded OCR error rate for LitBase. The effects of text size, scanning resolution and image quality upon the OCR accuracy are covered to some extend by Chapter 2. A more thorough understanding was desired, however, in particular the relationship between text size, scanning resolution and OCR accuracy. Knowing the relationship between text size, scanning resolution and OCR accuracy would enable the OCR accuracy performance to be optimised. After some initial investigations, a series of preliminary experimental tests were conducted. This lead to the results reported in Chapter 4 and analysis of those results in Chapter 5.

3.4.2.Speed Performance

The speed performance of the LitBase system can be calculated by examining the speed performance of the individual LitBase modules. These include the scanning module, OCR module, and search module. These module's speed performances can be added to compute an article entry speed.

The average scanning speed of the LitBase system using a HP Scanjet IIC (200DPI, A4 sized, binary images) was 57.2 seconds per page. With an average of 8.81 pages per document, this equates to 8.40 minutes per document average scanning speed. The Scanjet IIC scanner is a low volume color scanner which is capable of, but not ideally suited to, high volume black and white scanning work. Use of a faster and more suitable scanner would have substantially improved scanning speed performance.

The average OCR speed of the LitBase system on a 486DX50 PC was 23.7 characters per second system and is calculated using Equation 2.2. With an average of 39.8 thousand characters per document, this equates to 30.0 minutes per document average OCR speed. Use of a faster PC would have substantially improved OCR speed performance, since OCR speed is proportional to the processing power of the PC.

The average search speed of the LitBase system on a Pentium90 PC was 5,700 search term occurrences per second. The search speed varied considerably throughout the LitBase evaluation tests, but was always so fast as to seem almost instantaneous to the user.

Another speed performance measurement that may be useful is the time taken to enter the average article into the LitBase system. The article entry time is the sum of the reference detail entry time, scanning time, OCR time plus a system overhead time. The average article entry time computed by summing the individual module times was 43.9 minutes per article. For the 301 articles entered into the LitBase system, the total computed article entry time is 220 hours. The actual time taken to enter the 301 articles was much less then the computed time.

Several factors can cause the article entry speed to be substantially faster. Use of an automatic document feeder (ADF) on the scanner means that while several articles are being fed through the scanner, the article reference details can be typed into LitBase operating on a second PC. This effectively makes reference detail entry and scanning concurrent. Since these processes are concurrent, the use of a slower color scanner to handle the scanning work has less of an impact on the overall article entry speed.

The OCR of the scanned articles can be deferred and batched for OCR at a later time. The OCR can therefore be conducted at a time when the PC is normally inactive, e.g. overnight. This effectively reduces the impact of OCR time on the article entry speed. Since the OCR time is effectively reduced, the use of a slower PC to carry out the OCR work has less impact on the overall article entry speed.

The two remaining components of the article entry time are reference detail entry time and system overhead time. These components are affected by operator data entry speed and paper handling efficiency and can be seen as the main components affecting article entry time. The actual time logged by the article entry operators for the 301 research articles was 76.5 hours. This gives an average article entry time of 15.3 minutes per article. Comparing the average article entry time with the computed article entry time of 43.9 minutes shows a saving of 28.6 minutes per article. This can be directly attributed to the concurrent scanning and differed OCR processing and represents an improvement of 187% to the speed performance of LitBase's article entry.

3.4.3.Sensitivity Performance

The evaluation tests on LitBase showed some of the sensitivity limits of the system. The LitBase scanning module proved to be fairly robust and was able to digitise and binarise all the A4 sized photocopied documents presented to it. All the scanned documents text was human readable when displayed on screen or printed. The LitBase OCR module, however, was particularly sensitive to poor quality documents. Poor quality documents with breaks in characters and touching characters were especially difficult to OCR correctly. The LitBase OCR module was also sensitive to text sizes smaller than or equal to six points. The small text sizes occurred as a result of photocopy size reduction. The small text sizes lowered the OCR speed and the OCR accuracy performance, in some cases preventing recognition altogether. The small text size accuracy problem was overcome to some extent by photocopy enlarging the documents and scanning them again.

The accuracy, speed and sensitivity tests conducted for the analysis of the LitBase system provided valuable background knowledge for beginning the experimental work which is reported in Chapter 4.

3.5.LitBase Summary

The performance of LitBase can be summarised into three areas; accuracy, speed and sensitivity. The mean term search accuracy of LitBase was 94.4 per cent for the 301 research paper collection, while the mean OCR accuracy of the text was also 94.4 per cent. The mean speed of article entry was 15.3 minutes per article, while the speed of term searches was less than a second. LitBase proved especially sensitive to low quality photocopied articles and text sizes smaller than 6 points.

The findings reported in this chapter are comparable with those in the relevant literature. The work reported by Croft et al. [78] on the evaluation of information retrieval accuracy with simulated OCR output draws similar conclusions to those presented in this chapter; that low quality devices used with databases can result in significant degradation in information retrieval accuracy.

As a research literature specific database and analysis system, LitBase represents an original and useful contribution to the area of digital image document processing.

CHAPTER 4

PRELIMINARY EXPERIMENTAL WORK AND RESULTS

CHAPTER CONTENTS PAGE

4. PRELIMINARY EXPERIMENTAL WORK AND RESULTS

This chapter covers the preliminary experimental work which was conducted and presents the results obtained. An introduction to the experimental work and the approach taken is given. The development of the DIAC experimental laboratory is explained. The hardware, software and data sets used to conduct the experimental work are examined. The OCR performance classes considered for the experimental work are described. The individual experiments are classified and reported according to the characteristic variables being examined. A summary of the results concludes the chapter by examining areas of further work and comparing the results to those reported by others.

4.1. INTRODUCTION

The purpose of the preliminary experimental work was to investigate the avenues of research identified by the analysis of the literature conducted in Chapter 3. The literature analysis identified the need for a greater understanding of the variables which affect OCR performance. The work by Baird [8], [9] and others [4], [6], [51] identified several characteristic variables which affect the OCR performance classes described in Section 2.1.4. These characteristic variables include typeface, font type, text size, image resolution, printing device and image skew.

A series of experiments were devised to quantify the effect of these six characteristic variables on the OCR performance classes. In order to conduct the experimental work, a series of data sets and a laboratory had to be established.

4.2. LABORATORY DEVELOPMENT

At the beginning of the research program there were few suitable pieces of apparatus for conducting experimental work into OCR and digital image document processing in the DIAC laboratory. The laboratory at DIAC had to be properly developed before any experimental work could proceed.

A series of personal computers were installed in the DIAC laboratory to handle general processing tasks (apparatus reference codes H8 and H11). In addition to these personal computers were two special purpose computers. The first of these special purpose computers (H9) was equipped to handle digitising, display and processing of documents. It was attached to a scanner (H1) for digitising documents and was equipped with a large sized high resolution monitor (H6) for viewing images. The second special purpose computer (H10) was also equipped with a high resolution monitor (H6). It was attached to a compact disk recorder (H4) and a compact disk juke box (H5) for image archival and retrieval.

All the computers were then networked to facilitate data transfer between them. Additional scanners, processor upgrades and other equipment were added to the laboratory throughout the research program. A more complete description of the major pieces of apparatus used and their apparatus reference codes are given in Section 4.3 and in Appendix D.

4.3. EQUIPMENT AND METHODOLOGIES

This section describes the software and hardware equipment and methodologies used to carry out the preliminary experimental work. A description of the hardware and software systems is given, followed by a description of the data sets used for the experimental work. Further details of the equipment used can be found in the apparatus schedules of Appendix D.

4.3.1. Hardware Systems

The major hardware systems used for conducting the preliminary experimental work consisted of a personal computer, flatbed scanner and several printers. The personal computer used an Intel 80486 DX processor. The flatbed scanner was a Hewlett Packard grey scale scanner. The printers included a Toshiba laser printer, a Hewlett Packard inkjet printer and an Epson dot matrix printer.

Further details of the hardware systems used are listed in Table D.1 in Appendix D. The apparatus reference codes for the major hardware items used in the preliminary experimental work are: H1, H8, H12, H13 and H14.

4.3.2.Software Systems

The major software systems used for conducting the preliminary experimental work consisted of the Microsoft DOS and Microsoft Windows operating systems, the OmniPage OCR software, and the Aldus Photostyler image processing and scanning software.

Further details of the hardware systems used are listed in Table D.2 in Appendix D. The apparatus reference codes for the major software items used in the preliminary experimental work are: S1, S2, S3 and S14.

4.3.3.Data Sets

An examination of the available document image databases used by other researchers for experimental work showed that none were really suited for the proposed experimental work. It was therefore necessary to develop and document the data sets used for the experimental work reported in this chapter. The standard image database "English Document Database CD-ROM" reported by Baird [9] became available only after the preliminary experimental work was completed. Data sets developed later by Ho and Baird [11] are consistent with those reported in this chapter.

Several different data sets were established for conducting the preliminary experimental work. Two character sets were chosen, and from these two, further data sets were established.

The two character sets which were established for conducting the preliminary experimental work are a full character set and a limited character set. The full character set is composed of most of the machine typed characters, punctuation and common symbols. The full character set is shown in Figure 4.1 in a twelve point size Courier typeface. The limited character set is composed of numerals, upper case characters and lower case characters only. The limited character set is shown in Figure 4.2 in a twelve point size Courier typeface

```
 !"#$%&'()*+,-./0123456789
:;<=>?@ABCDEFGHIJKLMNOPQRS
TUVWXYZ[\]^_`abcdefghijklm
    nopqrstuvwxyz{|}~
```

Figure 4.1 Full character set
The full character set is composed of most of the machine typed characters, punctuation and common symbols. It includes ASCII characters 32 to 127. In this example the full character set is shown in a twelve point size Courier typeface.

```
        1234567890
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
```

Figure 4.2 Limited character set
The limited character set is composed of numerals, upper case characters and lower case characters. This includes ASCII characters 48 to 57, 65 to 90 and 97 to 122. In this example the limited character set is shown in a twelve point size Courier typeface.

The full character data set used in the preliminary experimental work consists of twenty rows of twenty six randomly ordered characters from the full character set for a total of 520 characters per page. The limited character data set consists of ten rows of twenty six randomly ordered upper case characters, followed by ten rows of twenty six randomly ordered lower case characters, followed by a further ten rows of twenty six randomly ordered numerals, for a total of 780 characters per page. This arrangement of characters is chosen because it can comfortably be printed in text sizes up to 24 points on a single A4 page.

An example of the full character data set is provided in Figure 4.3. This randomised synthetic data set, composed of ASCII characters commonly used for such tests [11], [76] is chosen in order to eliminate contextual influence and to permit control of the quality of the test data.

```
}v%d\>mRDuIgU'GJihGd0R8i$M
q2@n k8nQV&j]--841*SLE$fDT
zI.(#*Sfa_>DL8007J7,~*r\n)
^TSS~dTegyL4&TUAlfJ${i.k;(
sxBK-|7=+7U(hoIwd$h?w((TPT
mIM|cFQ.{Ky}S 7d:Ju=HRue6A
TOx*l8tPYAu#}&1QNq{loljEs\
Q4?~9uE`Wc_z&)W7|IG#RR(W<C
LHk}Iu}A#Z(e>LbLf/BfE wOF1
H-Ookd:3 T+[^f/wl_[c8Ys6XT
wMk6ArPxy_U^zm8xh^f(T-CB"{
ZJV;#z{ua-2a]H?u%,Pt'nDf\k
6 qIu^Zd'[Q{1t2/O`[pho8Xpp
zcE[N=1wl^,Py32gj-OzT1J{42
7XmSR Jo4nt,\\Nj^$!*n!['};
ZNL<;;IXJ=RaR"%Vr}!5GvR==u
_t^Ie)`$MwIiYZ;gl"b|UC%.ab
8xsxt-%6LuWBW!tab3;z"#4N7q
w$FGQTpG\C<+yR4K3JPFUil;cN
Sa0yT=rgh0FJ86>Z&M$\5"[=^U
```

Figure 4.3 Synthetic data set example for the full character set
This data set consists of 520 randomly ordered characters from the full character set in 20 rows of 26 characters each. In this example the full character set is shown in a twelve point size Courier typeface.

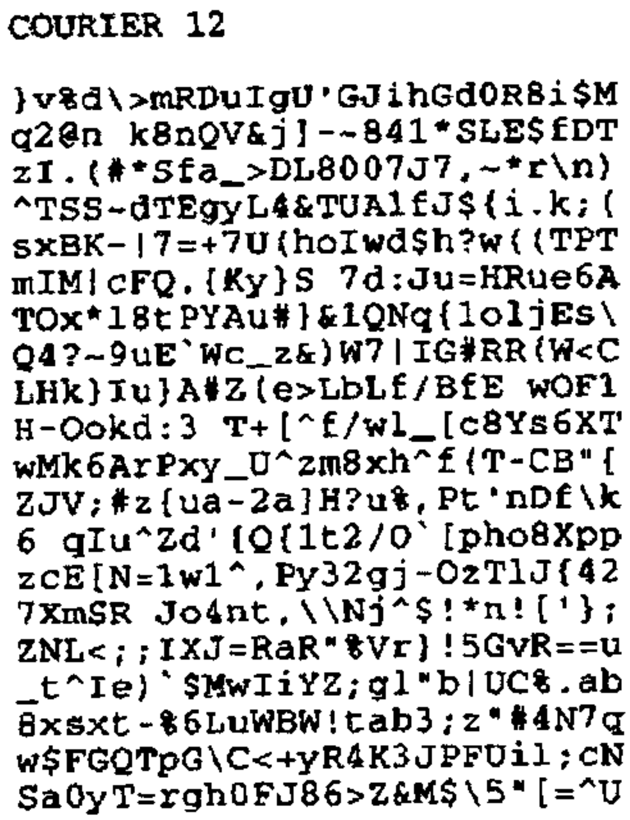
The separate groupings for numerals, upper case characters and lower case characters for the limited character data set was done to allow for separation of experimental results for these sub-divisions of the limited character set. An example of the limited character data set is provided in Figure 4.4.

```
FEFPWBQPCCHOLKBCRBARUUHOBH
CYVRVKBDWBBJFDNHICIWXSDRRD
QIEICKNIQUYLWOSEWDGWNPPHJW
LNVMRJISUYHBFOANUJGOANTDMW
QBXGYMCGGIHJQCBXYHMLKNWJXM
IVURILWVUNQPSSWNRSKOJUWFWX
YNCROUVIQVZMPDYBGNHMYJZADG
IRDGZJIRSDCYKSTKEKZIKKWZFZ
MXBPFXJCKQWFVRVJLHCTWWZTHI
CDIQPKDMCHVOACUQUZQNRYKCGL
ujqhzsnduedywmuouifhsokwab
dpkummydlzdhzgjxlpnbwsudsp
okyfexlvuirqcdfpzfuszlgapp
yjzoxmwsvnhuzwgdfybmedpsyz
lxzafglbvkjmybjqooglfttfgh
pvbqixjzneqcruuurnphrfeywt
gppglqltaznbbhtwjfrwmzwrox
hmuiilniyalyvxfignlwoowsue
vayditgormchdemgwdxfghueko
ldpkydfdtyqbnfnfumhluowfry
70719598417152838377207021
88452437152029977773214562
57367289680987862923395296
26367480876108626819426555
85881134946478205921974288
34723069578821849338633898
43465679905827675493699643
74426299702594397988555282
33790980283318926376469986
88752656214575519425134216
```

Figure 4.4 Synthetic data set example for the limited character set
This data set consists of 520 randomly ordered characters from the limited character set in 10 rows of 26 upper case characters, 10 rows of 26 lower case characters and 10 rows of 26 numerals. In this example the full character set is shown in a twelve point size Courier typeface.

The synthetic data sets were printed in Times Roman, Helvetica and Courier typefaces in sizes of 8 to 24 points. An example of these typefaces is shown in Figure 1.2 of Chapter 1. These data sets were scanned at 100, 200, 300 and 400 DPI, which are resolutions typically used for document imaging [60]. These scanned synthetic data sets form the basis for most of the preliminary experimental work. An example of these scanned synthetic data sets is given in

Figure 4.5 which shows a 100 DPI scan of the 12 pt size Courier Typeface from the synthetic data set using the full character set.



```
COURIER 12

}v%d\>mRDuIgU'GJihGd0R8i$M
q2@n k8nQV&j]--841*SLE$fDT
zI.(#*Sfa_>DL8007J7,~*r\n)
^TSS~dTEgyL4&TUA1fJ${i.k;(
sxBK-|7=+7U(hoIwd$h?w((TPT
mIM|cFQ.{Ky}S 7d:Ju=HRue6A
TOx*18tPYAu#}&1QNq{loljEs\
Q4?~9uE`Wc_z&)W7|IG#RR(W<C
LHk}Iu}A#Z(e>LbLf/BfE wOF1
H-Ookd:3 T+[^f/wl_[c8Ys6XT
wMk6ArPxy_U^zm8xh^f(T-CB"{
ZJV;#z{ua-2a]H?u%,Pt'nDf\k
6 qIu^Zd'{Q{1t2/O`[pho8Xpp
zcE[N=1w1^,Py32gj-OzTlJ{42
7XmSR Jo4nt,\\Nj^$!*n![']};
ZNL<;;IXJ=RaR"%Vr}!5GvR==u
_t^Ie)`$MwIiYZ;gl"b|UC%.ab
8xsxt-%6LuWBW!tab3;z"#4N7q
w$FGQTpG\C<+yR4K3JPFUil;cN
Sa0yT=rgh0FJ86>Z&M$\5"[=^U
```

Figure 4.5 Example of scan of synthetic data set
This figure shows a 100 DPI scan of the 12 pt size Courier Typeface from the synthetic data set using the full character set. The typeface and size are part of the scanned image as an identifier only and are not included in the recognition process.

4.4.OCR PERFORMANCE CLASSES

The three OCR system performance classes are defined by, and based upon, the criteria for evaluating OCR system performance which is described in Section 2.1.4. The three OCR performance classes which pertain to the high volume document environment are accuracy, speed and sensitivity.

4.4.1.OCR Accuracy

The OCR accuracy performance class is the performance class given most attention in the preliminary experimental work. The OCR accuracy performance class is based on the OCR accuracy criterion for evaluating OCR system performance which is described in

Section 2.1.4.1. For the preliminary experimental work, the OCR accuracy is defined as a percentage of correctly recognised characters. The equation used to calculate the OCR accuracy is given by Equation 2.1.

4.4.2.OCR Speed

The OCR speed performance class is based on the OCR speed criterion for evaluating OCR system performance which is described in Section 2.1.4.2. For the preliminary experimental work, the OCR speed is defined as the number of correctly recognised characters per second. The equation used to calculate the OCR speed is given by Equation 2.2.

4.4.3.OCR Sensitivity

The OCR sensitivity performance class is based on the OCR sensitivity criterion for evaluating OCR system performance which is described in Section 2.1.4.3. The treatment of the OCR sensitivity performance class in the preliminary experimental work is done in terms of the maximum variation from the mean in the sets of experimental results.

4.5.CHARACTERISTIC VARIABLES

There are six characteristic variables which were considered for the preliminary experiments. These characteristic variables are typeface, font type, text size, image resolution, printing device and image skew. A series of experiments was established to measure the effects of these characteristic variables on some or all of the OCR performance classes. The tabulated results of these experiment are listed in a series of tables in Appendix E.

4.5.1.Typeface

The typeface experiment was designed to measure the effects of different typefaces and character sets on the OCR accuracy performance class.

The typefaces chosen for the experiment included OCR-A, OCR-B, Courier, Helvetica and Times Roman. The first two typefaces were chosen because they are typefaces standardised for optical character recognition. The last three typefaces were chosen because they are representative of the typefaces used for machine printed text documents and because they exhibit several classes of typeface: proportional, non-proportional, serif and sans-serif.

The data sets used in the experiment were the 12 point size, 300 DPI scanning resolution data sets which are described in Section 4.3.3. These data sets were selected because they consisted of a text size and scanning resolution that are representative of those used for machine printed text documents.

The data sets were processed five times to measure the variation in OCR accuracy for the different typefaces. The mean values of the processed data sets are shown in Figure 4.6. The maximum variation in OCR accuracy from the mean values shown in Figure 4.6 for each set of results was less than 0.8 per cent.

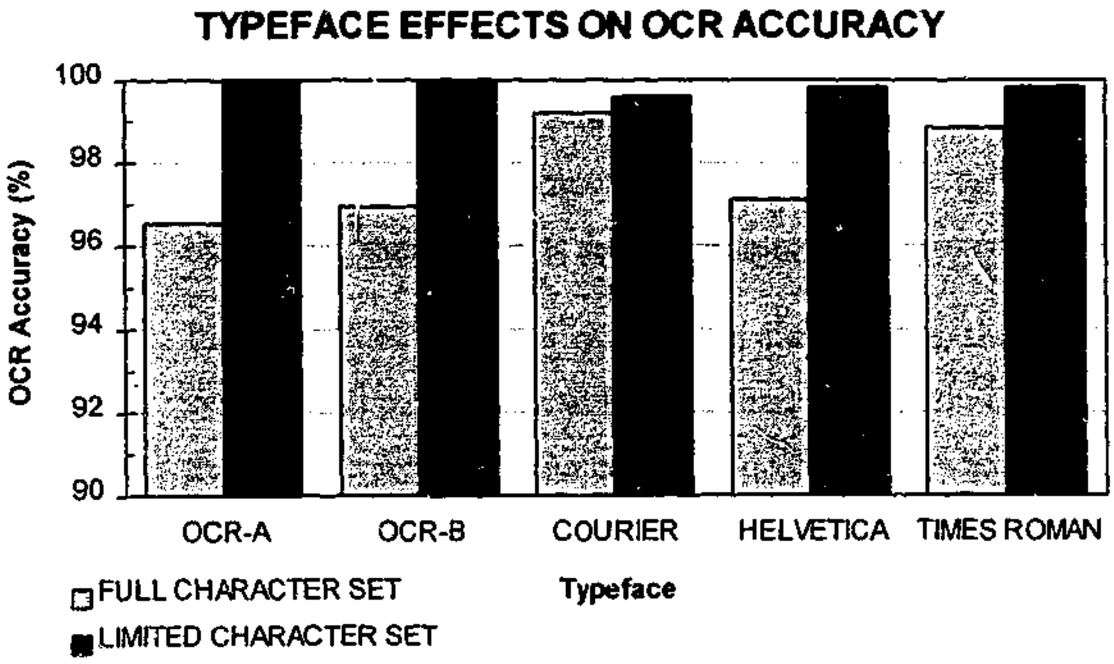


Figure 4.6 Typeface effects on OCR accuracy
This chart shows the effects of different typefaces on OCR accuracy for both the full and limited character sets.

Several general observations are made from Figure 4.6 concerning the effect of the typeface and character set characteristic variable on the OCR accuracy performance class:

- While the OCR-A and OCR-B typefaces have a lower OCR accuracy than the other three typefaces for the full character set, all the typefaces have an OCR accuracy of over 99.5 per cent for the limited character set.
- For all five typefaces used, OCR accuracy is improved by limiting the character set.
- The Courier typeface (a non-proportional serif typeface) shows the least improvement in OCR accuracy when changing from the full character set to the limited character set.

Analysis of the results of this experiment has led to some explanations for these general observations.

The full character set (refer to Figure 4.1) includes symbols and punctuation that are not part of the limited character set (refer to Figure 4.2). The analysis of the results showed that the lower OCR accuracy of the full character set is evidence of the difficulty in recognising these symbols and punctuation.

In the case of the Courier typeface, the analysis showed that its symbols and punctuation were recognised with higher OCR accuracy that other typefaces. Consequently, that typeface showed the least improvement in OCR accuracy when changing from the full character set to the limited character set.

The tabulated results of the typeface experiment are listed in Table E.1 and Table E.2 in Appendix E.

4.5.2.Font Type

The font type experiment was designed to measure the effects of different font types on the OCR accuracy performance class.

The basic font types chosen for the experiment included normal, bold, italic and underline. The four combinations of the basic font types were also included, for a total of eight font type variations. These font types were selected because they are representative of font types used in machine printed text documents.

The data sets used for the experiment were the 12 point size, 300 DPI scanning resolution, Courier typeface data sets which are described in Section 4.3.3. These data sets were selected because they are representative of the text size, scanning resolution and typeface used in machine printed text documents.

The data sets were processed five times to measure the variation in OCR accuracy for the different font types. The mean values of the processed data sets are shown in Figure 4.7. The maximum variation in OCR accuracy from the mean values shown in Figure 4.7 for each set of results was less than or equal to 0.6 per cent.

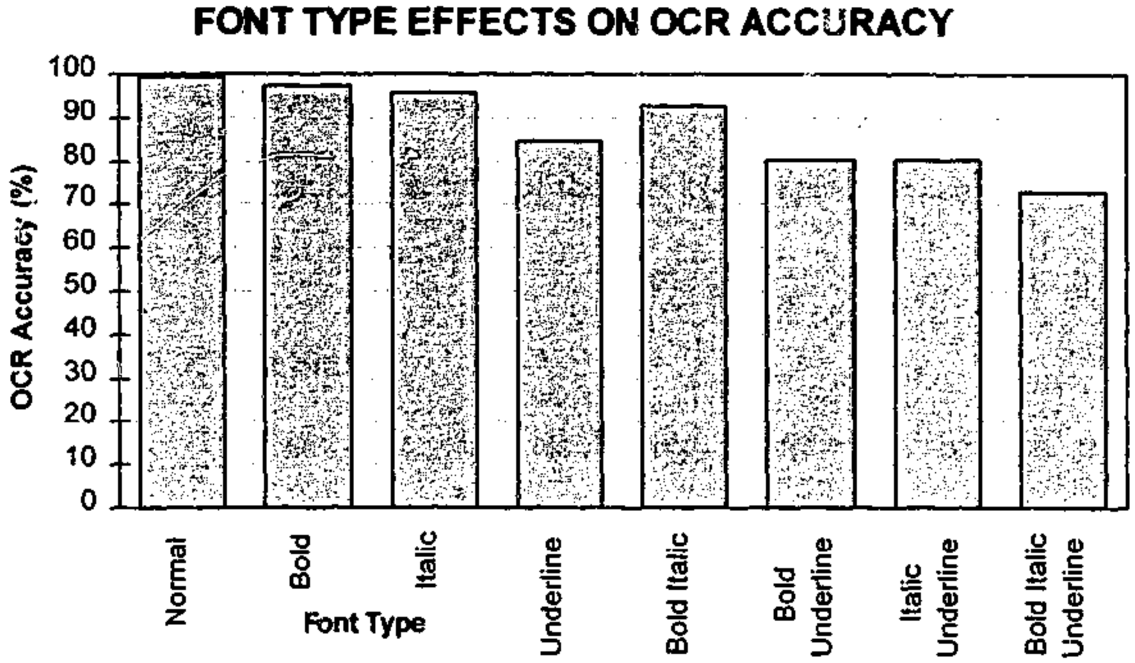


Figure 4.7 Font type effects on OCR accuracy
This chart shows the effects of different font types and combinations of font types on OCR accuracy for the twelve point size Courier typeface.

Two general observations are made from Figure 4.7 concerning the effect of the font type characteristic variable on the OCR accuracy performance class:

- The underline font type and font type combinations including underline have a lower OCR accuracy than the other font types.
- The more font types applied to the normal font type, the lower the OCR accuracy.

Analysis of the results of this experiment has led to some explanations for these general observations.

The underline font type places a line through the descending portions of characters, thereby reducing the OCR accuracy of those characters. Analysis of the results showed that this was the primary cause of the underline font type's lower OCR accuracy than either the bold or italic font types.

As more font types are applied to characters, the more these characters become distorted. Analysis of the results showed this distortion to be the cause of the lower OCR accuracy for characters with combinations of font types.

The tabulated results of the font type experiment are listed in Table E.3 in Appendix E.

4.5.3. Text Size

Two text size experiments were designed. The first text size experiment was designed to measure the effects of the text size characteristic variable on the OCR accuracy performance class. The second text size experiment was designed to measure the effects of the text size characteristic variable on the OCR speed performance class.

The text sizes chosen for both the experiments included 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 16, 18, 20, 22, and 24 point size text. These text sizes were selected because they are representative of text sizes used in machine printed text documents.

The data sets used in both of the experiments were the 300 DPI scanning resolution, Helvetica typeface data sets which are described in Section 4.3.3. These data sets were selected because they consisted of a scanning resolution and typeface which are representative of those used in machine printed text documents.

The data sets were processed five times for the first text size experiment to measure the variation in OCR accuracy for the different text sizes. The mean values of the processed data sets are shown in Figure 4.8. The maximum variation in OCR accuracy from the mean values shown in Figure 4.8 for each set of results was less than 1.8 per cent.

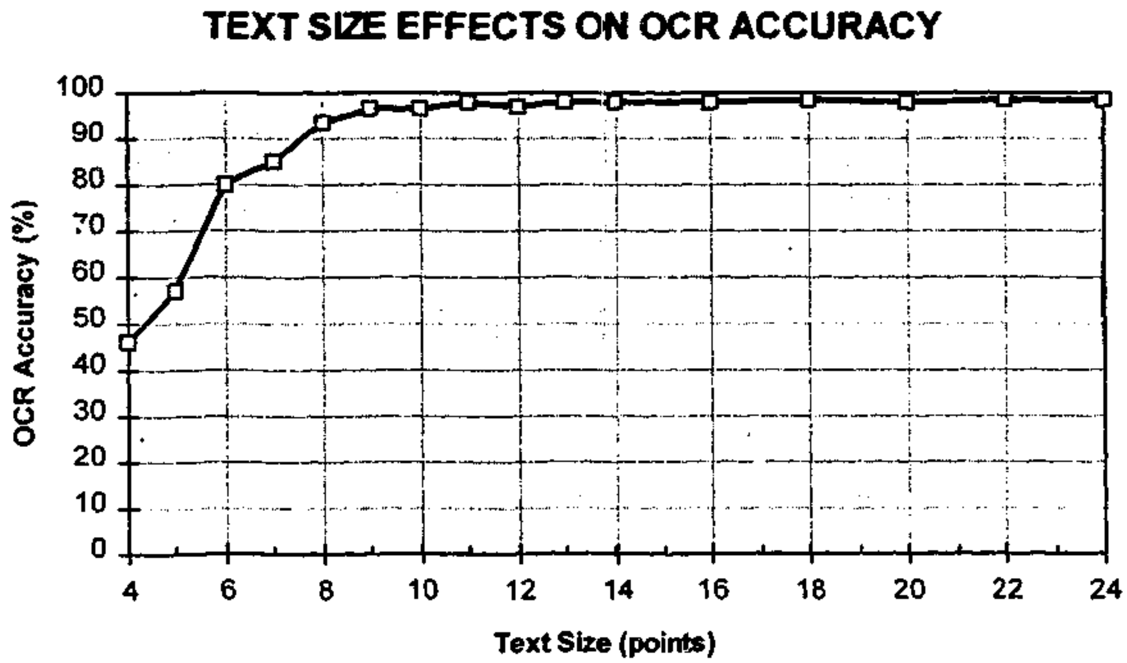


Figure 4.8 Text size effects on OCR accuracy
This chart shows the effects of different text sizes on OCR accuracy for the Helvetica typeface.

Two general observations are made from Figure 4.8 concerning the effect of the text size characteristic variable on the OCR accuracy performance class:

- As the text size falls below 8 point size, the OCR accuracy gradient decreases rapidly.
- For text sizes greater that 14 point size, there is little increase in OCR accuracy.

Analysis of the results of this experiment has led to some explanations for these general observations.

The analysis indicated that when the text size had fallen below 8 point size, the character image data had become insufficient for accurate recognition of the text. As the text size decreased further, so did the OCR accuracy.

Once the text size exceeded 14 points, the analysis showed that the character image data was no longer the primary cause of OCR errors. Increases in text size beyond 14 points did not effect these remaining causes of OCR errors.

The data sets were processed five times for the second text size experiment to measure the variation in OCR speed for the different text sizes. The mean values of the processed data sets are shown in Figure 4.9. The maximum variation in OCR speed from the mean values shown in Figure 4.9 for each of th= sets of results was less than 3.8 characters per second.

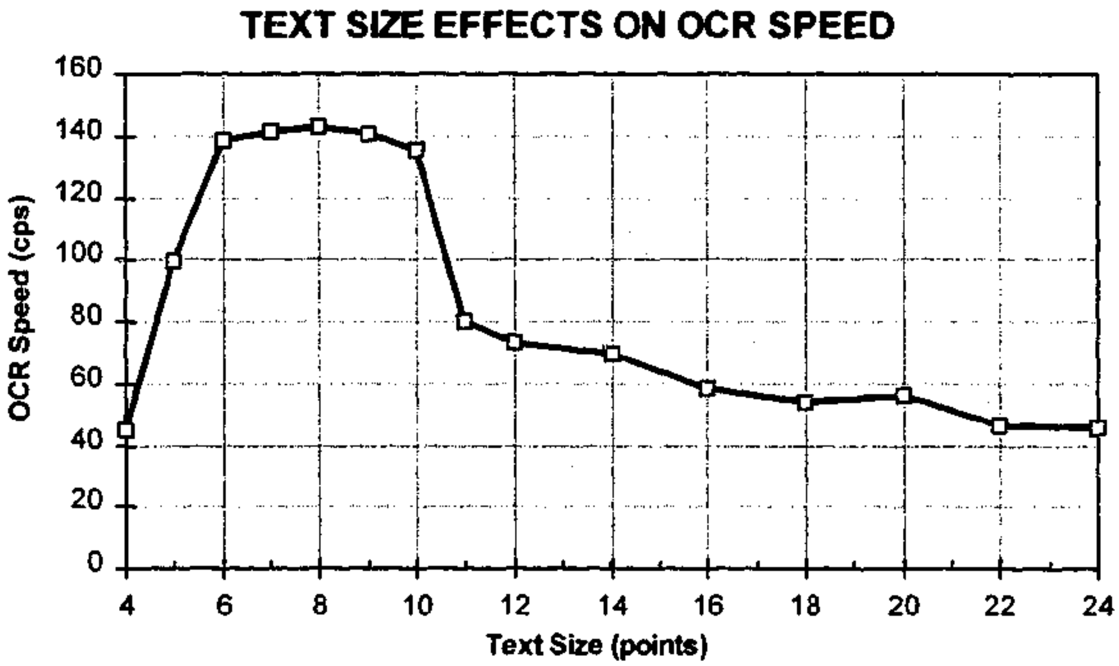


Figure 4.9 Text size effects on OCR speed
This chart shows the effects of different text sizes on OCR speed for the Helvetica typeface.

Several general observations are made from Figure 4.9 concerning the effect of the text size characteristic variable on the OCR speed performance class:

- While the text size is between 6 and 10 point size, the OCR speed stays at a high of approximately 140 characters per second.
- As the text size decreases below 6 point size, the OCR speed drops sharply.

- As the text size increases above 11 point size, the OCR speed drops slowly.

Analysis of the results of this experiment has led to some explanations for these general observations.

For text sizes below 6 points, the analysis attributed the decreasing OCR speed to the additional processing time taken to resolve the inaccuracies which resulted from low text sizes (refer to Figure 4.8).

For text sizes greater than 14 points, the analysis showed that the cause of the gradually declining OCR speed was due to the increase in character image data that accompanied larger text sizes. The larger characters take more data to represent them as images and therefore take a longer time to process.

The tabulated results of the text size experiments are listed in Table E.4 and Table E.5 in Appendix E. Further analysis of the text size experimental results is conducted in Chapter 5.

4.5.4.Image Resolution

Two image resolution experiments were designed. The first image resolution experiment was designed to measure the effects of different image resolutions and text sizes on the OCR accuracy performance class. The second image resolution experiment was designed to measure the effects of different image resolutions on the OCR speed performance class.

The two image resolutions chosen for the first experiments were 200 and 300 DPI. These image resolutions were selected because they are representative of resolutions used for scanning machine printed text documents. The text sizes chosen for both the experiments included 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 16, 18, 20, 22, and 24 point size text. These text sizes were selected because they are representative of text sizes used in machine printed text documents.

The data set used for both experiments was the Courier typeface data set which is described in Section 4.3.3. This data set was selected because it is representative of typefaces used for machine printed text documents.

The data sets were processed five times for the first image resolution experiment to measure the variation in OCR accuracy for the different image resolutions and text sizes. The mean values of the processed data sets are shown in Figure 4.10. The maximum variation in OCR accuracy from the mean values shown in Figure 4.10 for each set of results was less than or equal to 3.1 per cent.

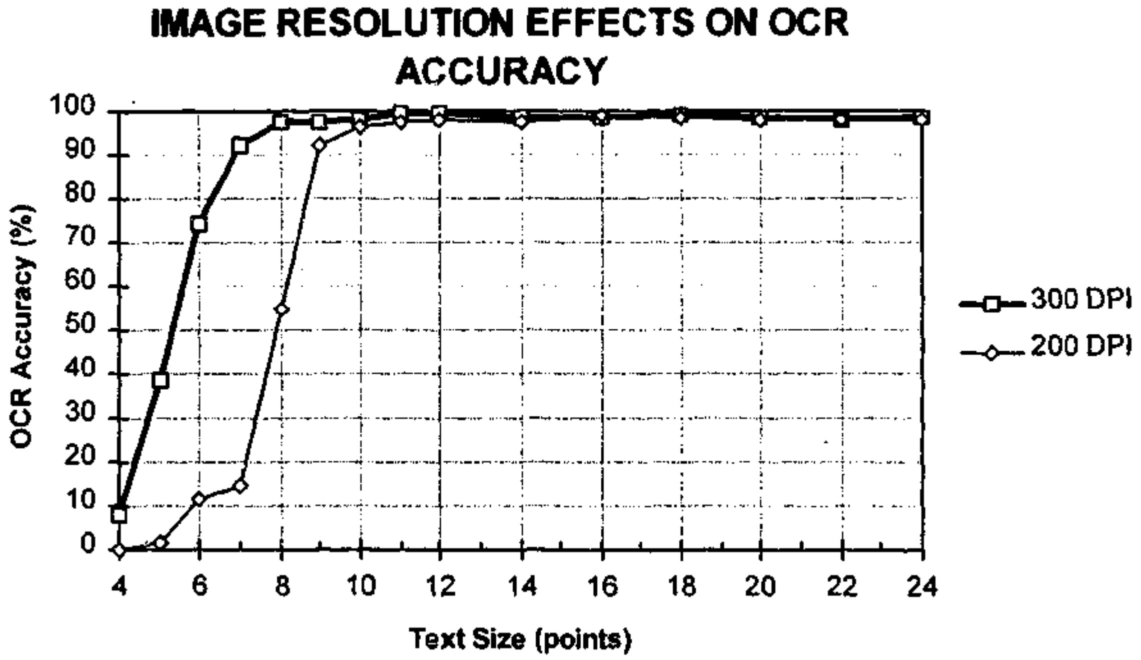


Figure 4.10 Image resolution effects on OCR accuracy
This chart shows the effects of different image resolutions on OCR accuracy for the Courier typeface

Two general observations are made from Figure 4.10 concerning the effect of the image resolution and text size characteristic variables on the OCR accuracy performance class:

- The OCR accuracy curve for the 300 DPI image resolution is a similar shape to the OCR accuracy curve for the 200 DPI image resolution.

- If shifted left by 3 points, the OCR accuracy curve for the 200 DPI image resolution approximates the OCR accuracy curve for the 300 DPI image resolution

Analysis of the results of this experiment has led to some explanations for these general observations.

The analysis showed that the increase in character image data due to an increase in resolution is similar to the character image data increase that can be obtained by increasing the text size (refer to Section 4.5.3). Hence the similarity between the shapes of the two OCR accuracy curves.

The data sets were processed five times for the second image resolution experiment to measure the variation in OCR speed for the different image resolutions and text sizes. The mean values of the processed data sets are shown in Figure 4.11. The maximum variation in OCR speed from the mean values shown in Figure 4.11 for each set of results was less than or equal to 4.7 characters per second.

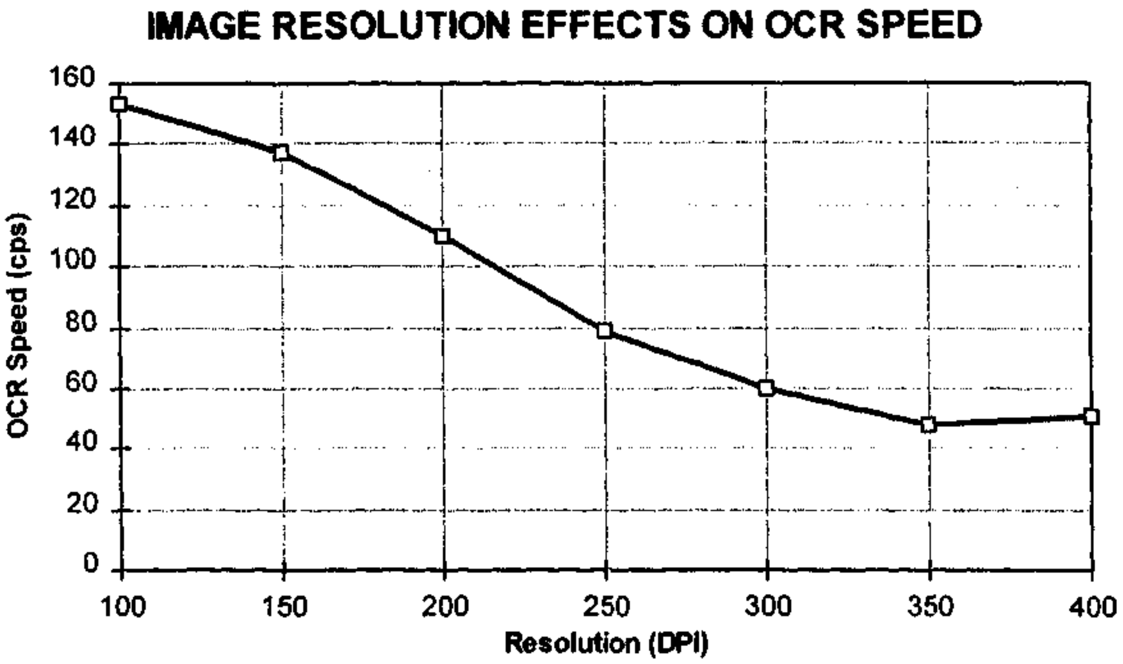


Figure 4.11 Image resolution effects on OCR speed
This chart shows the effects of different image resolutions on OCR speed for the twenty point size Courier typeface

Several general observations are made from Figure 4.11 concerning the effect of the image resolution and text size characteristic variables on the OCR speed performance class:

- As the image resolution increases, the OCR speed decreases.
- As the image resolution increases, the rate at which the OCR speed decreases levels out.
- As the image resolution reaches 400 DPI, OCR speed increases instead of decreases.

Analysis of the results of this experiment has led to some explanations for these general observations.

The analysis showed that with increasing image resolution there is a corresponding increase in character image data. The increased character image data (as previously explained in Section 4.5.2) results in increased processing time and therefore lower OCR speed.

The tabulated results of the image resolution experiments are listed in Table E.6, Table E.7 and Table E.8 in Appendix E. Further analysis of the image resolution experimental results is conducted in Chapter 5.

4.5.5.Printing Device

The printing device experiment was designed to measure the effects of different printing devices on the OCR accuracy performance class.

The printing devices chosen for the experiment included laser, dot matrix and ink jet printers. These printing devices were selected because they are representative of printing devices used for printing text documents. The typefaces chosen for the experiment included Courier, Helvetica and Times Roman. These typefaces were chosen because they are representative of the typefaces used for machine printed text documents.

The data sets used for the experiment were the 12 point size, 300 DPI scanning resolution data sets which are described in Section 4.3.3. These data sets were selected because they consisted of a document text size and scanning resolution which are representative of those used for machine printed text documents.

The data sets were processed five times to measure the variation in OCR accuracy for the different printing devices and font types. The mean values of the processed data sets are shown in Figure 4.12. The maximum variation in OCR accuracy from the mean values shown in Figure 4.12 for each set of results was less than 1.8 per cent.

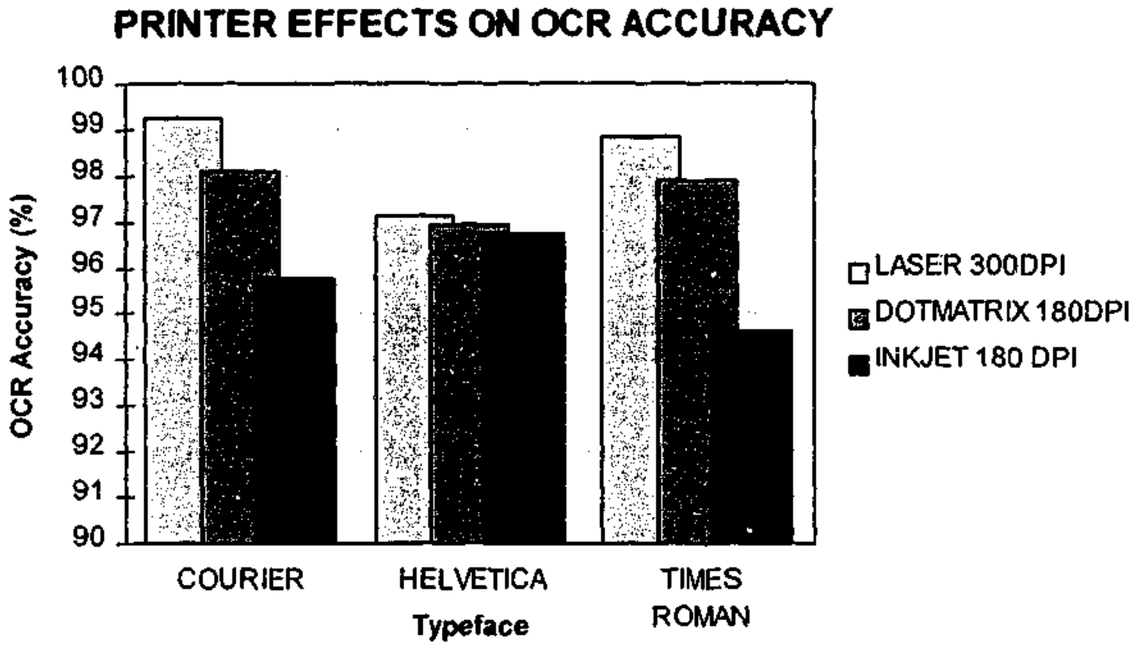


Figure 4.12 Printer effects on OCR accuracy
This chart shows the effects of different printing mechanisms on OCR accuracy for the Courier, Helvetica and Times Roman typefaces.

Two general observations are made from Figure 4.12 concerning the effect of the printing device and font type characteristic variables on the OCR accuracy performance class:

- For all three font types, the laser printer produced a higher OCR accuracy than the dot matrix printer, which produced a higher OCR accuracy than the ink jet printer.

- For the Helvetica typeface (a sans-serif typeface) shows the least variation in OCR accuracy with change in printing device.

Analysis of the results of this experiment has led to some explanations for these general observations.

The analysis of the results showed that the printing quality was the primary cause of the difference in OCR accuracy for this experiment. The laser printer produced a higher resolution (300DPI) printout than the other two printers (180DPI) which resulted in a higher OCR accuracy. The inkjet printer's printout showed occasional ink spatter, giving the characters a rough outline. This was the cause of the inkjet printer's lower OCR accuracy than the dot matrix printer, even though they both have the same printing resolution (180DPI).

The smaller variation in OCR accuracy due to printing device for the Helvetica typeface is attributed to the lack of serifs on this particular typeface. Analysis showed that serifs in combination with lower printing resolution were the cause of several joined characters, thus lowering OCR accuracy for seriffed typefaces printed in lower resolution.

The tabulated results of the printing device experiment are listed in Table E.9, Table E.10 and Table E.11 in Appendix E.

4.5.6.Image Skew

The image skew experiment was designed to measure the effects of different image skew angles on the OCR accuracy performance class.

The image skew angles chosen for the experiment were within the range of zero to seven degrees. This range of image skews was based on a trial experiment which showed OCR accuracy to be zero per cent for image skew greater that seven degrees.

The data sets used for the experiment were the 12 point size, 300 DPI scanning resolution, Courier typeface data sets which are described in Section 4.3.3. These data sets were selected because they consisted of a document text size, scanning resolution and typeface which are representative of those used for machine printed text documents.

The data sets were processed five times to measure the variation in OCR accuracy for the different image skews. The mean values of the processed data sets are shown in Figure 4.13. The maximum variation in OCR accuracy from the mean values shown in Figure 4.13 for each set of results was less than 3.2 per cent.

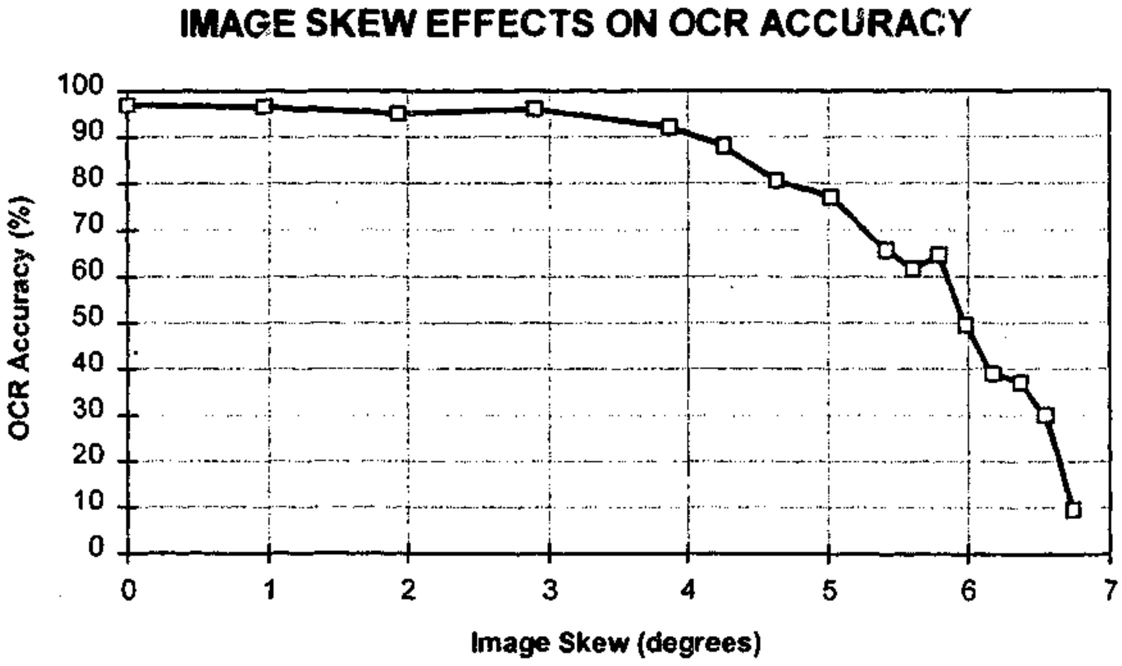


Figure 4.13 Image skew effects on OCR accuracy
This chart shows the effects of image skew on OCR accuracy for the Courier typeface.

Several general observations are made from Figure 4.13 concerning the effect of the image skew characteristic variable on the OCR accuracy performance class:

- For an image skew range of 0 to 3 degrees there is only a marginal decrease in OCR accuracy

- As the image skew exceeds 4 degrees, the OCR accuracy gradient decreases rapidly such that when the image skew reaches 6.8 degrees, the OCR accuracy has fallen to below 10 per cent.
- The is a slight peak in OCR accuracy as the image skew reaches 5.8 degrees.

Analysis of the results of this experiment has led to some explanations for these general observations.

As the image skew began to exceed 4 degrees, the analysis of the results showed that the ability of the OCR software to classify the characters began to decline, therefore leading to a decline in OCR accuracy. This decline continues to the point where the OCR software is unable to classify the characters, there by reducing OCR accuracy to zero.

The slight peak in OCR accuracy as the image skew reached 5.8 degrees is attributed to the OCR software interpreting the text as being italicised. This momentarily increased the OCR accuracy as it rapidly declined with the rise in image skew angle.

The tabulated results of the image skew experiment are listed in Table E.12 in Appendix E.

4.6. SUMMARY OF RESULTS

This section summarises the results obtained for the preliminary experimental work in terms of OCR performance classes and OCR characteristic variables. It examines areas of the results in which further analysis could prove insightful. A comparison between the results reported in this chapter and those presented in the literature is also given.

The results of the preliminary experimental work are summarised into the following general observations:

- Eliminating difficult to recognise characters from the character set can lead to improved OCR accuracy performance.
- Application of different font styles leads to a decline in potential OCR accuracy performance.
- Reduction of the text size characteristic variable below a certain size results in a rapid decrease in OCR accuracy performance.
- The OCR speed performance class is optimised for a specific range of the text size characteristic variable.
- Increasing the image resolution characteristic variable leads to higher OCR accuracy performance for low text sizes and a decrease in OCR speed performance.
- Higher resolution printers offer potentially better OCR accuracy performance.
- Above a certain angle, the image skew characteristic variable severely degrades OCR accuracy performance.

The areas where further experimental work and analysis could prove insightful include the effects of text size and image resolution characteristic variables on the OCR accuracy and OCR speed performance classes. Further experimental data and analysis could establish a relationship between the text size and image resolution characteristic variables and the OCR accuracy and OCR speed performance classes. This further data and analysis is described in Chapter 5.

The results presented in this chapter are similar in some respects to those reported by Baird [8] on document image defect models. However, where Baird concentrates on the document image defects (refereed to as characteristic variables in this thesis) themselves, the work reported in this chapter concentrates on the effects of these document image defects on the performance of OCR systems.

Later work by Ho [11] on the evaluation of OCR accuracy using synthetic data also compares well with the work reported in this chapter. Ho's selection of character sets for experimental work matches that selected for the data sets chosen for this chapters experimental work (refer to Section 4.3.3). Ho however, concentrates on the effects of different image defect parameters from those reported in this chapter.

The results reported in this chapter are consequently original and supported by similar results in the literature. The analysis of the results provides a useful set of guidelines for designing OCR systems and represents an advancement in the knowledge of digital image document processing.

CHAPTER 5

ANALYSIS OF PRELIMINARY EXPERIMENTAL RESULTS

CHAPTER CONTENTS PAGE

5. ANALYSIS OF PRELIMINARY EXPERIMENTAL RESULTS

This chapter analyses and extends the experimental work and results reported in Chapter 4. An introduction to the analysis and the methods used is given. The analysis produces plots of two surface maps. The first surface map shows the combined effects on OCR accuracy of resolution and text size (referred to as the ART curve). The second surface map shows the combined effects of resolution and text size on OCR speed (referred to as the RTS curve). An empirical representation of the ART and RTS curves is presented. A simplified mathematical model is developed to represent the ART curve. A separate approach is presented for modelling the RTS curve. A summary of the extended results and analysis is presented which lists areas of future work and compares the analysis with others reported in the literature.

5.1. INTRODUCTION

The results of the preliminary experimental work, which are reported in Chapter 4, indicate a possible relationship between the text size and image resolution characteristic variables and the OCR accuracy and OCR speed performance classes. By accurately defining the relationship it is possible to model the dependence. The model can then be used to predict OCR performance based on the characteristic variables.

To assist the definition of this relationship, further experimental work was conducted. The range of the image resolution characteristic variable was extended to cover image resolutions from 100 DPI to 400 DPI inclusive, with samples taken every 100 DPI within the range. The range of the text size characteristic variable was maintained at 4 to 24 point sizes.

5.2. THE ART CURVE

The ART (Accuracy, Resolution, Text size) curve is the surface map produced by plotting the OCR accuracy as a function of both the image resolution and text size characteristic variables.

The ART curve is essentially an extension of the results presented in Section 4.5.3 and Section 4.5.4 using extended data sets. The equipment used to conduct the extended experimental work is the same as that used in Chapter 4 for the text size and image resolution experiments.

5.2.1.Empirical Representation

The empirical representation of the ART curve is displayed in Figure 5.1, with text size and image resolution shown on the *x* and *z* axis respectively, and OCR accuracy shown on the vertical *y* axis.

The extended data sets described in Section 5.1 were processed five times to measure the variation in OCR accuracy for the different text sizes and image resolutions. The mean values of the processed data sets are shown in Figure 5.1. The maximum variation in OCR accuracy from the mean values shown in Figure 5.1, for each of the sets of results, was less than 3.9 per cent.

The results depicted at 150, 250 and 350 DPI in Figure 5.1 are linearly interpolated from surrounding results. They are included to improve the surface map continuity.

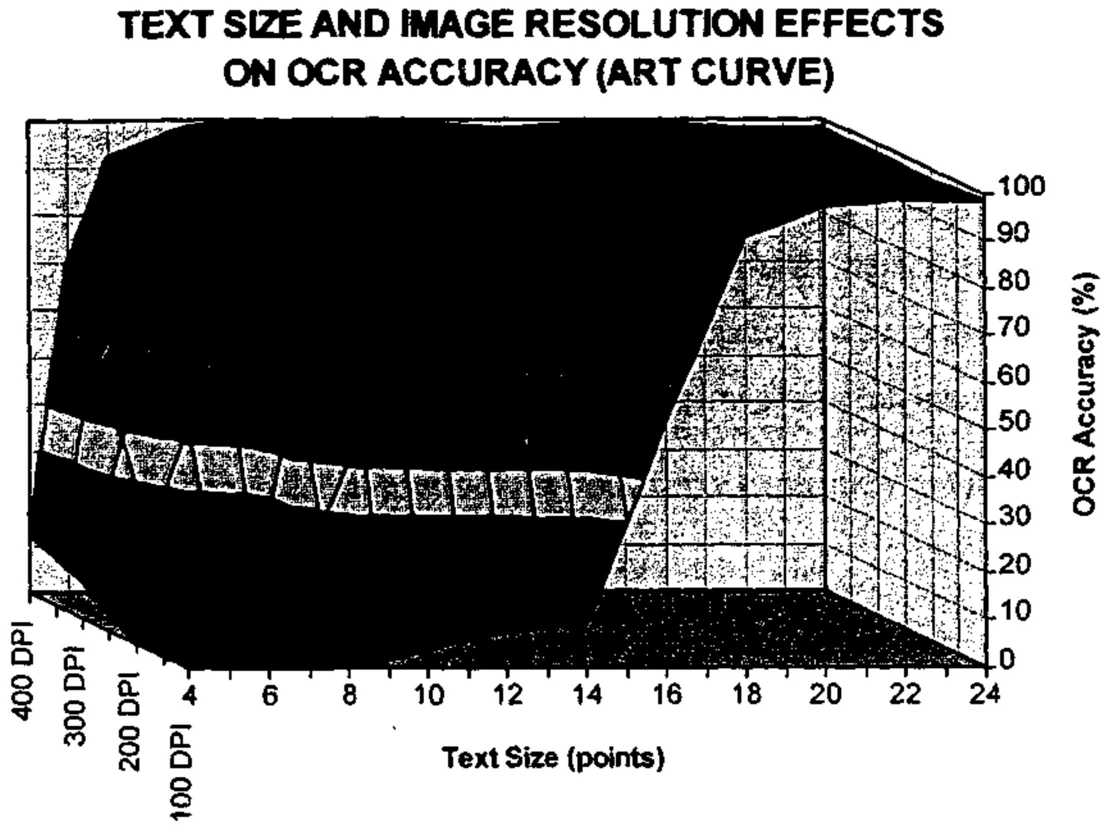


Figure 5.1 Text size and image resolution effects on OCR accuracy
This surface map shows the combined effects of different text sizes and image resolutions on OCR accuracy for the Courier typeface. This surface map is also referred to as the ART curve.

Two general observations are made from Figure 5.1 concerning the combined effect of the text size and image resolution characteristic variables on the OCR accuracy performance class:

- The OCR accuracy curves at 100, 200, 300 and 400 DPI image resolutions are similarly shaped, as per Figure 4.10.
- The shift in the OCR accuracy curve between different image resolutions is less with higher image resolutions.

Analysis of the results of this experiment has led to some explanations for these general observations.

The similar shape of the OCR accuracy curves at 100, 200, 300 and 400 DPI is attributed to the similar effects of increased image resolution and increased text size on the quantity of character image data. The analysis of this similarity is conducted in Section 4.5.4 and Section 5.4.3 and showed that increases in image resolution have a similar effect on OCR accuracy as do increases in text size.

Further analysis of the results reported in this section showed that character image data is proportional to both text size and image resolution. This can be seen in the somewhat hyperbolic shape of horizontal cross sections of Figure 5.1. As a consequence of this relationship between character image data, text size and image resolution, the shift in the OCR accuracy curve between different image resolutions is less with higher image resolutions.

The tabulated results of the ART curve experiment are listed in Table E.13 in Appendix E.

5.2.2.Simplified Mathematical Model

A simplified mathematical model was developed from the empirical results plotted in Figure 5.1. The mathematical model was based on an inverse trigonometric function with twin asymptotes. To describe the model mathematically, the following variables and coefficients are used:

A	OCR accuracy (%)
R	image resolution (DPI)
T	text size (points)
a	y axis offset co-efficient
b	x axis offset co-efficient
c	x axis divisor co-efficient
d	y axis multiplier co-efficient

The model expresses the OCR accuracy, A, as a function of both the image resolution, R, and the text size, T. The model was developed using Microsoft Excel (apparatus code S5 from Apparatus Schedule 2: Software in Appendix D) and is described in Equation 5.1.

$$A = f(R,T) = d\tan^{-1}\left[\frac{(TR-b)}{c}\right] + a \qquad (5.1)$$

The coefficients *a*, *b*, *c* and *d* were optimised using interactive regression techniques with Microsoft Excel so that the mean error between the empirical OCR accuracy values and the models computed OCR accuracy values was reduced to 5.62 per cent. The optimised co-efficient values were:

$$a = 49$$

$$b = 1800$$

$$c = 80$$

$$d = \frac{\pi}{100}$$

The computed surface plot of OCR accuracy as a function of text size and image resolution uses the simplified mathematical model described by Equation 5.1 and is shown in Figure 5.2.

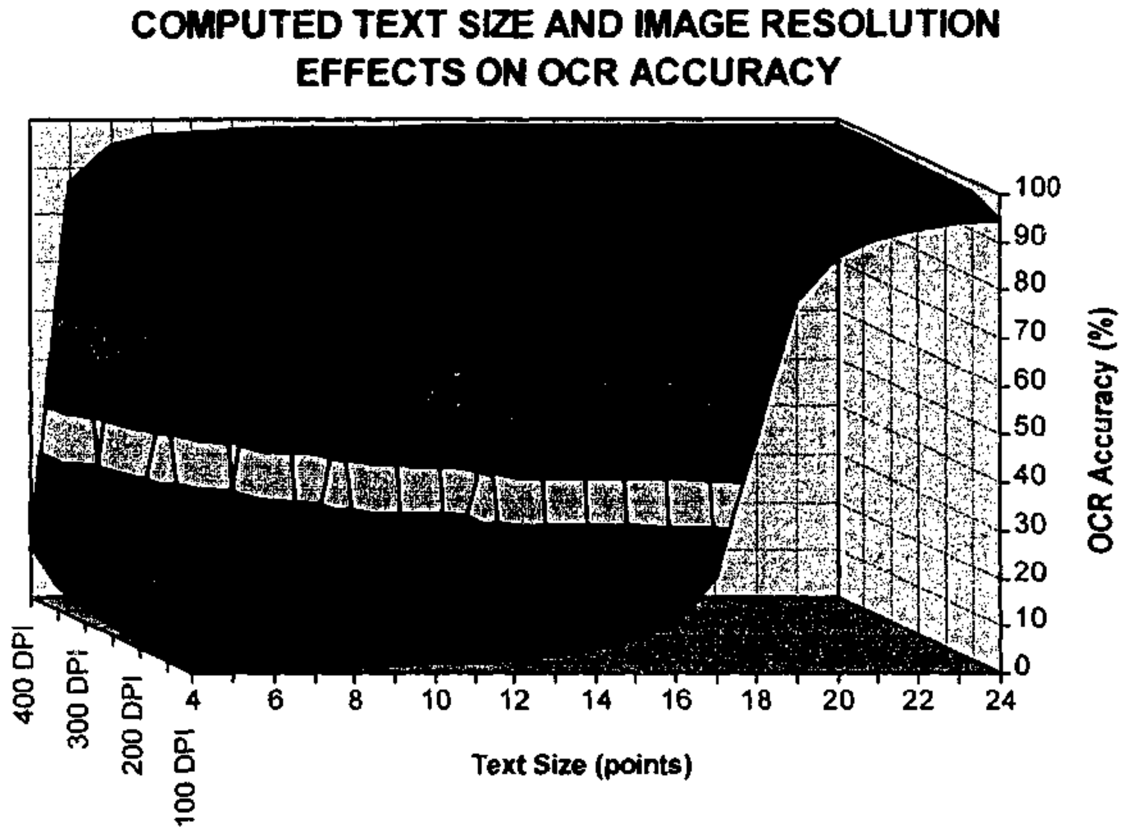


Figure 5.2 Computed text size and image resolution effects on OCR accuracy
This surface map shows the computed effects of different text sizes and image resolutions on OCR accuracy for the Courier typeface.

A comparison of the surface plots depicted in Figure 5.1 and Figure 5.2 reveals a general similarity between the plots. As expected, the experimental perturbations of the empirical plot are not represented by the smoothed plot computed using the simplified mathematical model.

Having established a simplified mathematical model, it can be used in a number of ways. The model can be used to predict the OCR accuracy performance of an OCR system given the text size and image resolution values. The model can also be used to determine the minimum image resolution or text size required to achieve a specified OCR accuracy.

By taking Equation 5.1 and expressing image resolution, *R*, as a function of both the text size, *T*, and OCR accuracy, *A*, Equation 5.2 can be derived.

$$R = f(T, A) = \left[\frac{c \tan\left(\frac{A - a}{d} \right) + b}{T} \right] \tag{5.2}$$

Similarly, by taking Equation 5.1 and expressing text size, *T*, as a function of both the image resolution, *R*, and OCR accuracy, *A*, Equation 5.3 can be derived.

$$T = f(R, A) = \left[\frac{c \tan\left(\frac{A - a}{d} \right) + b}{R} \right] \tag{5.3}$$

Thus the equations can also be used to determine the minimum image resolution (Equation 5.2) or text size (Equation 5.3) required to achieve a specified OCR accuracy.

As an example of the application of the ART model, consider an OCR system which requires an OCR accuracy level of 90 per cent or greater and which must accommodate text sizes as small as seven point size. To meet these two criteria, Equation 5.2 can be used to compute the required image resolution for the OCR system. For *A* and *T* equal to 90 and 7 respectively, and using the optimised co-efficient values, Equation 2 yields an *R* value of 296.5 DPI for the minimum required image resolution for the OCR system.

For the same constrains as the example above, linear interpolation of Figure 5.1 yields an *R* value of 297.0 DPI for the minimum required image resolution for the OCR system. For this example the difference between the computed value and the value obtained through linear interpolation of the empirical results is less that 0.2 per cent.

5.3.THE RTS CURVE

The RTS (Resolution, Text size, Speed) curve is the surface map produced by plotting the OCR speed as a function of both the image resolution and text size characteristic variables. The RTS curve is similar to the ART curve, with the exception that the RTS curve depicts OCR speed whereas the ART curve depicts OCR accuracy.

The RTS curve is essentially an extension of the work presented in Section 4.5.3. using the extended data sets described in Section 5.1. The equipment used to conduct the extended experimental work for the RTS curve data is the same as that used in Chapter 4 for the text size and image resolution experiments.

5.3.1.Empirical Representation

The empirical representation of the RTS curve is displayed in Figure 5.3, with text size and image resolution shown on the *x* and *z* axis respectively, and OCR speed shown on the vertical *y* axis.

The extended data sets described in Section 5.1 were processed five times to measure the variation in OCR speed for the different text sizes and image resolutions. The mean values of the processed data sets are shown in Figure 5.3. The maximum variation in OCR speed from the mean values shown in Figure 5.3 for each of the sets of results was less than 4.2 characters per second.

Initially, the results at 150, 250 and 350 DPI were linearly interpolated from surrounding results. However, given the shape of the surface map, it was decided to improve the detail by extending the data sets to include 150, 250 and 350 DPI resolutions. The final results, including those at 150, 250 and 350 DPI, are shown in Figure 5.3.

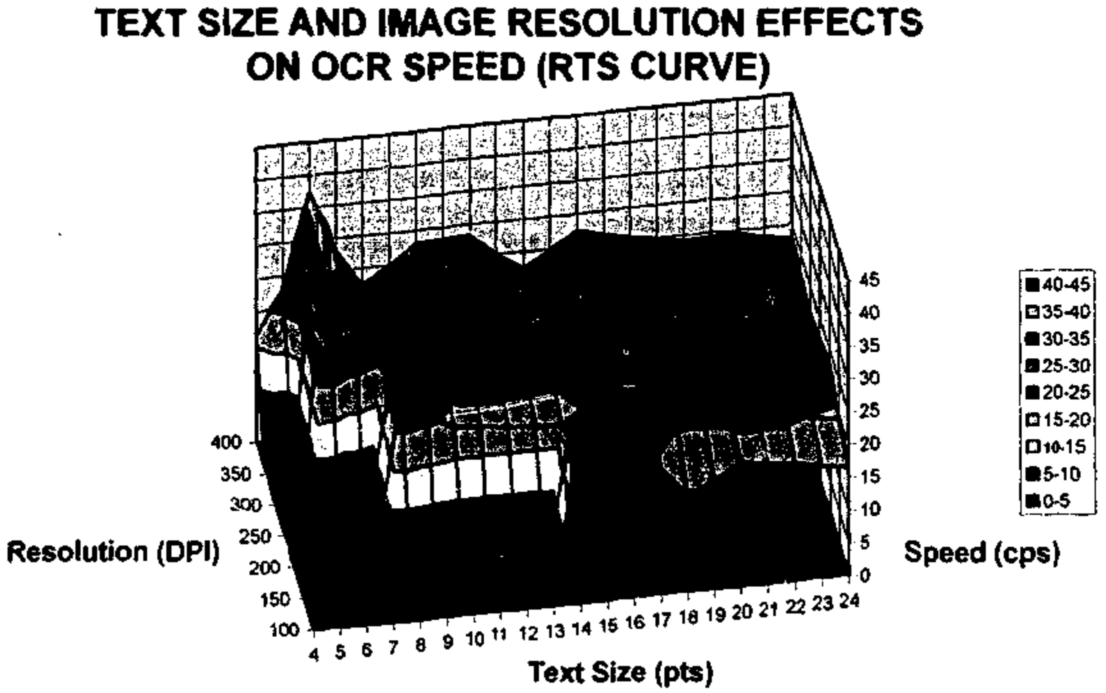


Figure 5.3 Text size and image resolution effects on OCR Speed
This surface map shows the combined effects of different text sizes and image resolutions on OCR Speed for the Courier typeface. This surface map is also referred to as the RTS curve.

Several general observations are made from Figure 5.3 concerning the combined effect of the text size and image resolution characteristic variables on the OCR speed performance class:

- Although the general shape of the surface plot is similar to that depicted for OCR accuracy in Figure 5.1 and Figure 5.2, there is a much greater degree of variation in the results for Figure 5.3.

- There are several peak areas in the surface plot where certain combinations of image resolution and text size values give higher OCR speed performance than other areas.
- Increases in image resolution do not necessarily result in decreases in OCR speed.
- Increases in text size do not necessarily result in increases in OCR speed.

Analysis of the results of this experiment has led to some explanations for these general observations.

The results shown in Figure 5.3 differ slightly from those shown in Section 4.5.3 and Section 4.5.4. Figure 4.9 and Figure 4.11 are similar to appropriately taken vertical cross sections (at 300 DPI and 20 points respectively) of Figure 5.3. As in Figure 4.9, there are peaks in Figure 5.3 where OCR speed is at its highest. The slight difference between Figure 4.9 and Figure 5.3 is attributed to the difference in typefaces used for these experiments.

The effect of image resolution on OCR speed is analysed in Section 4.5.4. The effect of text size on OCR speed is analysed in Section 4.5.3.

The tabulated results of the ART curve experiment are listed in Table E.13 in Appendix E.

A portion of the mean OCR speed measurements at an image resolution of 300 DPI is shown in Table 5.1. Zero values for the OCR speed at 4 and 5 point text sizes indicate that no characters were recognised. Table 5.1 shows that the highest OCR speed of 41.6 character per second occurs at a text size of 6 points.

Table 5.1 OCR speed results for several text sizes
This table shows the OCR speed for several text sizes at an image resolution of 300 DPI and using the Courier typeface.

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
OCR Speed (cps)	0	0	41.6	26.7	19.6	20.9	22.4	25.4	29.2	28.0
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
OCR Speed (cps)	26.8	30.6	35.6	34.8	34.0	27.4	22.9	25.6	29.1	24.3

Given that the shape of the RTS curve is more complex than that of the ART curve, a different solution to mathematical modelling was sought to predict and optimise the OCR speed performance class. A software tool was developed to predict and optimise the OCR speed performance of an OCR system using the RTS curve. This software tool is called OSPO (OCR System Performance Optimiser) and is described in further detail in Chapter 6.

5.4. SUMMARY OF ANALYSIS

This section summarises the results and analysis of the ART and RTS curves. It lists areas in which further analysis could prove insightful. A comparison between the analysis reported in this chapter and those presented in the literature is also given.

The analysis of the preliminary experimental work is summarised into the following general results:

- The relationship between the text size and image resolution characteristic variables, and the OCR accuracy and OCR speed performance classes is defined for a given OCR system.
- The plot of OCR accuracy versus text size and image resolution produces a smooth surface map from which a mathematical model is developed. This model is used for predicting OCR accuracy based on the text size and image resolution characteristic variables.

- The plot of OCR speed versus text size and image resolution produces a rough surface map for which a different approach (OPSO) rather than mathematical modelling is proposed.

There are several areas in the results and analysis presented which could benefit from improvement. As the effort taken to produce the ART and RTS curves is considerable, a less time consuming method would make specific OCR model development more effective. An adaptive sampling method for example, could be used to reduce the number of points required to adequately represent the ART and RTS curves. Integration of the ART and RTS curves is another area of potential improvement to the model, and is addressed in Chapter 6.

The results reported in this chapter are comparable with related results in the available literature. The ART curve described in Section 5.2 and shown in Figure 5.1 as a surface map, with OCR accuracy plotted against the characteristic variables text size and image resolution, is similar to the plots later presented by Ho and Baird [11]. Ho and Baird's plots show accuracy plotted against the synthesized image defect parameters; blur, binarisation threshold and pixel sensor sensitivity. The characteristic variables reported in the thesis directly relate to Ho and Baird's image defect model parameters.

Similar work reported by Blando et al. on prediction of OCR accuracy [21] also compares favorably with the results presented in this chapter. Their approach to OCR accuracy prediction differs however, in that it is not based upon OCR system output (as the results presented in this chapter are), but upon other document characteristics.

The analysis presented in this chapter are therefore unique among those reported in the available literature. They present a useful methodology for predicting the accuracy and speed performance of an OCR system based upon its characteristic variables. The model reported in Section 5.2.2 is original and represents a significant advancement in the knowledge of digital image document processing.

CHAPTER 6

PROTOTYPE OCR SYSTEMS

CHAPTER CONTENTS PAGE

6. PROTOTYPE OCR SYSTEMS

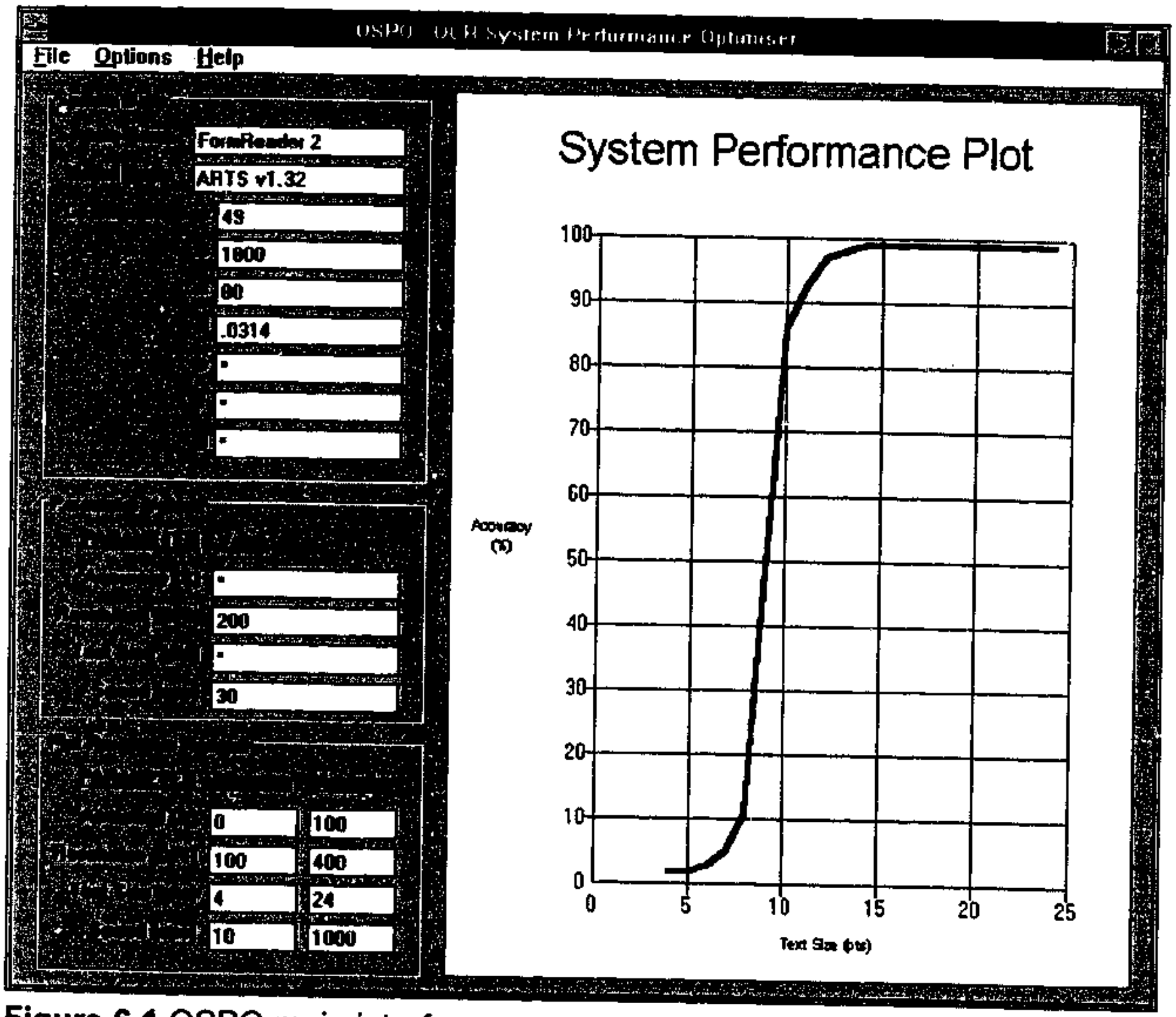
This chapter examines the prototype OCR systems which were developed as part of the research work. An introduction to the prototypes is given which explains the purpose of each of the prototypes. The OCR System Performance Optimiser (OSPO) tool is described, and examples given of its performance. The FormReader OCR system prototype is described and examined in terms of its function and performance.

6.1.INTRODUCTION

Three prototype OCR systems were developed as part of the research work. These prototype OCR systems were called LitBase, OSPO, and FormReader. LitBase is a literature database analysis package which is described in Chapter 3. OSPO is an OCR system performance optimiser which is a software tool which integrates the preliminary experimental results reported in Chapter 4 and analysed in Chapter 5. FormReader is a form processing OCR system incorporating innovative interface features. These interface features include an intuitive graphical user interface with Heads Up Data Entry (HUDE) and an audio feedback and control system. Although FormReader is the last prototype presented in the thesis, it was the first one to be designed. The knowledge and experience gained from developing FormReader formed the basis for the specifications and design of the other two prototypes; LitBase and OSPO.

6.2.OSPO - OCR SYSTEM PERFORMANCE OPTIMISER

OSPO is an interactive software tool developed for optimising the performance of OCR systems. OSPO was developed using Visual Basic (apparatus reference code S7 in Table D.2 of Appendix D) running under Microsoft DOS (S1) and Windows (S2), and operated on a number of personal computers (H8, H9, H10 and H11 in Table D.1). Its origins lay in the experimental work reported in Chapter 4 and the analysis of that work which is reported in Chapter 5. The preliminary experimental work indicated a relationship between the image resolution characteristic variable, text size characteristic variable, OCR accuracy performance class and OCR speed performance class. This relationship was further defined by extending the experimental results which resulted in the ART and RTS curves reported in Section 5.2 and Section 5.3 respectively. While it was possible to develop a simplified mathematical model to represent the ART curve, another approach was necessary for the RTS curve. OSPO was therefore developed to integrate the RTS curve and ART curve into a unified ARTS model for use as an optimising tool for OCR systems. The main interface screen of OSPO is shown in Figure 6.1.



OSPO OCR System Performance Optimiser

FormReader 2
ARTS v1.32
49
1000
80
.0314
*
*
*

*
200
*
30

0 100
100 400
4 24
10 1000

Figure 6.1 OSPO main interface screen
This screen view of OSPO illustrates the graphical interface of the OSPO program. The system performance plot is visible on the right, whilst the system data, parameter values and plot parameter ranges are visible on the left.

Figure 6.1 shows the division of the main interface screen into two main areas. The right part of the screen is reserved for the system performance plot. The left part of the screen is reserved for OCR system data, parameter values and plot parameter ranges. The various parts and functions of the OSPO program are described in Section 6.2.1 while the performance of the OSPO system is described in Section 6.2.2.

6.2.1.System Functions

OPSO uses the integrated ARTS (Accuracy, Resolution, Text size, Speed) model to graphically depict the optimised OCR system. OSPO treats the two performance classes (OCR accuracy and OCR speed) and two characteristic variables (images resolution and text size) as four system parameters. There are three possible modes of operation for OSPO which depend on the number of system parameters supplied:

- If supplied with three of the four system parameters, the ranges of values of the remaining parameter which satisfy the optimisation criteria are computed and displayed.
- If supplied with two of the four system parameters, a curve of possible ranges of the remain parameters which satisfies the optimisation criteria is computed and displayed.
- If supplied with one of the four system parameters, a surface map of possible ranges of the remain parameters which satisfies the optimisation criteria is computed and displayed.

OSPO requires a set of system co-efficient values and ARTS data set to be loaded or entered before computational work can be undertaken. Different sets of system co-efficient values and ARTS data sets can be stored and retrieved for each different system being optimised. The plot ranges for the computed system parameters can also be specified, stored and retrieved. The example of the OSPO main interface screen which is given in Figure 6.1 shows the OCR accuracy performance class as a function of the text size characteristic variable.

OSPO can print its graphical display or export it to another program. It can also print the OCR system data, parameter values and plot parameter ranges.

The source code iisting for version 2.00 of OSPO is given in Appendix A.

6.2.2.System Performance

To test the performance of the OSPO system, three different OCR systems were optimised using OSPO and the results compared. These three OCR systems were WordScan, OmniPage and FormReader.

An overall optimisation level, Z, is defined to provide a numerical basis for the comparison of these OCR systems. The overall optimisation level achieved by OSPO for the three OCR systems is described using the following variables:

Z overall optimisation level (%)

O_A OCR accuracy performance class optimisation level (%)

O_S OCR speed performance class optimisation level (%)

A_B OCR accuracy before optimisation (%)

A_O optimised OCR accuracy using OPSO (%)

S_B OCR speed before optimisation (cps)

S_O optimised OCR speed using OSPO (cps)

The OCR accuracy performance class optimisation level, O_A, is defined as the ratio of the OCR accuracy before optimisation, A_B, and the optimised OCR accuracy using OSPO, A_O, and is given by Equation 6.1.

$$O_A = \left[\frac{A_B}{A_O}\right] \tag{6.1}$$

The OCR speed performance class optimisation level, O_S, is defined as the ratio of the OCR speed before optimisation, S_B, and the optimised OCR speed using OSPO, S_O, and is given by Equation 6.2.

$$O_S = \left[\frac{S_B}{S_O}\right] \tag{6.2}$$

The overall optimisation level, Z, is defined as the multiple of the OCR accuracy performance class optimisation level, O_A, and the OCR speed performance class optimisation level, O_S, and is given by Equation 6.3.

$$Z = O_A \times O_S \tag{6.3}$$

Before the different OCR systems could be optimised using OSPO, it was necessary to obtain the ARTS data for each OCR system. Experimental work identical to that described in Chapter 5 was conducted to determine the ARTS data for each OCR system. Once the ARTS data was obtained, each system was configured identically with default settings. The image resolution was set to 300 DPI and the text size set to 11 points. The Courier typeface was used as per Chapter 5.

An example of the overall optimisation level calculation is given for the OmniPage OCR system in Equation 6.4.

$$Z = O_A \times O_S = \left[\frac{A_B}{A_O}\right] \times \left[\frac{S_B}{S_O}\right] = \left[\frac{98.3}{99.2}\right] \times \left[\frac{35.6}{41.6}\right] = 84.8\% \tag{6.4}$$

The table of the results of the optimisation test are shown Table 6.1. Since the best possible OCR accuracy and best possible OCR speed may not occur at the same values of image resolution and text size, it may not be possible for an OCR system to achieve a value of 100% for the optimisation level defined here. The optimisation level results shown in Figure 6.2 appear to be fairly consistent for the initial conditions specified.

Table 6.1 Optimisation levels for various OCR systems
This table shows the overall optimisation levels calculated using Equation 6.3 for various OCR systems.

[illegible]	[illegible]
WordScan	82.5%
Omnipage Professional	84.8%
FormReader	80.2%

6.2.3.OSPO Summary

OSPO integrates the ART model and RTS curve reported in Chapter 5 into a software tool for OCR system performance optimisation. When tested on three different OCR systems, OSPO was able to achieve a mean optimisation level of 82.5 per cent (calculated using Equation 6.3).

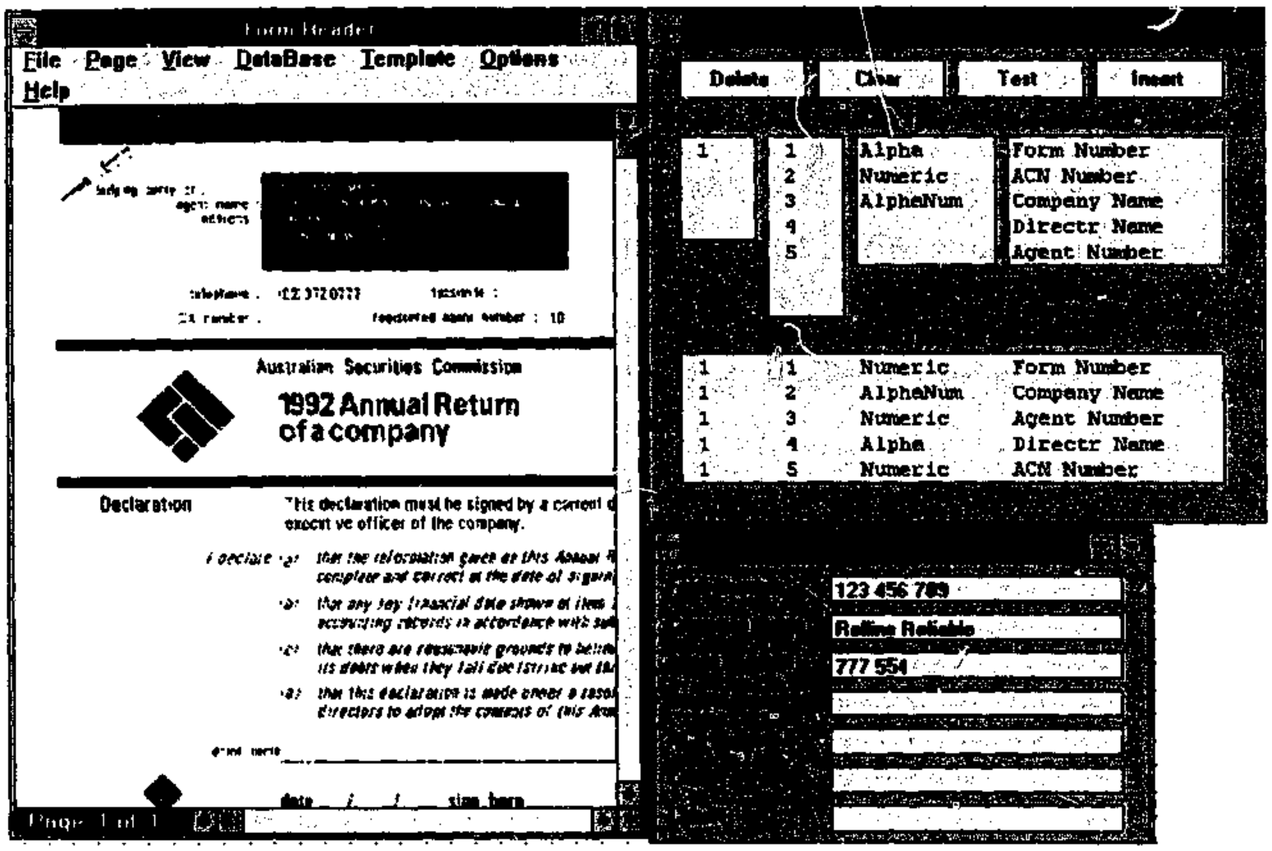
As a tool for analysing OCR systems, OSPO represents an unique and useful contribution to the area of digital image document processing.

6.3.FORMREADER

FormReader is a form processing OCR system incorporating several innovative interface features. These features include the HUDE system, audio feedback of numeric data and a simple voice recognition system. These system features are explained in more detail in Section 6.3.1. FormReader was developed using Visual Basic (apparatus reference code S7 in Table D.1 of Appendix D) running under Microsoft DOS (S1) and Windows (S2), and operated on a number of networked personal computers (apparatus reference codes H8, H9, and H10 in Table D.2). A flatbed scanner (H1) was used with FormReader to digitise the laser (H14) printed forms. Form image display was made on 21 inch monitors (H6).

The OCR software (S3) integrated into the FormReader system was Omnipage Professional.

FormReader was designed to read forms used by the Australian Securities Commission (ASC). It can be readily adapted to read most other types of forms by designing appropriate templates. The main interface screen of FormReader is shown in Figure 6.2. The main image viewing screen is shown on the left window. The template controls are shown in the top right window. The database entry form is shown in the bottom right window.



Australian Securities Commission

1992 Annual Return of a company

Declaration

This declaration must be signed by a current d
executive officer of the company.

I declare (a) that the information given on this Annual R
complete and correct at the date of signin

(b) that any key financial data shown at item
accounting records in accordance with su

(c) that there are reasonable grounds to belie
its debts when they fall due (strike out th

(d) that this declaration is made under a reso
directors to adopt the contents of this Ann

print name

date / / sign here

```
1     1   Alpha      Form Number
      2   Numeric    ACN Number
      3   AlphaNum   Company Name
      4              Directr Name
      5              Agent Number

1     1   Numeric    Form Number
1     2   AlphaNum   Company Name
1     3   Numeric    Agent Number
1     4   Alpha      Directr Name
1     5   Numeric    ACN Number
```

123 456 789

Rallina Reliable

777 554

Figure 6.2 FormReader Main Interface Screen
This screen view of FormReader shows three separate parts of the program. The main image viewing screen is shown on the left window. The template controls are shown in the top right window. The database entry form is shown in the bottom right window.

FormReader was developed as a test-bed program to verify the results reported in Chapter 4 and Chapter 5. It also provided experience in designing digital image processing systems which was later used in the development of the other two prototypes; LitBase and OPSO.

6.3.1.System Functions

The FormReader form processing is shown in Figure 6.3 as a process flow diagram. Incoming forms are digitised and stored in an image database where they are available for display and manipulation. The physical forms are stored for retrieval or disposal. Form images are taken from the image storage and fed through a keyfield selection mask which removes areas from the forms which contain non-required text, such as form layout instructions. The resultant masked image is then made available to the text recognition module which converts the image into text. In some cases, OCR errors occurred which were due to poor image quality or incorrect data. For images which contain OCR errors, interaction with a text storage database and/or an error correction operator is used to resolve the errors. This occurs before the text is forwarded to the text storage database where it can provide further feedback to the text recognition module. The performance of the FormReader form processing system can be monitored by a controller and reporting mechanism to provide automatic quality control.

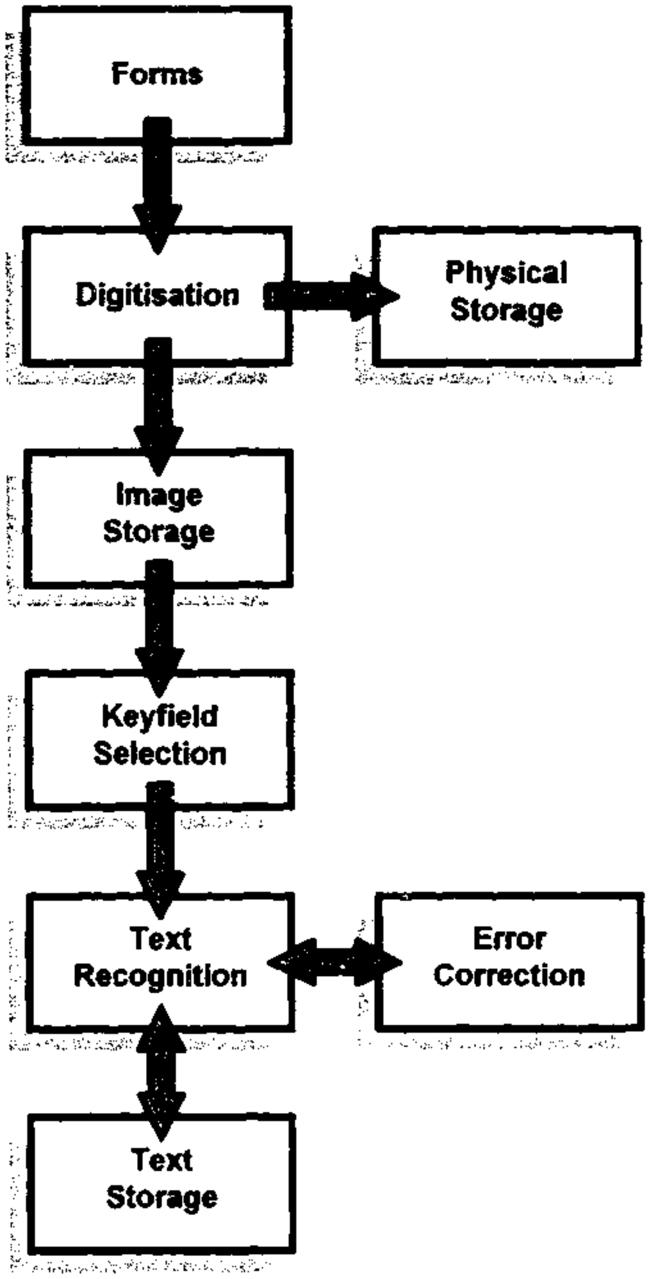


Figure 6.3 FormReader process flow diagram
The diagram shows the sequence of the major processes that occur in the FormReader system. Forms are digitised, their images stored in an image database and the physical forms are stored. The keyfields of the image are selected and their text is recognised. The recognised text is placed in a text storage database. The text storage database and error correction operators resolve errors detected by the text recognition system.

The FormReader system architecture is illustrated in Figure 6.4. The FormReader control software lies at the center of the system. The operator can communicate with the FormReader control software via the graphical and audio user interface. The FormReader control software is linked to the image viewing, audio, and scanning and OCR modules. The operator and modules can access the image and text databases via the FormReader control software. The system controller can monitor the performance of the FormReader form processing system directly through FormReader control software.

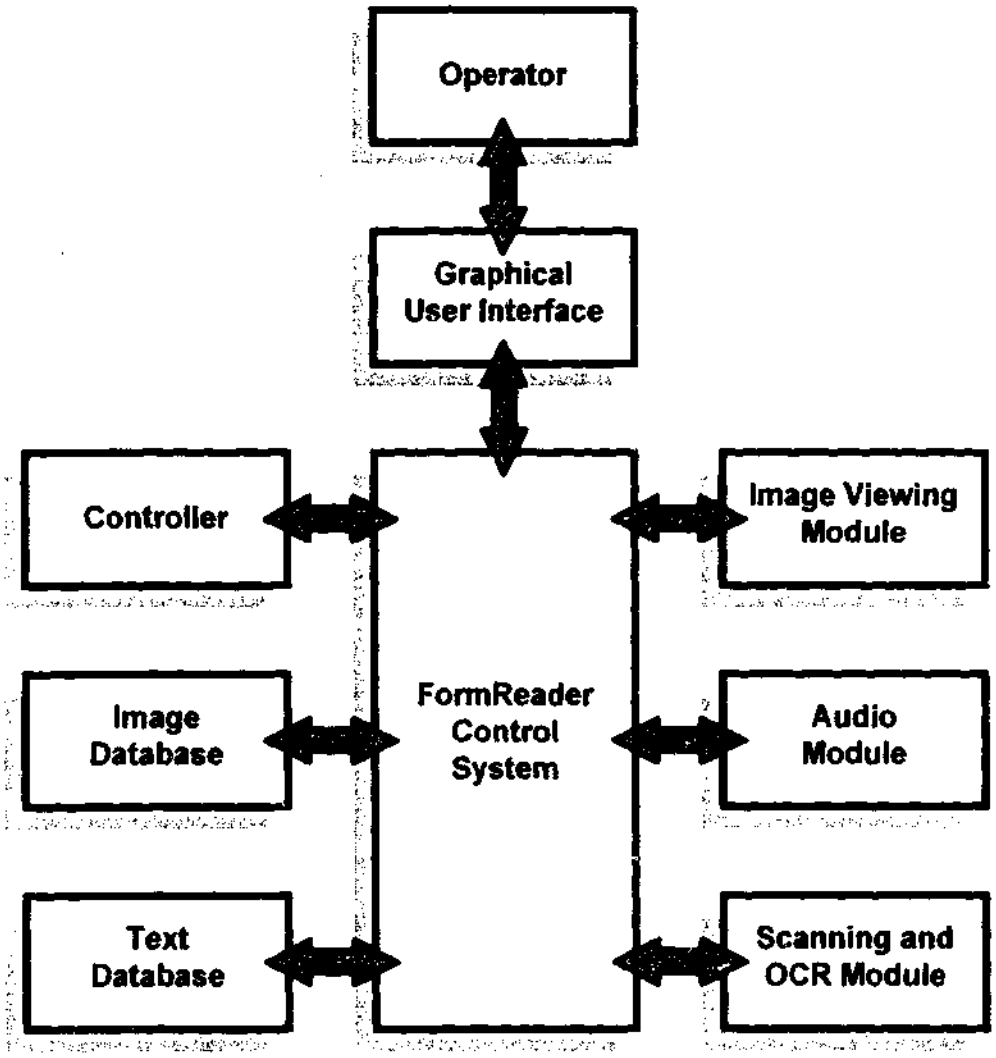


Figure 6.4 FormReader system architecture
This block diagram shows the architecture of the FormReader system. The operator communicates with the FormReader control system via the graphical user interface. The external modules on the left can access the databases on the left via the FormReader control system. The system controller can communicate directly with the FormReader control system.

The FormReader form processing system is designed to function across a network of machines and can be configured to operate in various modes to accommodate varying processing requirements. Should a higher proportion of forms be of poor quality or hand written, then more machines can be switched to a fully operator driven mode which allows the operator to enter the data into the database from an on-screen image of the form.

6.3.2.User Interface Features

An important consideration when designing the FormReader form processing system was the user interface. The interface was designed to be as productive and non-fatiguing as possible by using a graphical mouse-driven system supplemented by audio feedback facilities. The user interface is composed of two parts; the graphical interface and the audio interface.

6.3.2.1.Graphical Interface

The graphical interface used for FormReader is similar to that described in Chapter 3 for LitBase. The HUDE system is an additional and innovative graphical interface feature of FormReader.

Heads up data entry for non-recognisable forms was implemented to eliminate the eye-fatigue [79] associated with re-focusing between printed forms and a data entry screen. The image of the form is displayed on-screen beside the data entry window. The operators type the values they see on the image into the data entry window. An example of the screen layout of the form processing prototype is show in Figure 6.2. On the left is the form image, with zooming and panning facilities for display and manipulation. The top, right area of the screen contains template controls for masking keyfield areas for OCR. The database entry area is located to the bottom left of the screen, where recognised data is fed, or where the operator may type the data. The screen layout is dynamically adjustable, with each of the individual windows capable of being moved and re-sized.

The resolution of the operator's screen is important for displaying the form images with adequate image clarity. For displaying a full A4 sized form image on-screen, along with a data-entry window, and to be able to provide sufficient image quality for an operator to perform HUDE, a megapixel (one million picture elements) display [80] or better is desirable. This avoids zooming and panning across portions of the image. A screen resolution of 1280 x 1024 pixels on a 21 inch monitor (apparatus reference code H6) is capable of displaying a full A4 sized form image at approximately 80 DPI. This was found to be satisfactory for an operator to read 10 point size text or greater without inducing significant eye fatigue. Non-interlaced screens were used for image display, as the screen flickering observed when interlacing was found to increase operator fatigue.

6.3.2.2.Audio Interface

Audio prompts and feedback of data to the operators are used to augment visual prompts and on-screen data display. This helps prevent overloading the operators' vision by diverting information to their hearing [81]. The operators need not shift their sight from one point of the screen to another and back again to verify data entered or to acknowledge system prompts.

Audio feedback is obtained by playback of digitised human voice samples. Digitised samples were used instead of synthesized speech because the synthesized speech modules tested did not sound natural and were prone to mis-pronunciation. A digitised sample must therefore exist for a particular word or phrase to be 'spoken', and only a limited vocabulary of common words can be maintained because of the sample's large file size. Numeric data requires only a few samples to be 'spoken' to the operator; the numbers 'zero' through 'nineteen', 'twenty', 'thirty', 'forty' ... 'ninety', 'hundred', 'thousand', 'million' and 'and'. The numeric samples need only be played back in correct sequence to properly pronounce any number required. Prompts such as error reporting (e.g. 'invalid entry') are also be handled via audio feedback to prevent distracting the operators vision.

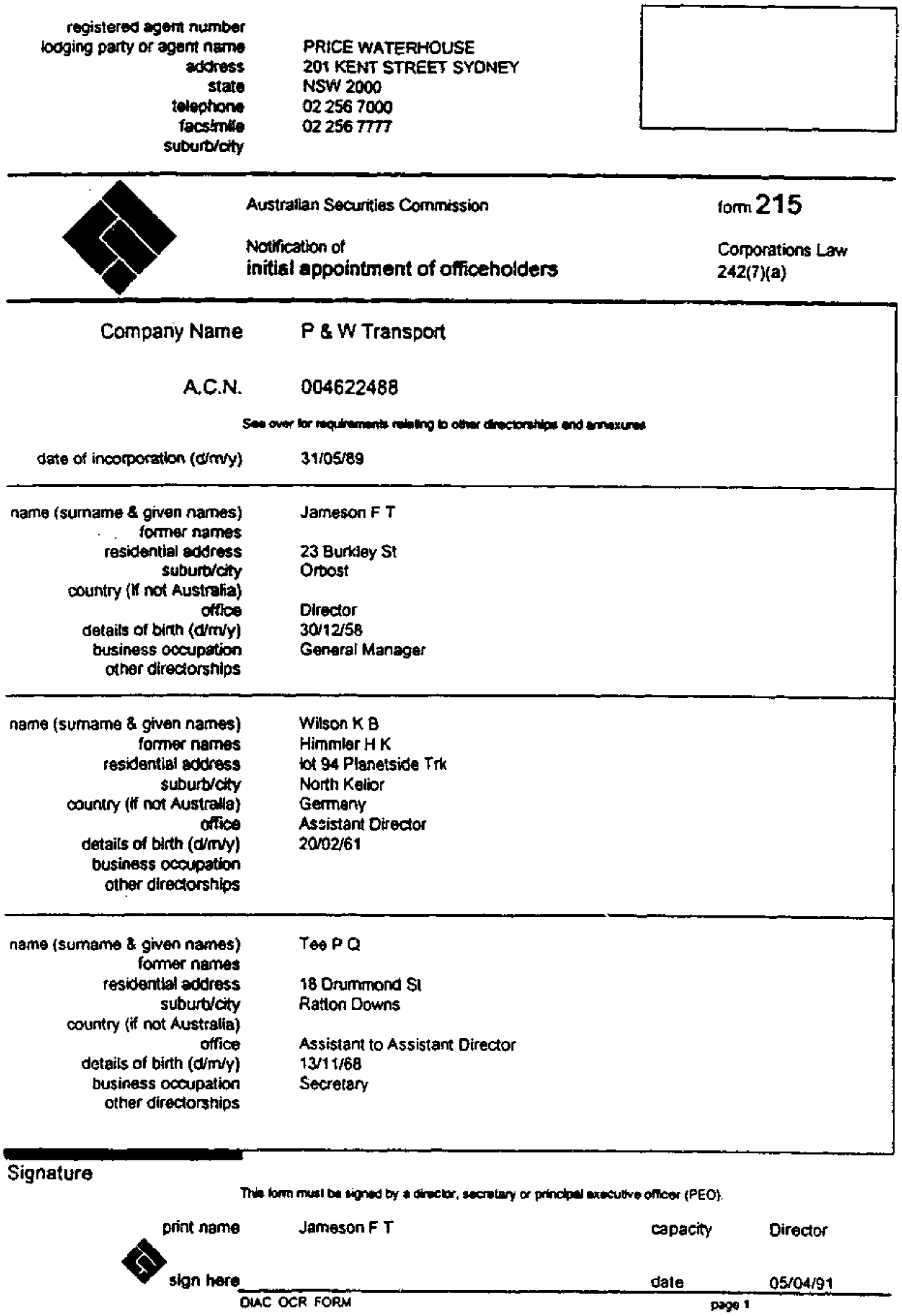
6.3.3.System Performance

This section examines the performance of the FormReader forms processing system in terms of two of the OCR system performance classes described in Chapter 4.4; OCR speed and OCR accuracy.

6.3.3.1.Speed Performance

The speed at which forms can be processed through the prototype FormReader system is dependent upon the number of keyfields and quantity of text per keyfield to be recognised per form. The particular PC used for the speed test was a 33 MHz 80486 (H8). The time taken for a 50 form data set of single page ASC forms containing seven fields of actual company information to be processed was an average of 28 seconds per form including form scanning time. This yields a form processing speed of 128 forms per hour.

The speed performance in this case is expressed in forms per hour, rather than characters per second because changes in the quantity of text in the keyfields did not significantly alter the time taken to process the form. An example of the redesigned ASC test form is shown in Figure 6.5. Several keyfields have had data entered into them in a 12 point sized Helvetica typeface and the form has been scanned at 300 DPI.



registered agent number	
lodging party or agent name	PRICE WATERHOUSE
address	201 KENT STREET SYDNEY
state	NSW 2000
telephone	02 256 7000
facsimile	02 256 7777
suburb/city	

Australian Securities Commission

form 215

Notification of
initial appointment of officeholders

Corporations Law
242(7)(a)

Company Name	P & W Transport
A.C.N.	004622488

See over for requirements relating to other directorships and annexures

date of incorporation (d/m/y)	31/05/89

name (surname & given names)	Jameson F T
former names	
residential address	23 Burkley St
suburb/city	Orbost
country (if not Australia)	
office	Director
details of birth (d/m/y)	30/12/58
business occupation	General Manager
other directorships	

name (surname & given names)	Wilson K B
former names	Himmler H K
residential address	lot 94 Planetside Trk
suburb/city	North Kelior
country (if not Australia)	Germany
office	Assistant Director
details of birth (d/m/y)	20/02/61
business occupation	
other directorships	

name (surname & given names)	Tee P Q
former names	
residential address	18 Drummond St
suburb/city	Ratton Downs
country (if not Australia)	
office	Assistant to Assistant Director
details of birth (d/m/y)	13/11/68
business occupation	Secretary
other directorships	

Signature

This form must be signed by a director, secretary or principal executive officer (PEO).

print name	Jameson F T	capacity	Director
sign here		date	05/04/91

DIAC OCR FORM

page 1

Figure 6.5 Example of scan of ASC test form

This diagram shows the 300DPI scan of a redesigned A4 sized ASC form which has had data entered into several of the keyfields in a 12 point sized Helvetica typeface.

It should be noted that while the speed of 128 forms per hour is for a single machine, multiple machines do not provide machine x 128 processing rates because of networking overheads which constrict form flow between machines. Using a faster scanner or pre-scanned forms could improve performance to over 300 forms per hour, since approximately 60 per cent of the time taken to process each form involved scanning. Upgrading the system to a faster PC (H9) improved the speed performance to 182 forms per hour.

6.3.3.2.Accuracy Performance

The OCR accuracy for the above sample set was 98.2 per cent before error correction by an operator, and 100 per cent after error correction. The OCR accuracy level reported for this test (prior to error correction) occurs between the values of 97.1% and 99.8% reported in Section 4.5.1 for full character set and limited character set OCR accuracy's for the 300DPI image resolution,12 point size, Helvetica typeface data sets.

The keyfield data used in the FormReader speed test is composed mostly of characters from the limited character set, but does include some characters from the full character set (as shown in Figure 6.5). The OCR accuracy result for FormReader therefore appears to be consistent with OCR accuracy results reported in Chapter 4.

6.3.4. FormReader Summary

This section summarises FormReader in terms of its interesting interface, performance and contribution to the area of knowledge.

The FormReader prototype introduced some innovative interface features to OCR form processing systems, including the HUDE system and audio feedback. While not incorporated into the other two prototypes developed for this research, these interface features demonstrated the effective application of these new technologies.

The OCR accuracy performance of the FormReader prototype is 98.2 percent for the test forms used in Section 6.3.3, while the processing speed was 128 forms per hour.

As a prototype OCR forms processing system, FormReader represents an innovative and useful contribution to the area of digital image document processing.

CHAPTER 7

OCR SYSTEM MODEL DEVELOPMENT

CHAPTER CONTENTS PAGE

7. OCR SYSTEM MODEL DEVELOPMENT

This chapter covers the development of the OCR system models. An introduction is given to the model development which describes the five different evolutionary stages and the methods by which these stages were reached. All of the five models are presented with a complete description of the structure, design, and analysis of each model.

7.1. INTRODUCTION

There were two main reasons for modelling the OCR systems. The first reason was to describe both qualitatively and quantitatively the individual processes which occur in the OCR system. The second reason was to use the models to determine the response of OCR systems under specific conditions. This second reason is covered in Chapter 8 by model experimentation and analysis.

The selection of the overall modelling methodology was based on the requirements of the OCR system models. The models had to represent a discreet document processing system. They had to able to be developed in stages and be able to interface with the OCR systems previously described in Chapter 6. A discreet modelling methodology was therefore selected to meet these requirements. Similar methodologies were employed by other model developers listed in Section 2.6.2.

The models were all implemented using a spreadsheet package called Microsoft Excel version 5.0. There were two reasons for choosing this spreadsheet package over dedicated modelling packages. Excel has a built-in programming language which enables the model to be tailored to suit its design requirements. It has a number of plug-in modules to perform the necessary statistical and modelling operations required by the model. Having justified the use of Excel, it can be said that while it is perhaps more adaptable than a dedicated modelling package, it can require a lot more effort to achieve the same results as a dedicated modelling package.

The method by which the model was developed from its first form to its last was to take each model as it was developed and analyse its performance in terms of emulating the real world system. By examining the shortcomings of each model and implementing improvements, it was possible to evolve the model to its next stage of development. This model development process was carried out four times to arrive at the final model, and is depicted in Figure 7.1. The development of the model is shown in a series of increasingly complex process flow diagrams. These iconised process flow diagrams are smaller versions of the full sized diagrams shown later in this chapter. The final model described in this chapter does not purport to be without possibility of improvement. It is however sufficiently sophisticated to enable a real comparison between its performance and that of real world systems.

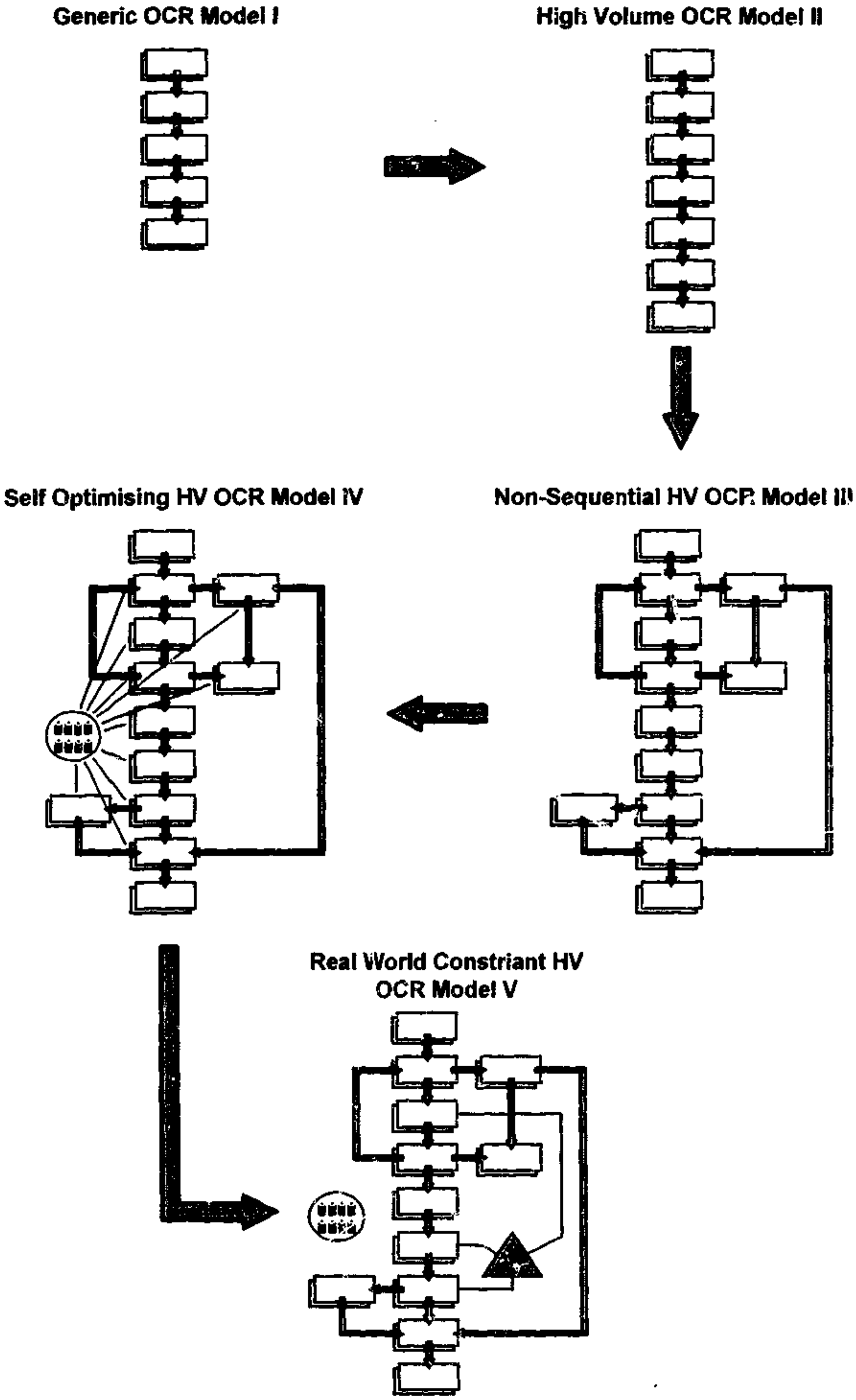


Figure 7.1 Visualisation of OCR model development

The development of the OCR model is shown using process flow diagrams at various stages of model development. The process flow diagrams are iconised and simplified so that the essential differences between them are more apparent.

7.2. GENERIC OCR MODEL I

The generic OCR model I represents a simple OCR model similar to the many presented in the literature [2], [3], [14], [38], [64], [82-88]. It forms the basis for the development of the more sophisticated models. The knowledge gained from the experimental results reported in Chapter 4 and analysis in Chapter 5 is used to direct the evolution of the model from the generic form presented in the literature. Integration of the optimisation procedures reported in Chapter 6 are integrated with the model in the later stages of the model's evolution. The generic model's structure is discussed in the following section. This is followed by an analysis of the model.

7.2.1. Model I Structure

The generic or elementary OCR model exhibits the basic processes which occur in typical OCR systems. Figure 7.2 shows the process flow of an elementary OCR model from physical documents to electronic text. The model has three basic processes; digitisation, segmentation and classification. These processes are described in Section 2.1.3.

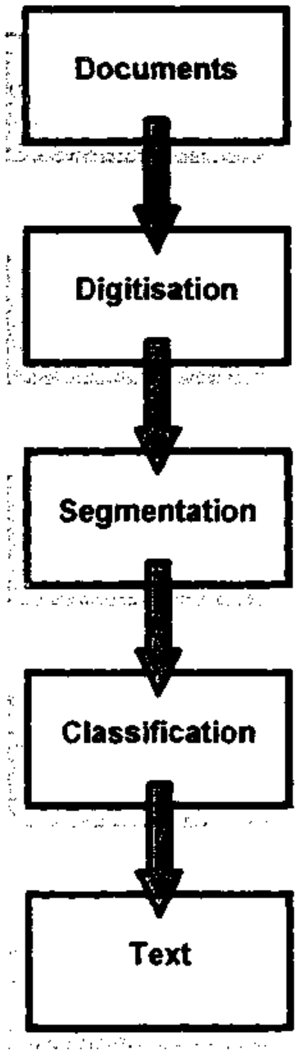


Figure 7.2 Generic OCR model I
This simple OCR model shows the basic processes which occur in typical OCR systems as physical documents are transformed into electronic text.

7.2.2.Model I Analysis

While the generic OCR model is good for illustrating the OCR processes which occur in typical OCR systems, it does not adequately show the quality control processes which occur in the high volume document environment.

7.3. High Volume OCR Model II

A specific OCR model for the high volume document environment is required to describe the process flow in this specialised and more complex OCR system environment. The high volume OCR model's design criteria is based on real world observations and analysis of the high volume document processing environment. As stated in Chapter 1, the Australian Securities Commission's (ASC) National Information Processing Centre forms a major part of the background of the development work presented in the thesis. Direct observations of their digital image document processing system (which in 1993 processed over 1.3 million documents [89]) are used to establish the high volume design criteria for the OCR syste·n model.

The sequential nature of the process flow distinguishes this model from the more advanced models which evolved from this one. The model's structure and design are discussed in the following sections. This is followed by an analysis of the model.

7.3.1.Model II Structure

The structure and process flow in the high volume OCR model II is illustrated in Figure 7.3.

For reference purposes and for later analysis, each process in the high volume document OCR model II is assigned a process number *n* to identify it. Each process number *n* is described by two transfer parameters:

D_n delay for process *n* (minutes)

O_n operator numbers for process *n* (operators)

The process delay represents the time taken for a document to be operated on by that process and then passed onto the next process. The operator numbers represent the maximum number of documents which can be operated on concurrently during that process.

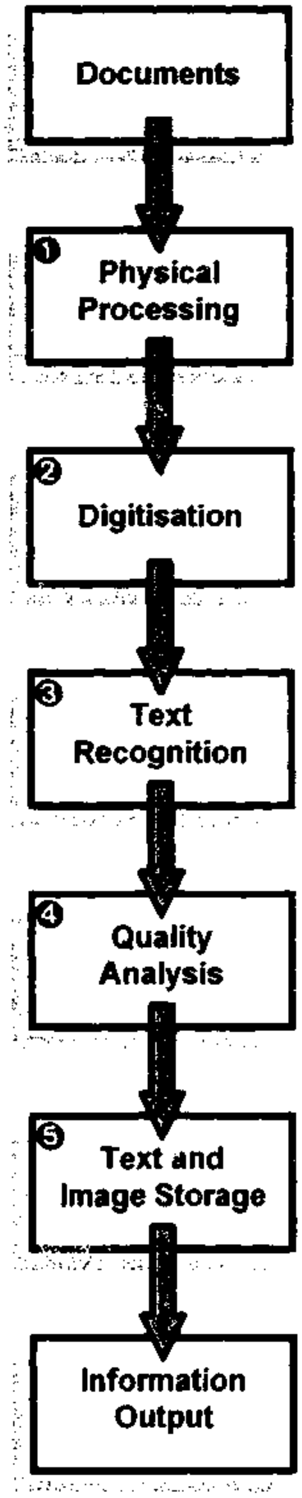


Figure 7.3 High Volume OCR model II process flow diagram
This model includes processes required by the high volume environment, but is still sequential in nature. The documents are passed through the various processing stages before becoming information output.

The processes in the high volume OCR model II are described in order of their process number (shown in Figure 7.3):

❶ Physical Processing. The physical processing includes every process required to transform the documents into a format which is acceptable by the scanner. This includes removing staples, paper clips, bindings and anything else the scanner will not handle and additional physical processing for documents which have not been imaged correctly. Also included is the decision process is the process of sending the documents to the physical data entry section (not shown in Figure 7.3) if the document cannot be transformed into a format acceptable to the scanner. The transfer parameters for this process depend heavily on the physical format of the documents.

❷ Digitisation. This process involves the digitisation of the documents into an image format capable of being processed by the document recognition system. This process also involves any temporary storage of images after image quality analysis and before document recognition, and any permanent storage of images. Where image storage is not required, image disposal is included in this process. Transfer parameters are dependent on paper transport mechanisms, the digitisation process, the storage media and the image file size.

❸ Text Recognition. Text recognition includes the entire process of converting the document image into text. Transfer parameters are dependent on the OCR module being used and the document image quality.

❹ Quality Analysis. This process involves the analysis of the text created by the document recognition process. If the text meets certain conditions indicating sufficient quality, then the text is sent to the text storage process, otherwise the text and image are sent to the image data entry process. The text quality analysis may refer to previously stored text to verify the current text's quality. Transfer parameters depend on knowledge of the text content (e.g. numeric only text) and the methods used for quality analysis.

❺ Text and Image Storage. This process involves the storage of text and images for later use in an electronic database. Where image storage is not required, image disposal is included in this process. The text and image storage may be accessed for quality analysis of following documents. Transfer parameters depend on the file size of the images and the storage media.

The information output stage consists of an electronic database which acts as a container for the documents text and document images. The document data is indexed in the database to facilitate location and extraction of information from the database.

7.3.2.Model II Design

The internal function of each process in the model is illustrated in Figure 7.4. The flowchart of internal process functions describes the individual functions and the order in which they occur. If the answer to a decision box in the flowchart is "yes", then flow proceeds down the page, otherwise flow proceeds to the side.

The internal functions start by asking whether there are any free operators. If there are free operators, then flow proceeds to the documents-in-buffer section. If not, then flow proceeds to the documents-to-process section. If there are documents in the buffer, then flow proceeds to the documents-greater-than-operators section, otherwise flow proceeds again to the documents-to-process section. If the number of documents in the buffer is greater that the number of free operators, then one document is shifted to each free operator, otherwise all the buffer documents are shifted to free operators. Operator-heid documents are then processed. If the operators have finished processing any documents, then these documents are passed to the next external processes buffer, otherwise internal process flow ends. If there are no operators free and/or no documents in the buffer, the question is asked whether there are any documents to process. If there are documents to process, then flow proceeds to where operator-held documents are processed, otherwise internal process flow ends. Once all finished documents are transferred to the next external process buffers, the internal process flow is ended.

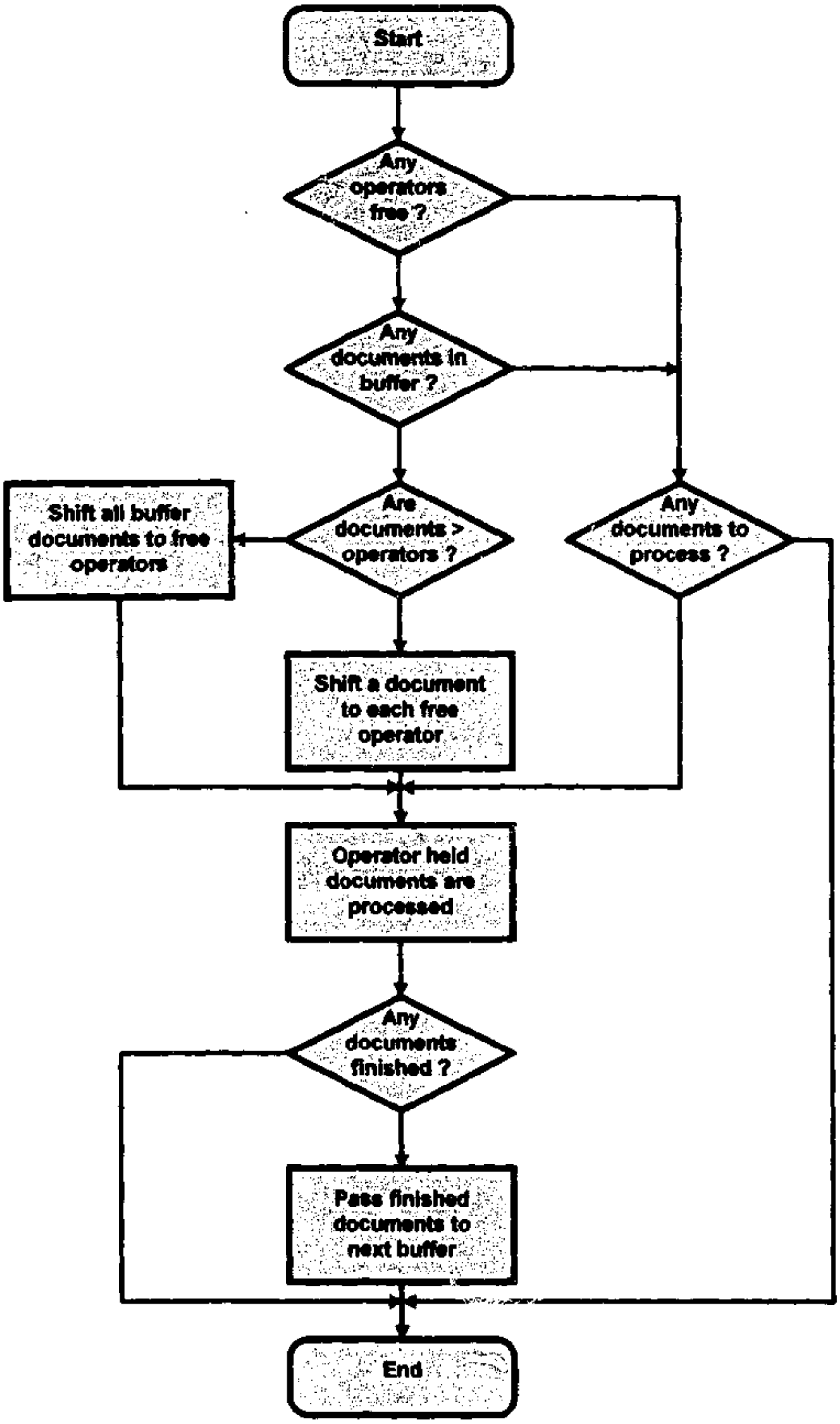

Start
Any operators free ?
Any documents in buffer ?
Are documents > operators ?
Shift all buffer documents to free operators
Any documents to process ?
Shift a document to each free operator
Operator held documents are processed
Any documents finished ?
Pass finished documents to next buffer
End

Figure 7.4 Flow chart of internal process function (model II)
This flow chart illustrates the internal functions which occur at each process stage of the Sequential HV OCR model II. Each process stage, from physical processing down to text & image storage, has the same basic internal functions as illustrated in this flow chart.

To describe the internal process functions mathematically, the following variables are used:

n process number

D_n delay for process n (minutes)

O_n operator numbers for process n (operators)

t time (minutes)

B_{nt} buffer level for process n at time t (documents)

P_{nt} documents started to be processed for process n at time t (documents)

The buffer levels, B_{nt}, can be calculated as shown in Equation 7.1.

$$B_{nt} = B_{n(t-1)} - P_{nt} + P_{(n-1)(t-Dn)} \qquad (7.1)$$

The new buffer level is equal to the old buffer level less the documents commenced processing plus the documents finished processing from the previous process.

The documents commenced processing, P_{nt}, can be calculated as shown in Equation 7.2.

$$P_{nt} = \begin{cases} B_{nt} & \text{for } B_{nt} \leq O_n - \sum\limits_{T=t-1}^{t-D_n-1} P_{nT} \\ O_n - \sum\limits_{T=t-1}^{t-D_n-1} P_{nT} & \text{for } B_{nt} > O_n - \sum\limits_{T=t-1}^{t-D_n-1} P_{nT} \end{cases} \qquad (7.2)$$

The new documents commenced processing is the minimum of the documents in the buffer and the number of free operators. The number of free operators is equal to the operators at that process less the sum of the documents currently being processed. Refer to Appendix F for the source code of the high volume OCR model II.

7.3.3.Model II Analysis

Analysis of the high volume OCR model II showed that the model did not consider certain situations which may occur in the high volume document environment:

- Some documents may not be able to be digitised. The document quality is insufficient and therefore the document data has to be entered by an operator via a keyboard.
- Some documents' images have been digitised incorrectly. The documents' image quality is poor and therefore the documents are returned for further physical processing.
- The recognised text of some documents is incorrect. The document text quality is poor and therefore the document data has to be entered by an operator via a keyboard.

To take these situations into account, the model should allow branching and feedback. This non-sequential model would allow documents to be passed to different processes depending on the document, image and text quality. Further analysis of the sequential HV OCR model II is given in Chapter 8.

7.4. NON-SEQUENTIAL HV OCR MODEL III

The non-sequential high volume OCR model III is the next stage in the evolution of the high volume OCR model II. By adding branching and feedback to the model, the shortcomings reported in the analysis of the previous model are overcome. The model's structure and design are discussed in the following sections and is followed by an analysis of the non-sequential HV OCR model III.

7.4.1. Model III Structure

The structure and process flow in the non-sequential high volume OCR model III is illustrated in Figure 7.5.

For reference purposes and for later analysis, each process in the non-sequential high volume document OCR model III is assigned a process number *n* to identify it. In most cases this number is different from that used in model II, even though the processes may be similar. Each process number n is described by three transfer parameters and a quality pass function:

D_n delay for process n (minutes)

O_n operator numbers for process n (operators)

Q_n quality pass function for process n

The process delay represents the time taken for a document to be operated on by that process and then passed onto the next process as per Section 7.3.1. The operator numbers represent the maximum number of documents which can be operated on concurrently at that process as per Section 7.3.1. The quality pass function determines the direction in which documents are routed. The quality pass function only operates for process numbers one, three and six. All other processes have only direction in which to pass documents and so the quality pass function is not required.

The quality pass function is necessary for modelling the probability of documents not conforming to their ideal form. There is a probability that a particular document will not be in a format suitable for digitisation. There is a probability that a particular document's image will not be in a format suitable for text recognition. There is also a probability that a particular document's text will not have been recognised properly.

These probabilities are represented by quality pass functions in the OCR system model. They are necessary because the process flow of documents in the high volume environment is determined by the quality of the document. The process flow determines the processes a document is subjected to which then effects the performance of the whole OCR system.

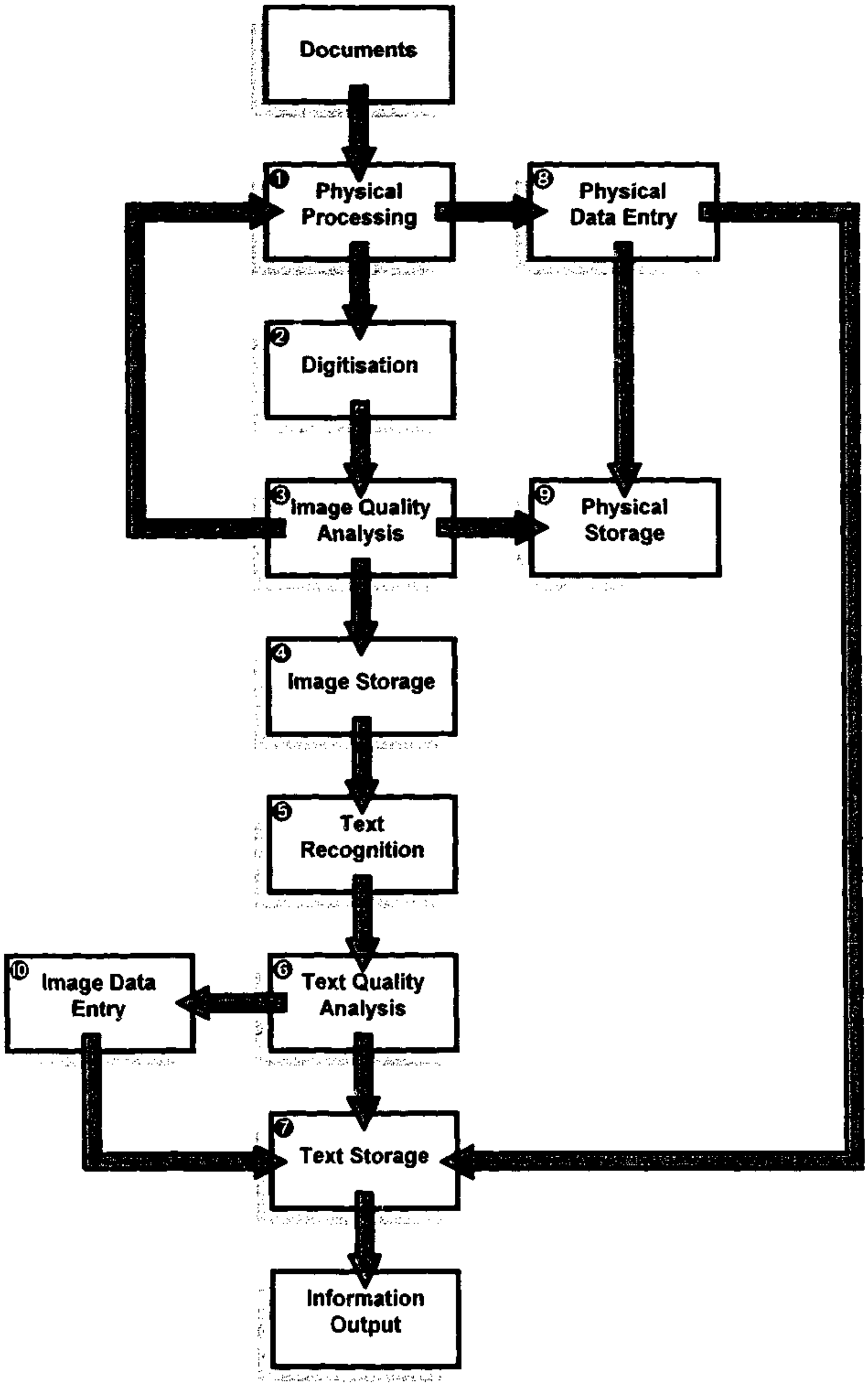


Figure 7.5 Non-sequential HV OCR model III process flow diagram

This model introduces non-sequential flow to the process flow diagram. The documents are passed through different processing stages depending on document quality. There are several paths a document can take as it is transformed from a physical document to information output.

The processes in the non-sequential high volume OCR model III are described in order of their process number:

➊ Physical Processing. The physical processing includes every process required to transform the documents into a format which is acceptable by the scanner. This includes removing staples, paper clips, bindings and anything else the scanner will not handle and addition physical processing for documents that have not been imaged correctly. Also included is the decision and process of sending the documents to the physical data entry section if the document cannot be transformed into a format acceptable to the scanner. The transfer parameters for this process depend heavily on the physical format of the documents.

➋ Digitisation. This process involves the digitisation of the documents into an image format capable of being processed by the document recognition system. Transfer parameters are dependent on paper transport mechanisms and the digitisation process and the image file size.

➌ Image Quality Analysis. Image quality analysis is perfcrmed to ensure the document recognition system can handle the document image. Images which do not pass the quality analysis are sent back to the physical processing stage for corrective measures to be carried out or physical data entry if necessary. Transfer parameters depend on the types of image analysis performed and the portion of images analysed and the image file size.

➍ Image Storage. This involves any temporary storage of images after image quality analysis and before document recognition and any permanent storage of images. Where image storage is not required, image disposal is included in this process. Transfer parameters depend on image file size and storage media.

❺ Text Recognition. Document Recognition includes the entire process of converting the document image into text. Transfer parameters are dependent on the OCR engine being used and the document image quality.

❻ Text Quality Analysis. This process involves the analysis of the text created by the document recognition process. If the text meets certain conditions indicating sufficient quality then the text is sent of the text storage process, otherwise the text and image are sent to the image data entry process. The text quality analysis stage may refer to text previously stored to analyse the current text. Transfer parameters depend on knowledge of the text content and the methods.

❼ Text Storage. Text is stored away for later use in an electronic database. The text storage database may be accessed for text quality analysis of following text entries. Transfer parameters depend on storage media.

❽ Physical Data Entry. Documents which cannot be processed into a digitisable format are sent to a human operator who keys in the data while looking at the physical document. The physical documents are then sent to the document storage process, while the keyed data is sent to the text storage process. The transfer parameters depend on the operator's skill.

❾ Physical Storage. This process involves the storage or disposal of the physical documents. The physical documents may need to be used again if image or text quality is insufficient. Transfer parameters depend on the physical format of the documents.

⑩ Image Data Entry. When the text quality is determined to be insufficient by the text quality analysis process, a human operator keys in the data. This is done while looking at the document image or physical document if the image quality is inadequate. The keyed data is then sent to the text storage process. The transfer parameters depend on the operator's skill.

The information output stage consists of an electronic database which acts as a container for the documents text and document images. The document data is indexed in the database to facilitate location and extraction of information from the database.

7.4.1.1.Quality Analysis

Three of the processes described above require some sort of decision to be made as to where to send the documents which have been processed. They are the physical processing stage 1, the image quality analysis stage 3, and the text quality analysis stage 6. The routing decision is dependent on the quality of the document, image, or text.

The physical processing stage 1 requires a decision to be made as to whether a document should go to the digitising stage or physical data entry stage 8. A quality pass function, Q_1, is used to determine the probability of a document being routed to the scanning stage 2. The quality fail function, 1-Q_1, can then be used to determine the probability of a document being routed to the physical data entry stage.

The image quality analysis stage 3 requires a decision to be made as to whether a document image should go to the image storage stage 4 or for the document to be returned to the physical processing stage 1. A quality pass function, Q_3, is used to quantify the probability of a document being routed to the image storage stage. The quality fail function, 1-Q_3, can then be used to determine probability of a document being routed back to the physical processing stage.

The text quality analysis stage 6 requires a decision to be made as to whether the document text should go the text storage stage 7 or for the document image to go to the image data stage 10. A quality pass function, Q_6, is used to quantify the probability of a document being routed to the text storage stage. The quality fail function, 1-Q_6, can then be used to determine the probability of a document being routed to the image data entry stage.

7.4.1.2.Document Transformation

As the documents are transformed into text, they go through three distinct stages. In the first stage, the document is in its physical form. In the second stage the document is in its image form. In the third stage the document is in its text form. Figure 7.6 shows the process flow from documents to information output but does not show what stage the document is in. Figure 7.6 however, shows which stage the document is in as it is transformed from a physical document to document images to document text. As can be seen from Figure 7.6, sometimes both the physical document and document image are transferred between processes, and sometimes both the document image and document text are transferred between stages. The dark grey arrows show the process flow of physical documents. The white arrows show the process flow of the document images. The light grey arrows show the process flow of the document text.

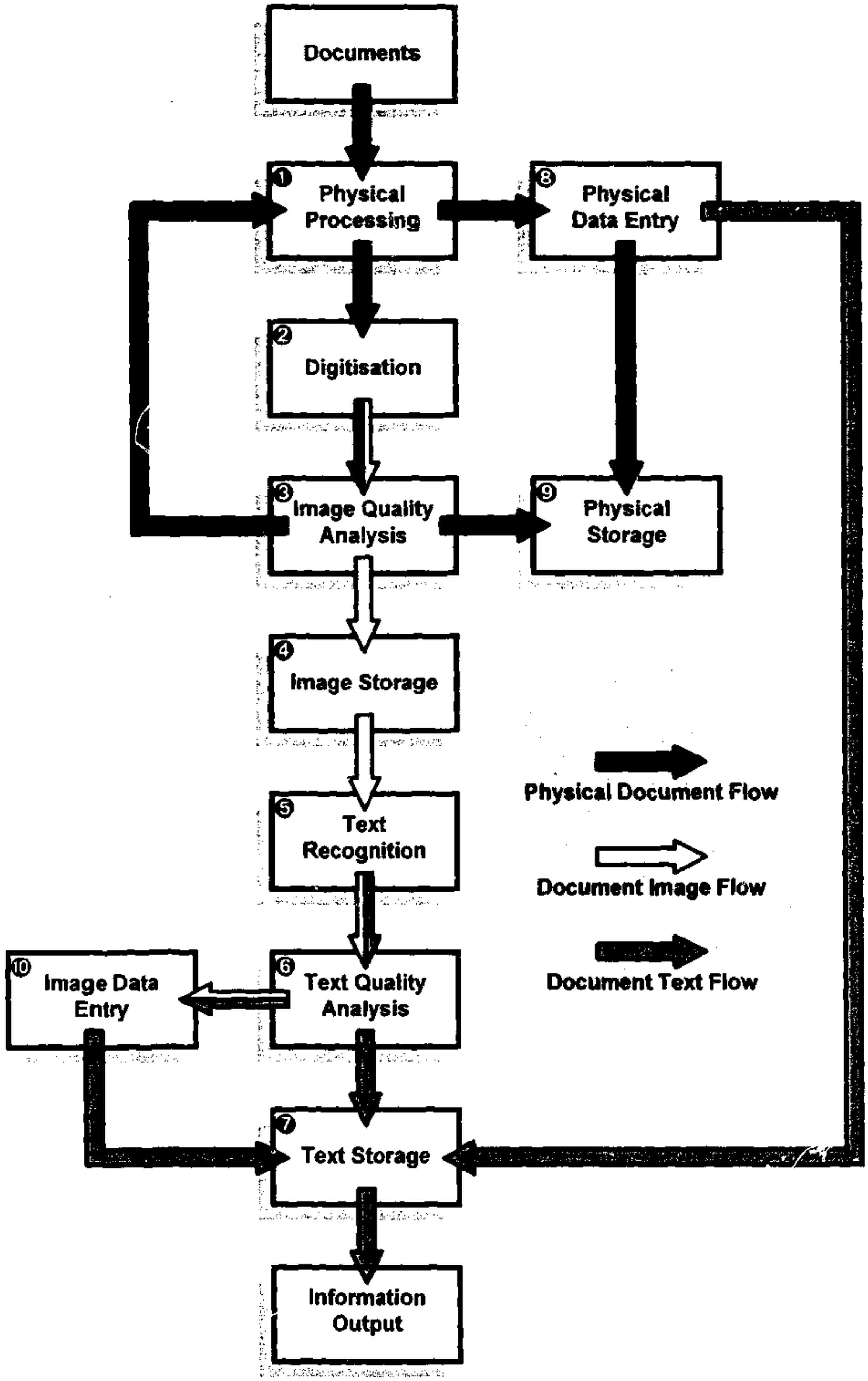


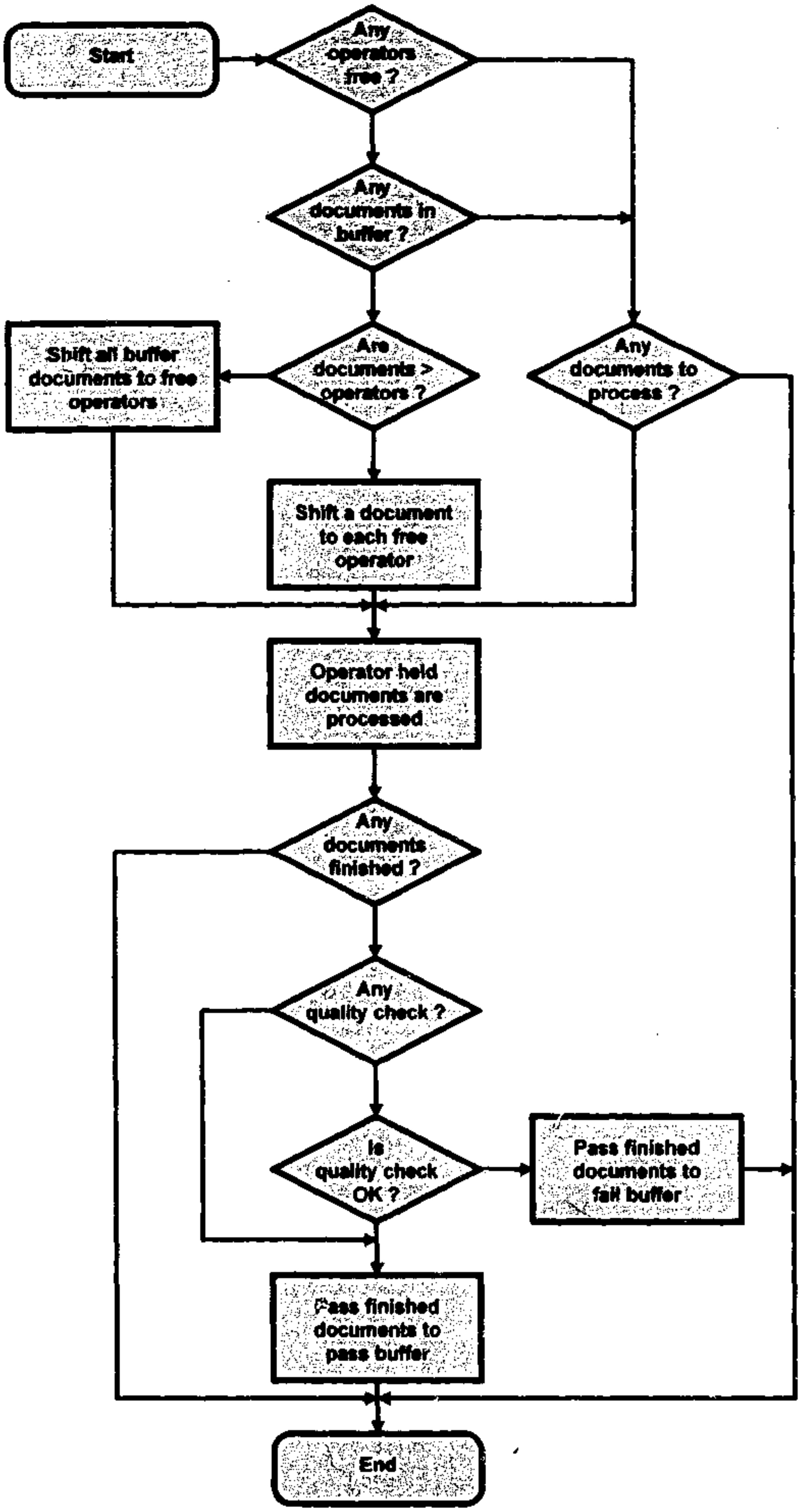
Figure 7.6 Model III document/image/text flow
Previous process flow diagrams show only the direction of process flow. This flow diagram also shows the different states of the document as it is transformed from a physical document to information output.

7.4.2.Model III Design

The internal function of each process in the model is illustrated in Figure 7.7. The flowchart of internal process functions describes the individual functions and the order in which they occur. The flowchart is similar to that for model II with most of the differences being in the lower part of the flowchart.

The description of this flowchart will proceed from the section after operator held documents are processed. Refer to section 7.3.2 on model II design for a description of the previous sections of the flowchart.

If any documents are finished being processed by the operators, then flow proceeds to the quality section, otherwise the process flow ends. If there is a quality check for this process, then flow proceeds to that quality check, otherwise the quality check is bypassed and all finished documents are transferred to the pass buffer. If the quality check for the documents is successful, then the finished documents are transferred to the pass buffer, otherwise the finished documents are transferred to the fail buffer. Once all finished documents are transferred to the appropriate processes buffers, the internal process flow is ended.



Start
Any operators free ?
Any documents in buffer ?
Are documents > operators ?
Shift all buffer documents to free operators
Any documents to process ?
Shift a document to each free operator
Operator held documents are processed
Any documents finished ?
Any quality check ?
Is quality check OK ?
Pass finished documents to fail buffer
Pass finished documents to pass buffer
End

Figure 7.7 Flow chart of internal process operation (Model III)

This flow chart illustrates the internal functions which occur at each process stage of the Non-Sequential HV OCR model III. Each process stage, from physical processing down to text & image storage, has the same basic internal functions as illustrated in this flow chart.

To describe the internal process functions mathematically, the following variables are used:

a_n number of pass inputs at process *n*

b_n number of fail inputs at process *n*

X_{ni} pass rate for buffer input *i* at process *n*

Y_{ni} fail rate for buffer input *i* at process *n*

The buffer levels, B_{nt}, can be calculated as shown in Equation 7.2.

$$B_{nt} = B_{n(t-1)} - P_{nt} + \left\{ \sum_{x=1}^{a_n} X_{xi} P_{x(t-Dx)} \right\} + \left\{ \sum_{y=1}^{b_n} Y_{yi} P_{y(t-Dy)} \right\} \qquad (7.3)$$

The new buffer level is equal to the old buffer level less the documents commenced processing plus the documents finished processing and passed from the proceeding passing processes plus the documents finished, processing, and failed from the proceeding failing processes.

The calculation for the documents which have commenced processing, P_{nt}, is the same as that for model II. Refer to Equation 7.2. Refer to Appendix F for the source code of the non-sequential HV OCR model III.

7.4.3.Model III Analysis

From an analysis of the idle or free operators, it can be seen that at certain times and at certain processes there are significant numbers of operators with no documents to process. Model performance could be improved if these idle operators were shifted to processes where there were insufficient operators to cope with buffered documents. A mechanism which would allow these operators to move between processes would provide the model with a self-optimising feature.

Further analysis of the non-sequential HV OCR model III is given in Chapter 8.

7.5.Self Optimising HV OCR Model IV

The self optimising high volume OCR model IV represents another evolutionary step forward in the development of the model. By implementing improvements indicated by analysis of the previous model's performance, a new model which can control the number of operators at each stage is created. The model's structure and design are discussed in the following sections, followed by an analysis of the model.

7.5.1.Model IV Structure

The structure and process flow in the self optimising HV OCR model IV is illustrated in Figure 7.8.

The processes in the self optimising HV OCR model IV are the same as for the non-sequential HV OCR model III (Refer to Section 7.4.1 on model III structure). The difference between model III and model IV is the self optimising aspect of model IV. The self optimising aspect refers to the moving of operators around the system to deal with work loads at each process. It is used to improve the performance of the model.

By examining the buffer levels and the number of idle operators at each process, it can be seen that model performance can be improved if operators were shifted from processes with low buffer levels to processes with high buffer levels. This operator shifting mechanism uses an operator pool. Processes with zero buffer levels release operators to the pool while processes with high buffer levels request operators from the pool. Factors such as the time taken for operators to move to and from the pool, and the number of operators requested from the pool are considered. Limitations such as the maximum and minimum operators able to work at particular processes are also taken into account.

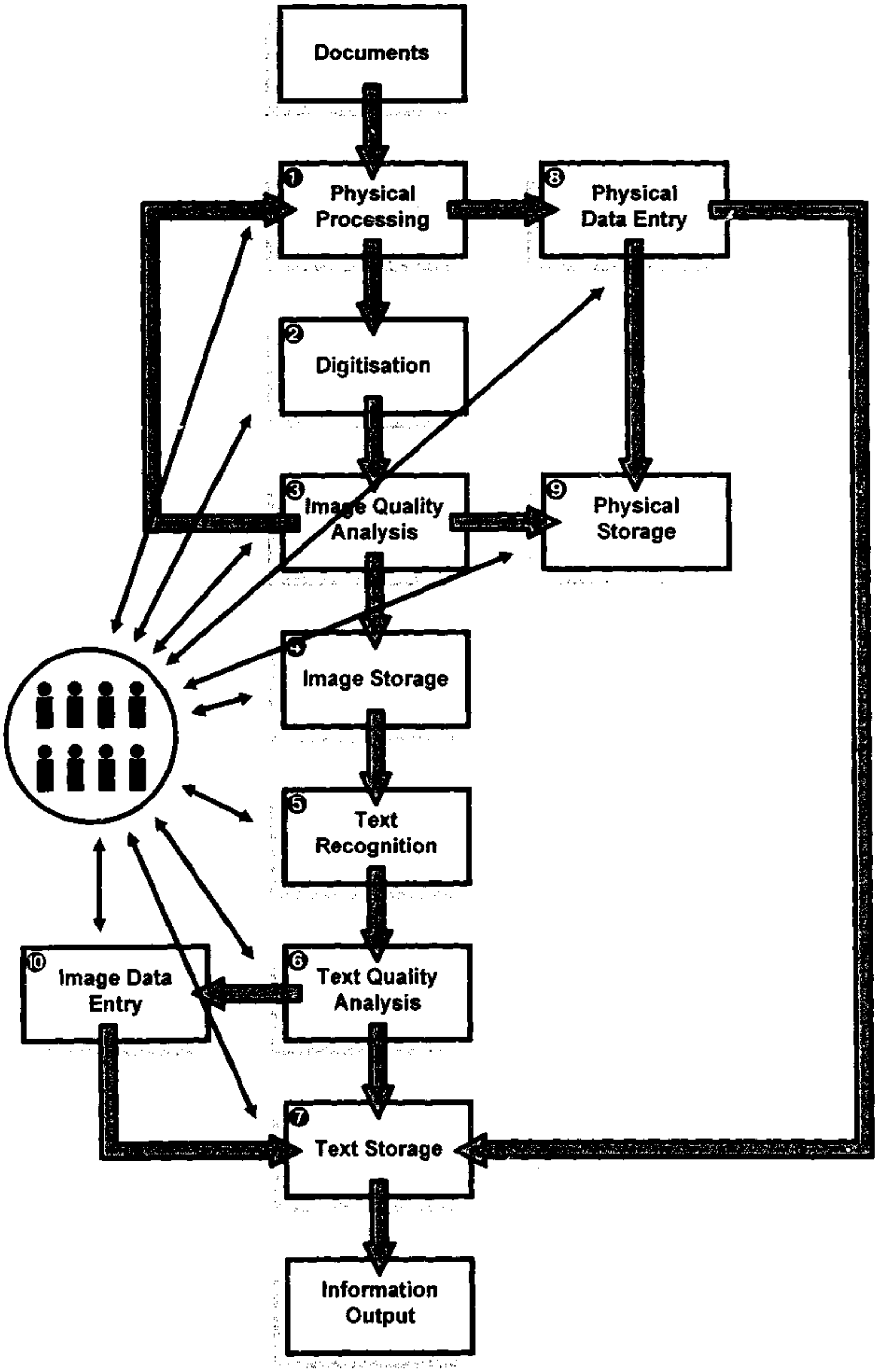


Figure 7.8 Self optimising HV OCR model IV process flow diagram

This model introduces an operator optimising feature to the process flow diagram. Operators can be released from processes that have excess operators to the operator pool. Operators can also be requested from the operator pool by processes with insufficient operators.

7.5.2.Model IV Design

The flowchart of the internal function of each process in the model is the same as that for model III. Refer to Section 7.4.2 on model III design for a description of the internal functions of each process. In addition to the flowchart of internal functions, a second flowchart is used to illustrate the internal optimising proces^ses in the model. The flowchart of internal optimising processes, shown in Figure 7.9 describes the individual optimising functions and the order in which they occur.

The optimising process checks to see if it is time for an operator check. The operator check is done periodically, not every minute, and the length of the period can be altered. This delay reflects the time taken for an operator to move to and from the operator pool. If it is time for an operator check, the number of idle operators are measured, otherwise the process ends. If there are no idle operators then process proceeds to see if more operators can be requested, otherwise operators may be released to the operator pool.

Before more operators can be requested for the process, a check is done to see if the current number of operators is greater than the maximum number that process can accept. Another check is also done to see if any operators are available in the operator pool. If both checks pass, then additional operators are requested. If either check fails, then no operators are requested and the process ends.

If there are some idle operators, a check is done to see if there is sufficient idle operators to consider releasing an operator to the operator pool. If there are many idle operators, a check is done to see if the number of operators at the process is more than the minimum required. If the minimum operator requirements are still met, then an operator is released to the operator pool. If there are not enough idle operators, or the number of operators for that process is at its minimum, then no operators are released to the operator pool and the process is ended.

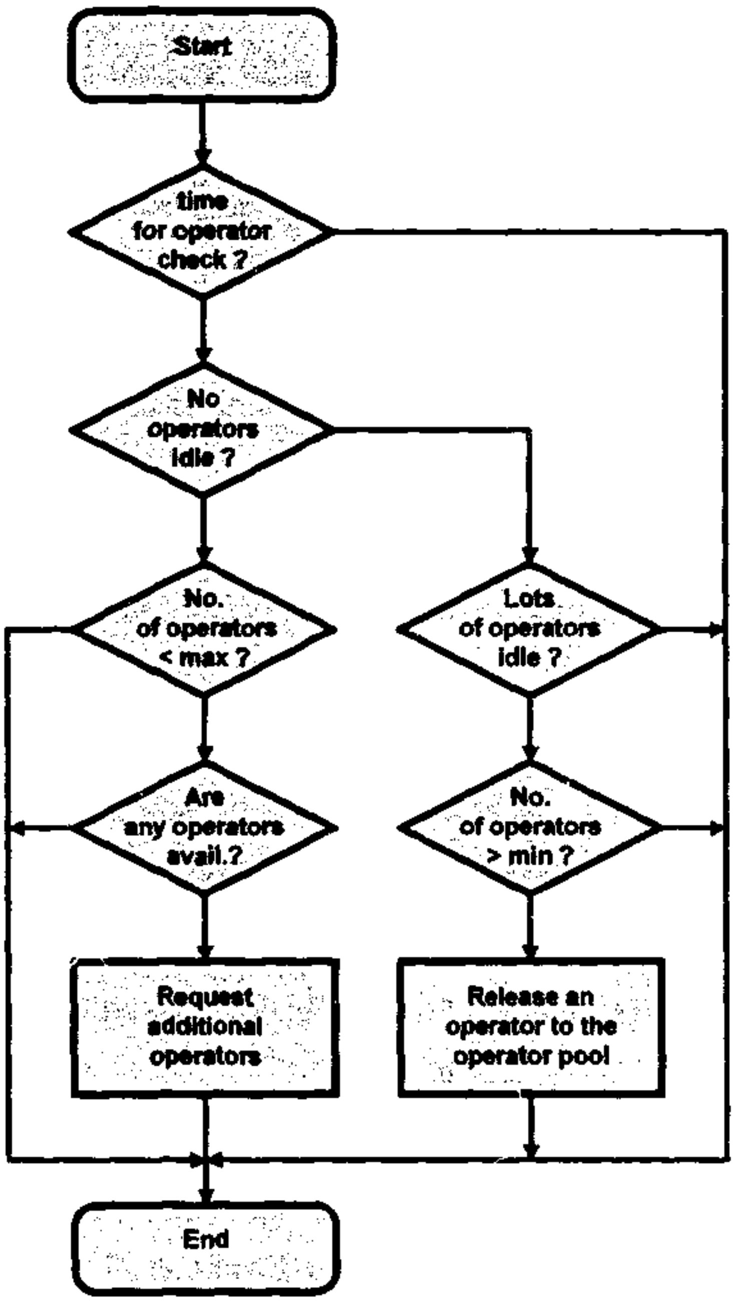

Start
time for operator check ?
No operators idle ?
No. of operators < max ?
Lots of operators idle ?
Are any operators avail.?
No. of operators > min ?
Request additional operators
Release an operator to the operator pool
End

Figure 7.9 Flow chart of internal optimising functions (model IV)
This flow chart illustrates the internal operator optimising functions that occur at each processing stage of the Self-Optimising HV OCR Model IV. Each process stage has the same basic internal operator optimising functions as illustrated in this flow chart.

The mathematical description of the internal process functions is the same as that for model III. Refer to Section 7.4.2 on model III design. For the source code of the self optimising HV OCR model IV, refer to Appendix F.

7.5.3.Model IV Analysis

An analysis of the self optimising HV OCR model IV's performance indicates that while it overcomes the previous model's shortcomings, there are still areas in which it can be improved. It does not take into account the real world constraints which would be imposed on the system it emulates. Constraints such as operator work durations, operator idle time costs and total processing time should be taken into account in the model so that the model can be applied to real world systems. Further analysis of the self optimising HV OCR model IV is given in Chapter 8.

7.6. Real World Constraint HV OCR Model V

The real world constraint HV OCR model V represents the final evolutionary step in the development of the OCR system model. By implementing improvements indicated by analysis of the previous models performance, a new model is created which is constrained by real world parameters such as operator work durations, processing time limits and economic considerations. The model's structure and design are discussed in the following sections, followed by an analysis of the model.

7.6.1.Model V Structure

The structure and process flow in the real world constraint HV OCR model V is illustrated in Figure 7.10. The links between the processes and the operator pool still exist, but the arrows have been removed for clarity (cf. Figure 7.9).

The processes in the real world constraint HV OCR model V are the same as those for model IV (refer to Section 7.5.1). The difference between model V and model IV is the addition of real world constraints and the integration of the OSPO OCR system performance optimiser described in Chapter 6.

As shown in Figure 7.10, OSPO monitors the text recognition accuracy (from process 6), the text recognition speed (from process 5) and the text size (from process 5). OSPO can then dynamically change the image digitisation resolution to optimise the text recognition accuracy and speed. The structure of the OSPO system is examined in Section 6.2. Data is sent between OSPO and Excel using the direct data exchange facilities of MS Windows.

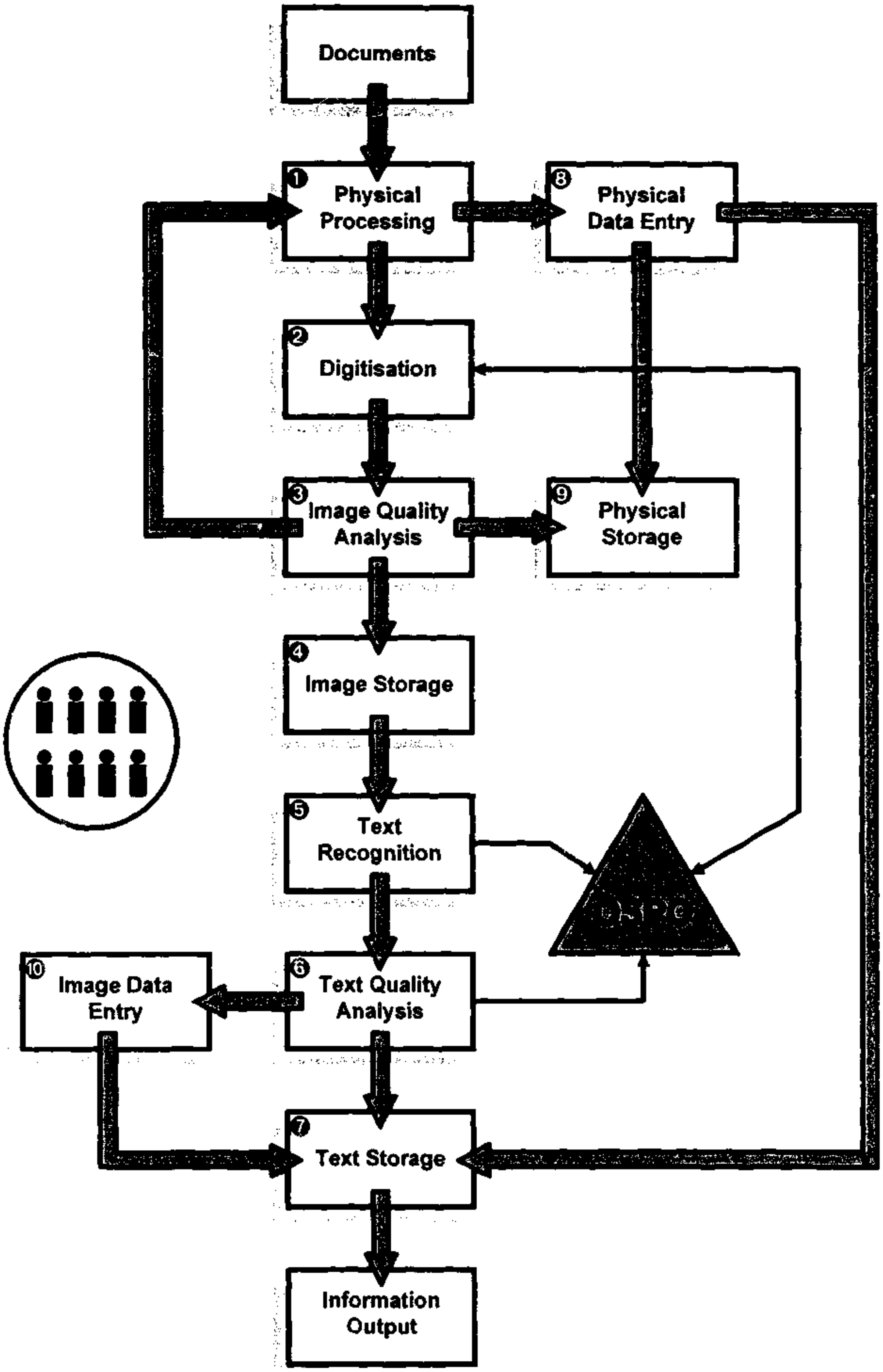


Figure 7.10 Real world constraint HV OCR model V process flow diagram
This process flow diagram shows the introduction of real world constraints and the OSPO system to the OCR system model. Performance data is fed to the OSPO system which changes the digitisation parameters to optimise text recognition.

7.6.2.Model V Design

The flowchart of the internal function of each process in model V is the same as that for model IV. Refer to Section 7.5.2 on model IV design for a description of the internal functions of each process.

Real world costs such as operator wages and operator idle time are taken into account by the new model. The model's time duration has been extended to represent a standard eight hour shift.

The total processing time is also computed by the new model. Two output levels are used to compute the total processing time. The output from the physical storage process is defined as the documents-physically-stored output level. The output from the text storage stage (shown as "Information Output" in Figure 7.10) is defined as the documents-stored-as-text output level. For documents to be completely processed, they must reach both output levels, i.e. documents must be stored physically and as text. The total processing time is therefore the time taken for all documents in the system to reach both output levels.

The operator idle cost is calculated by summing the number of idle operators for each minute during the eight hour shift and then multiplying the total by the operator per minute pay rate.

The mathematical description of the internal process functions is the same as that for model IV. Refer to Section 7.5.2 on model IV design.

Refer to Appendix F for the source code of the real world constraint HV OCR model V.

7.6.3.Model V Analysis

The analysis of the real world constraint HV OCR model V's performance is examined in detail in Chapter 8. Chapter 8 describes the experimentation with the OCR system model and analyses the experimental results. Experiments using the model examine the process buffers and output levels (Section 8.2), the number of idle operators (Section 8.3) and the number of assigned operators (Section 8.4) during the shift. Sensitivity analysis of the model is reported in Section 8.5 which examines the effects on operator idle cost and total processing time. A summary of the analysis of the OCR system model is given in Section 8.6.

7.7. SUMMARY OF MODEL DEVELOPMENT

This section summarises the development of the OCR system model. The development of the OCR system model is shown to evolve from the initial generic OCR model to a high volume OCR model. The next evolutionary stage is the non-sequential, high volume OCR model. A new feature is introduced to produce the next version which is the self optimising, high volume OCR model. Model development then progresses to the final, real world, high volume OCR model. The development process is visualised in Figure 7.1.

The final evolutionary version of the OCR system model is the "Real world constraint, high volume, OCR model V." This sophisticated model can be used as an analytical tool for conducting research into a wide range of digital image document processing areas, examples of which are described in Chapter 8. It can also be used to improve the performance of existing digital image document processing systems and in the design of new systems.

While the origins of the OCR system model can be found in the generic OCR model reported in the literature in Chapter 2, the final evolution of the model is unique in the high volume document processing environment. The OCR system model reported in this chapter is therefore original and represents a significant advancement in the knowledge of digital image document processing.

CHAPTER 8

MODEL EXPERIMENTATION AND ANALYSIS

CHAPTER CONTENTS PAGE

8. MODEL EXPERIMENTATION AND ANALYSIS

This chapter examines the OCR system model experimentation and analyses the results obtained from the model experimentation. The OCR system model and its development are described in Chapter 7. An introduction to the experimentation and analysis is given, followed by the experimental results and analysis. The first experimental results describe the OCR system model in terms of its process buffers and output levels. The next results examine the OCR system model's efficiency by studying the numbers of idle operators. A complimentary study is also carried out on the number of operators assigned to particular processes. Further analysis is conducted which examines the sensitivity of idle operator cost and processing time to variations in physical document quality, image quality and recognised text quality. A summary of the analysis is then presented which concludes the experimentation and analysis of the OCR system model.

8.1. INTRODUCTION

The purpose of the OCR system model experimentation and analysis work was to investigate the effects of different operating conditions on a high volume digital image document processing system. The system which the OCR model emulates is based loosely on the Australian Securities Commission (ASC) National Information Processing Centre's document processing system. Observations of the ASC system's performance were translated into parameters suitable for use with the OCR system model. Although certain aspects of the ASC system did not directly correspond to processes in the OCR system model, extensive experience with the ASC system enabled transposing of those ASC system aspects into appropriate OCR system model processes.

Three separate experiments were devised to show the effects of different document loads on the OCR system model. Each of these experiments examine the process buffer levels, idle operator levels and assigned operator levels. Rather than presenting an examination of the individual experiments categorised by experiment number, a different approach was taken. The reported results are categorised by process buffer level (Section 8.2), idle operator level (Section 8.3) and assigned operator level (Section 8.4). This was done so that the effects of the different document loads on these levels could more easily be compared. The response of the model to different operating conditions can then be determined.

8.2. ANALYSIS OF PROCESS BUFFERS

This section analyses the experimental results involving the OCR system model's process buffers and output buffers. As described in Chapter 7, the process buffers are interim storage areas for documents prior to that process operating on the documents. The process buffers are effectively in-trays for the processes. There are 10 process buffers and two output buffers (documents physically stored and documents stored as text) in the OCR system model.

Three experiments are conducted, each with slightly different parameters. The settings of the experimental parameters are listed in Appendix F. The process buffer and output level results of each experiment are examined in Section 8.2.1, Section 8.2.2, and Section 8.2.3.

8.2.1. Experiment One

For experiment one, all the process buffers begin the eight hour shift empty except for the physical processing buffer which starts with 2000 documents. This represents standard operating conditions which are based on observed values from the Australian Securities Commission's high volume document processing system [89]. Table 8.1 lists the pertinent OCR system model parameters which this experiment is concerned with. These parameters and others associated with experiment one are listed in Appendix F.

Table 8.1 Experiment one OCR system model parameters

This table lists the pertinent parameters of the OCR system model used for experiment one. The other parameters can be found in Appendix F.

[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
1	5	15	2000	4	30
2	2	4	0	4	20
3	3	8	0	4	20
4	4	6	0	3	20
5	2	6	0	5	20
6	5	15	0	4	20
7	2	4	0	4	20
8	6	8	0	4	20
9	4	6	0	3	20
10	2	6	0	4	20

This experiment is designed to show the response of the OCR system model to a single input of 2000 documents. The process buffer and output level results of experiment one are shown in Figure 8.1.

There are ten processes in the OCR system model. In order, these are; physical processing, digitisation, image quality analysis, image storage, text recognition, text quality analysis, text storage, physical data entry, physical storage and image data entry. There are also two output levels; documents stored as text (also referred to as information output) and documents physically stored. These processes and output levels are shown in Figure 7.5.

Not all processes in Figure 8.1 and the following figures are labeled, since 10 or 12 plots on the same chart would obscure the more significant processes. The unlabeled processes are still plotted because they indicate background activity levels.

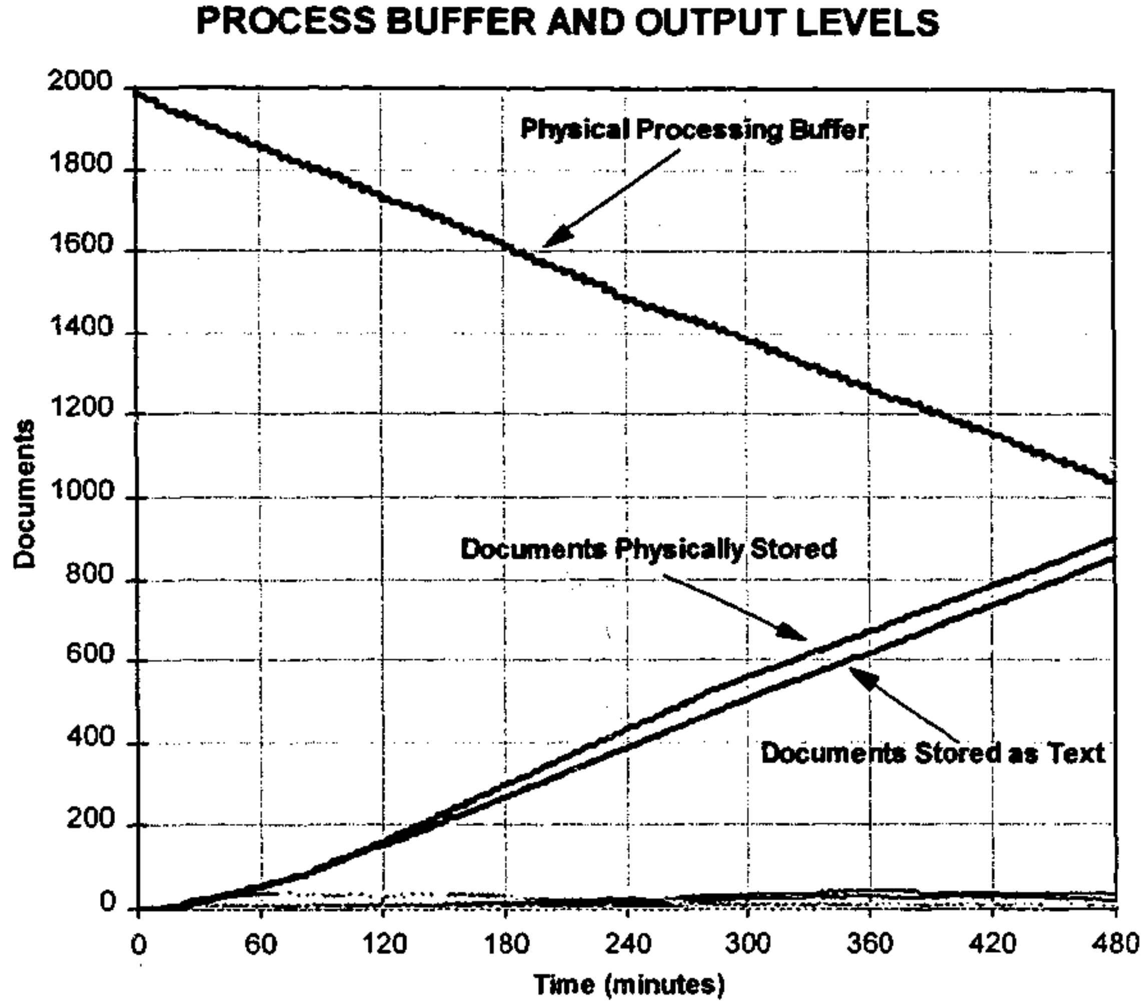


Figure 8.1 Experiment one process buffer and output levels
This chart shows the number of documents in each process buffer and output level during the eight hour work shift using the OCR system model. Significant process buffers and output levels are labeled while others are unlabeled to show background activity levels only.

Several general observations are made from Figure 8.1 concerning the process buffer and output levels for experiment one:

- The gradients of the documents-physically-stored and the document-stored-as text, between the times of 120 and 480 minutes, is approximately equal to the negative of the gradient of the physical-processing-buffer.

- The line representing the physical processing buffer level is more jagged that those representing the documents-physically-stored level and the document-stored-as text level.

- The buffer levels of the other processes do not rise above 50 documents at any time.
- The documents-physically-stored level and the document-stored-as text level do not both reach the level of the total number of documents in the system (2000 documents in this experiment). Thus the eight hour (480 minute) shift is not long enough to complete the processing of all the documents.

Analysis of the results of experiment one has led to explanations for the general observations. Some of these explanations and general observations also apply to later experiments.

The buffer levels for processes other than the physical-processing-buffer stay relatively small in this experiment. It is apparent that documents are not being held up for long by any of the processes. After an initial period during which the model optimises itself, the rate at which the documents are being output (the gradient of documents-physically-stored and documents-stored as text) is therefore equal to the rate at which they are being input (the inverse gradient of the physical processing buffer).

The jaggedness in the line representing the physical processing buffer (and some of the other buffers) is caused by the manner in which operators start to process documents. If many operators simultaneously take a document from a processes buffer, then that processes buffer level will show a sharp drop. When those operators finish those documents and pass them on to the next process, the next processes buffer level will show a sharp increase. As documents reach the later processes (and output stages) of the model, some will have failed quality checks and be diverted to other processes. The later processes therefore get documents on a less regular basis. This is reflected in their generally smoother buffer level lines.

8.2.2.Experiment Two

For experiment two, the pertinent OCR system model parameters are the same as those listed in Table 8.1, except that the text recognition buffer starts the eight hour shift with 500 documents also. These parameters and others associated with experiment two are listed in Appendix F.

The addition of 500 documents part way through the OCR system model represents the effects of unfinished documents from previous shifts. This occurrence is common within the ASC's document processing system [89].

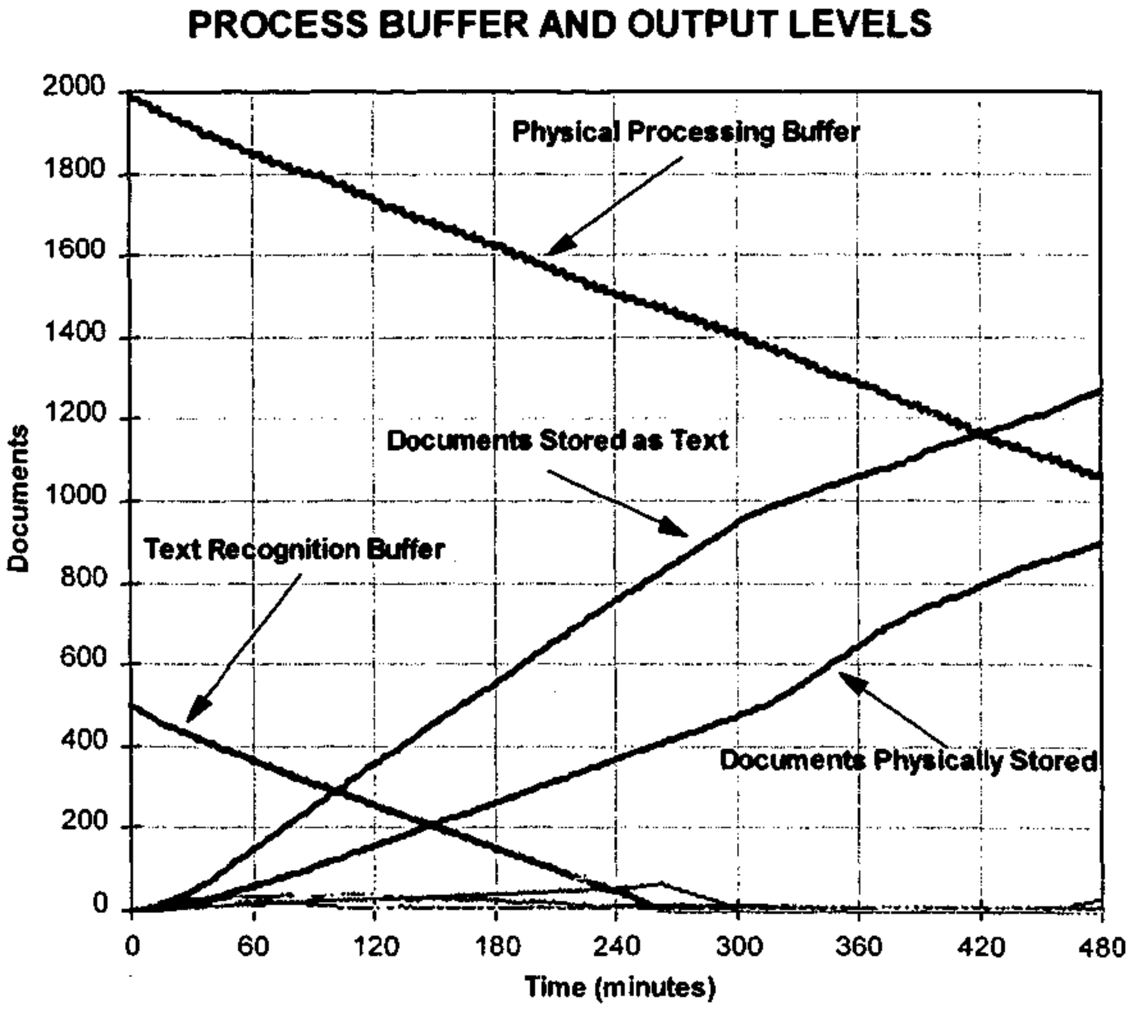


Figure 8.2 Experiment two process buffer and output levels
This chart shows the number of documents in each process buffer and output level during the eight hour work shift using the OCR system model. Significant process buffers and output levels are labeled while others are unlabeled to show background activity levels only.

Several general observations are made from Figure 8.2 concerning the process buffer and output levels for experiment two, some of which are covered by Section 8.2.1:

- After the text recognition buffer level reaches zero, several of the other smaller buffers levels that have slowly risen begin to rapidly drop to zero also. There is a delay between the peaks of these smaller buffers.
- The gradients of the two output levels are different at the start of the shift, whereas in experiment one they are the same.
- After the text recognition buffer level reaches zero, there is a change in gradient of the two output levels; documents stored as text and documents physically stored. The gradients become similar.

Analysis of the results of experiment two has led to explanations for the general observations. Some of these explanations and general observations also apply to later experiments.

Once a process buffer reaches zero, the OCR system model detects the idle operators, and sends them to the operator pool. From the operator pool these operators are then assigned to other processes with large buffers. Hence the buffer level gradient of the processes that get more operators changes. A buffer level reaching zero is therefore followed by gradient changes in other buffer levels.

The gradient of the documents-stored-as-text output level is initially higher that the gradient of the documents-physically-stored output level. This is because documents from both the large physical processing buffer and text recognition buffers make their way through the model to the documents-stored-as-text output level. The documents-physically-stored level only gets documents from the physical processing buffer. Refer to Figure 7.5 which shows the flow of documents through the OCR model. Once the Text recognition buffer is emptied, the gradient of the documents-stored-as-text output level falls to match that of the documents-physically-stored output level.

8.2.3.Experiment Three

For experiment three, the pertinent OCR system model parameters are the same as those listed in Table 8.1, except that the physical processing buffer starts the eight hour shift with only 800 documents. These parameters and others associated with experiment three are listed in Appendix F.

The reduction of initial documents to 800 is done so that the effects of processes halting as no further documents are available can be seen. This occurrence is not common within the ASC's document processing system [89], since their commercial orientation precludes the inefficiencies of such operating conditions. The use of the OCR system model can therefore determine the possible effects of such initial conditions where it is not viable to actually measure those effects.

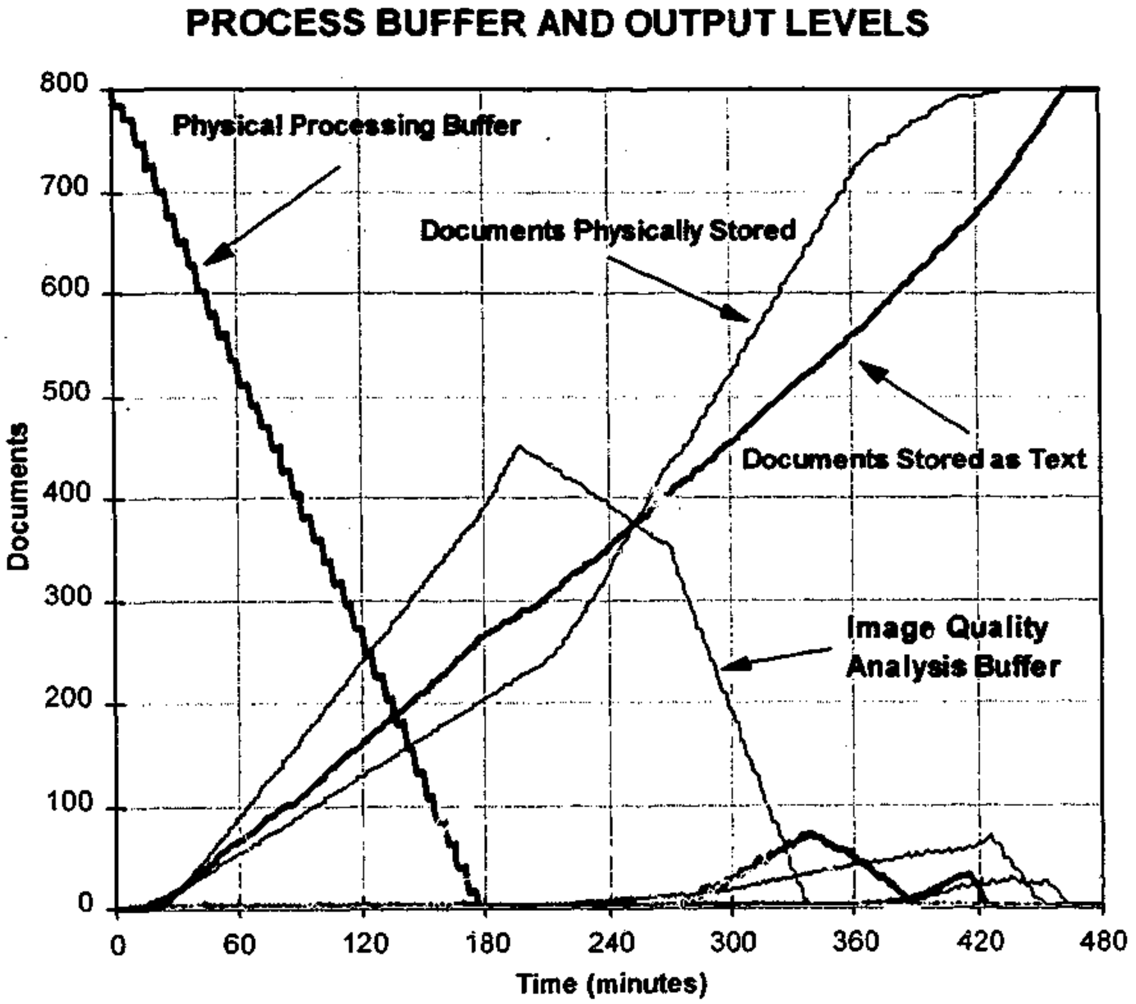


Figure 8.3 Experiment three process buffer and output levels
This chart shows the number of documents in each process buffer and output level during the eight hour work shift using the OCR system model. Significant process buffers and output levels are labeled while others are unlabeled to show background activity levels only.

General observations are made from Figure 8.3 concerning the process buffer and output levels for experiment three, some of which are covered by Section 8.2.1 and Section 8.2.2:

- This experiment differs mainly from the other two in that all the documents pass through the system. All the documents become physically stored and stored as text.
- The changes in gradient are more pronounced than those in experiment one and experiment two.
- After the physical processing buffer initially reaches zero, it begins to climb and fall in two small peaks.

The analysis of the results of experiment three serve mainly to reinforce those for experiment one and experiment two.

After a buffer level reaches zero, there is a pronounced change in gradient in following processes. In the case of the image quality analysis buffer level, there are two distinct drops in the gradient. The first drop in the image-quality-analysis buffer level gradient (at 200 minutes) is due to the physical processing buffer being emptied. The second drop in gradient (at 280 minutes) is due to a sudden increase in operators assigned to the image quality analysis process (refer to Figure 8.9).

The increase in the physical processing buffer after initially reaching zero can be attributed to an increasing number of document which fail the image quality analysis (from the large image quality analysis buffer level) and are returned for extra physical processing.

8.3.ANALYSIS OF IDLE OPERATORS

This section analyses the experimental results involving the OCR system model's idle operators. As described in Chapter 7, the idle operators are those operators who are waiting for documents to be passed to their process. There are 10 processes in the OCR system model, each with an idle operator level.

Three different experiments are conducted using the same parameters as those used in Section 8.2. The settings of the experimental parameters are listed in Appendix F. The idle operator level results of each experiment are examined in Section 8.3.1, Section 8.3.2 and Section 8.3.3.

8.3.1.Experiment One

The experiment one results for the idle operators are taken at the same time as the process buffer and output level results described in Section 8.2.1. Table 8.1 lists the pertinent OCR system model parameters which this experiment is concerned with. These parameters and others associated with experiment one are listed in Appendix F. The idle operator results for experiment one are shown in Figure 8.4.

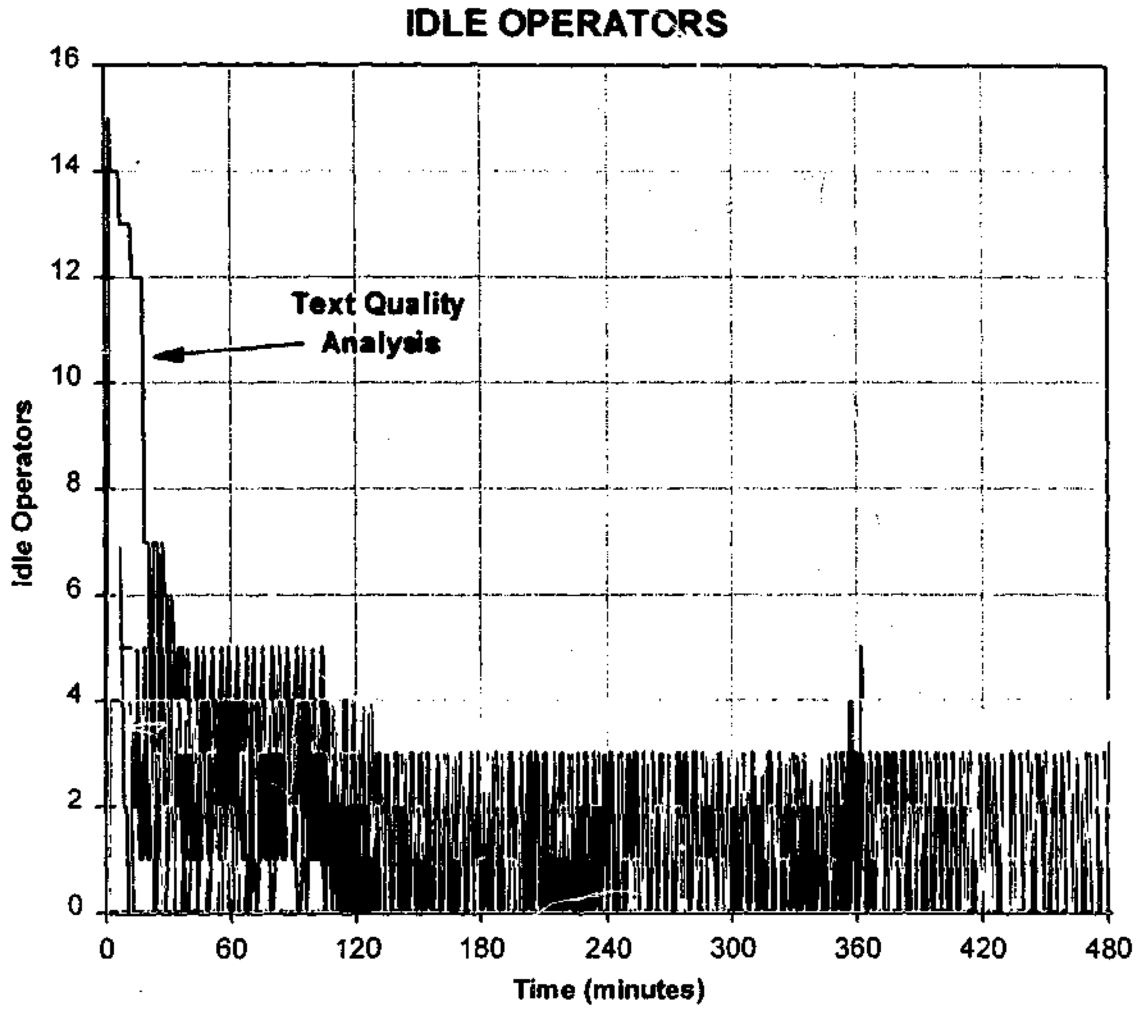


Figure 8.4 Experiment one idle operators
This chart shows the number of idle operators at each process during the eight hour work shift using the OCR system model. Significant idle operator processes are labeled while others are unlabelled to show background activity levels only.

Two general observations are made from Figure 8.1 concerning the idle operator levels for experiment one:

- The text-quality-analysis process's idle operator level initial starts at 15 operators then drops to 5 operators and later to 3 operators.
- The idle operator levels change frequently between 0 and 3 or 4 operators per process.

Analysis of the results of this experiment has led to explanations for these two observations, some of which also apply to later experiments.

The text-quality-analysis processes' idle operator level starts high because there is a high initial operator level (refer to Table 8.1) and because no documents have yet been passed down to that process. The physical processing process also starts with a high initial operator level, but because the buffer is never emptied in this experiment, there are no idle operators.

The idle operator levels can change frequently between two values because some processes have shorter processing times. Table 8.1 shows that process number 4 has a processing time of 4 minutes while process number 5 has a processing time of 2 minutes. Process 5 operators may therefore have to idle for 2 minutes until process 4 operators can complete more documents.

The values which the idle operator levels tend to oscillate between is determined by the minimum operator setting

8.3.2.Experiment Two

The experiment two results for the idle operators are taken at the same time as the process buffer and output level results described in Section 8.2.2. The pertinent OCR system model parameters which this experiment is concerned with are listed in Section 8.2.2. These parameters and others associated with experiment two are listed in Appendix F. The idle operator results for experiment two are shown in Figure 8.5.

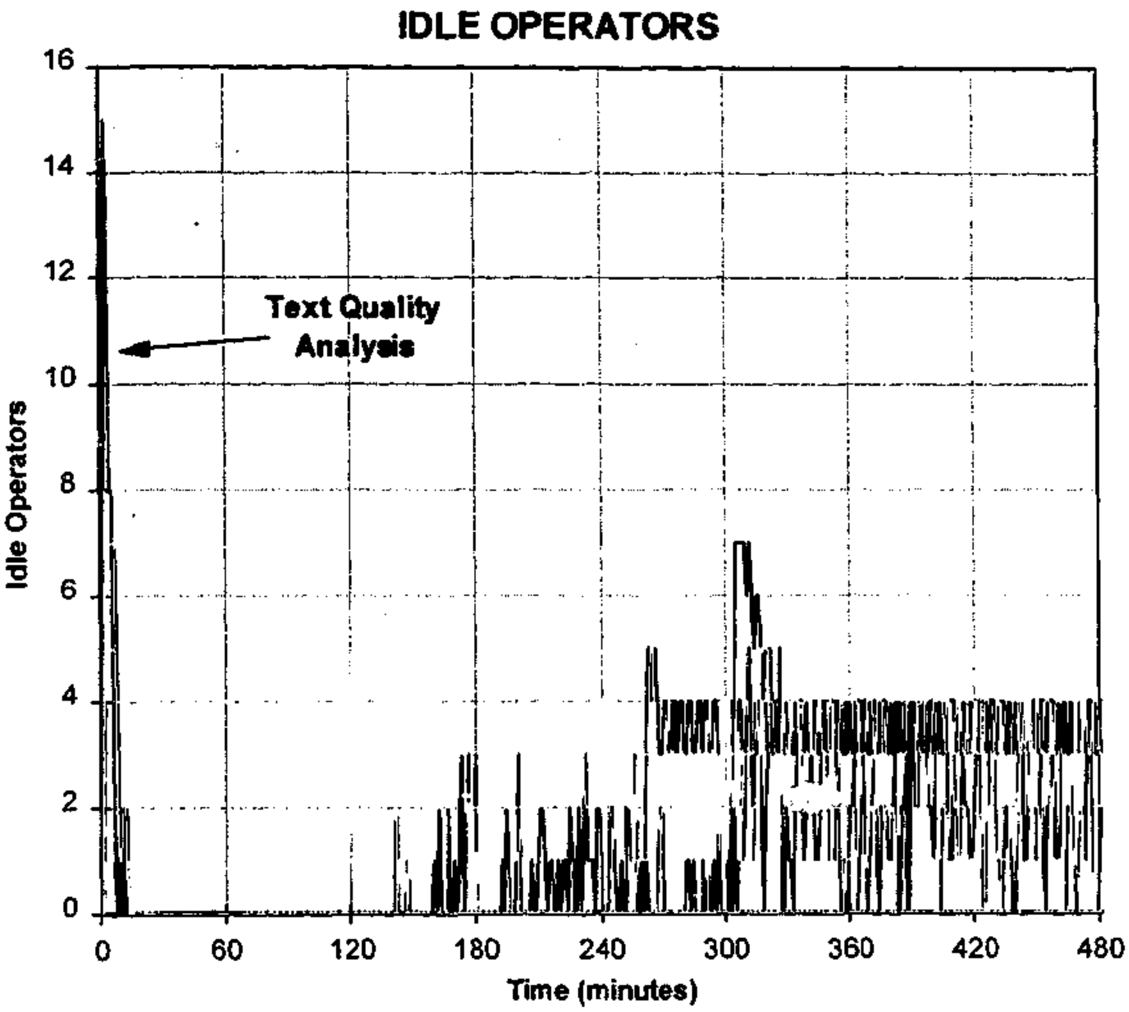


Figure 8.5 Experiment two idle operators
This chart shows the number of idle operators at each process during the eight hour work shift using the OCR system model. Significant idle operator processes are labeled while others are unlabeled to show background activity levels only.

General observations are made from Figure 8.5 concerning the idle operator levels for experiment two, some of which are covered by Section 8.3.1:

- The text-quality-analysis process's idle operator level initially starts at 15 operators then drops to 0 operators at a quicker rate that for experiment one.
- The text-quality-analysis process's idle operator level slowly rises to a peak 7 operators at just after 300 elapsed minutes.

Analysis of the results of this experiment has led to explanations for these two observations, some of which also apply to experiment three.

The text-quality-analysis process's idle operator level starts high because there is a high initial operator level (refer to Table 8.1) and because no documents have yet been passed down to that process. The text-quality-analysis process's idle operator level drops more quickly than in experiment one and stays at zero because of the additional 500 documents starting at process 5 (refer to Figure7.5). These extra documents at process 5 are soon passed on to the text-quality-analysis buffer thus reducing the text-quality-analysis process's idle operator level to zero.

The rise in the text-quality-analysis process's idle operator level is due to the buffer from process 5 (the text recognition process buffer level) being emptied of its original 500 documents (refer to Figure 8.2). As the text-quality-analysis process's buffer drops to zero, the number of idle operators increases until the optimising feature shifts operators to processes with more work to do. Thus there is a peak in the text-quality-analysis process's idle operator level at just after 300 elapsed minutes.

8.3.3.Experiment Three

The experiment three results for the idle operators are taken at the same time as the process buffer and output level results described in Section 8.2.3. The pertinent OCR system model parameters which this experiment is concerned with are listed in Section 8.2.3. These parameters and others associated with experiment three are listed in Appendix F. The idle operator results for experiment three are shown in Figure 8.6.

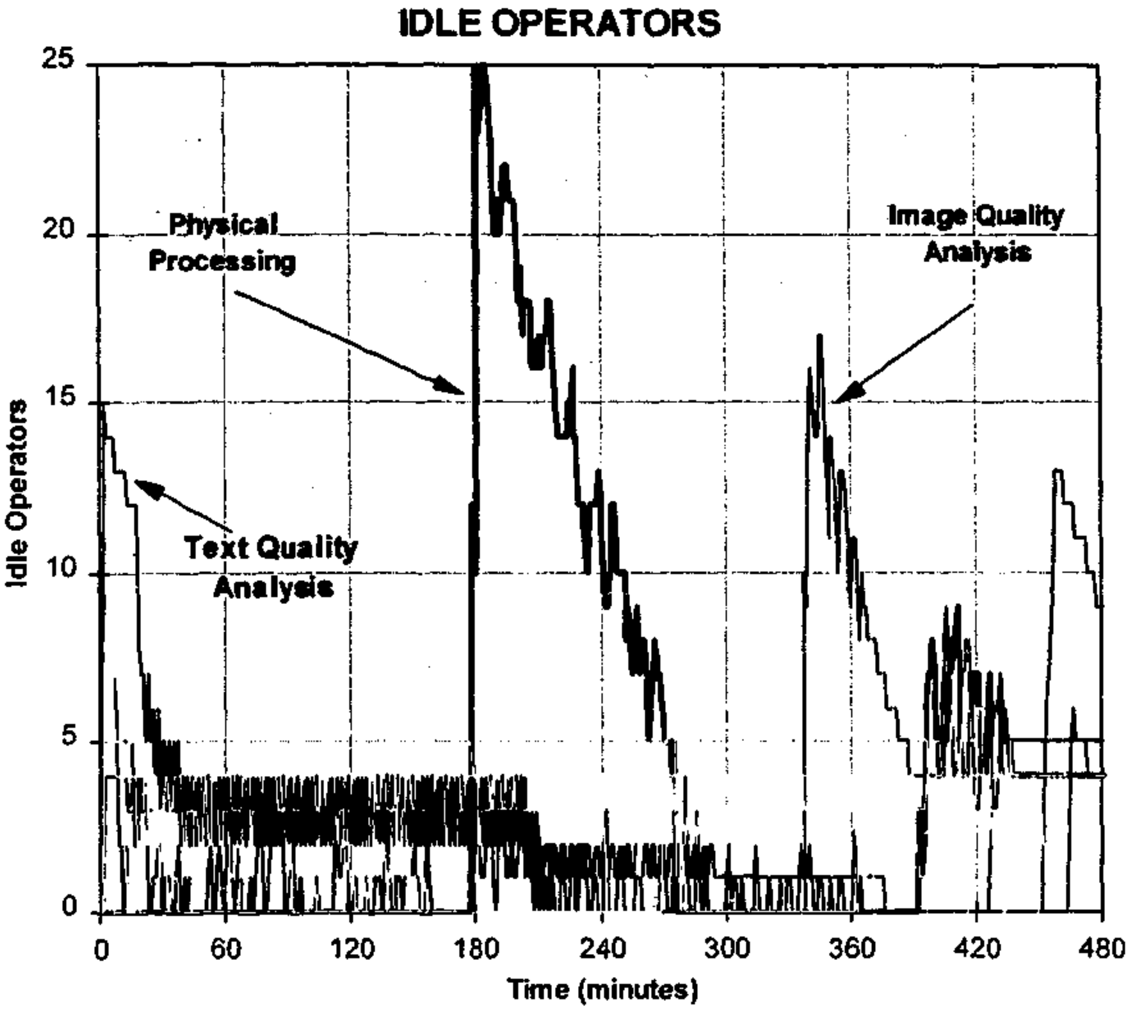


Figure 8.6 Experiment three idle operators
This chart shows the number of idle operators at each process during the eight hour work shift using the OCR system model. Significant idle operator processes are labeled while others are unlabeled to show background activity levels only.

General observations are made from Figure 8.6 concerning the assigned operator levels for experiment one, some of which are covered by Section 8.3.1 and Section 8.3.2:

- As per the general observations in section 8.2.3, there appears to be a lot more activity in experiment three's idle operator results than in the previous two experiments.
- The physical-processing idle operator level rises sharply to 25 operators at 180 minutes, and rises to 9 operators at 390 minutes.
- The image-quality-analysis-process idle operator level rises sharply to 17 operators at 340 minutes.

Analysis of the results of this experiment has led to explanations for these two observations. The analysis also serves to reinforce those findings for experiment one and experiment two.

The increase in activity in Figure 8.6 over Figure 8.4 and Figure 8.5 is due to the fact that all the document pass through the system, i.e. all the document become physically stored and stored as text. As a process's buffer reaches zero and no further documents arrive, operators are left idle and are eventually moved by the optimising feature to later processes which still have documents in their buffers.

The sharp rises in the physical processing and image-quality-analysis idle operator levels correspond to the buffers emptying in those processes as shown in Figure 8.3 and explained further in Section 8.3.2.

8.4. ANALYSIS OF ASSIGNED OPERATORS

This section analyses the experimental results involving operators assigned to each of the processes of the OCR system model. As described in Chapter 7, the assigned operator level at a process is the number of operators that have been selected to work at that process. There are 10 processes in the OCR system model, each with an assigned operator level.

Three different experiments are conducted using the same parameters as those used in Section 8.2 and Section 8.3. The settings of the experimental parameters are listed in Appendix F. The idle operator level results of each experiment are examined in Section 8.4.1, Section 8.4.2 and Section 8.4.3.

8.4.1. Experiment One

The experiment one results for the assigned operators are taken at the same time as the process buffer and output level results described in Section 8.2.1. Table 8.1 lists the pertinent OCR system model parameters which this experiment is concerned with. These parameters and others associated with experiment one are listed in

Appendix F. The assigned operator results for experiment one are shown in Figure 8.7.

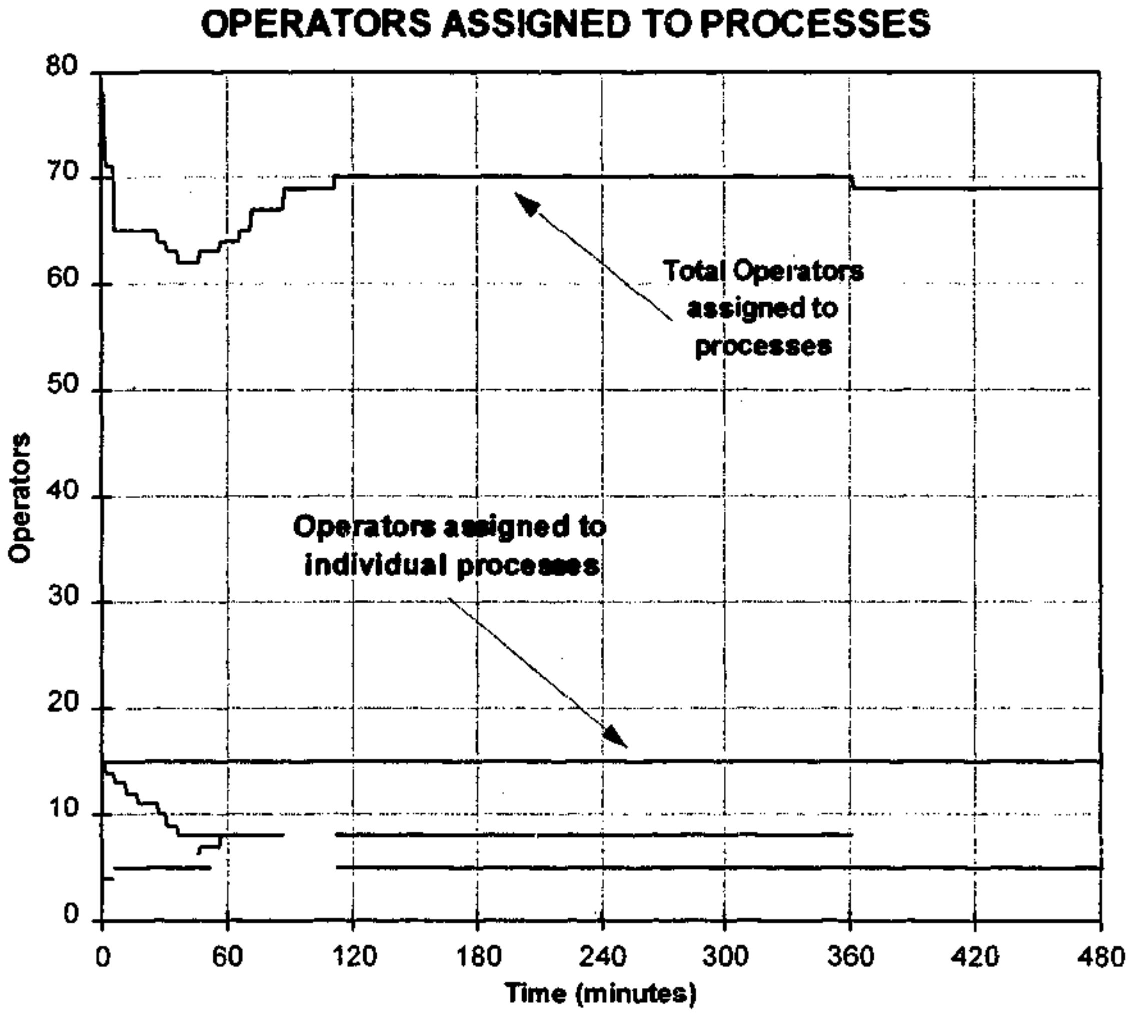


Figure 8.7 Experiment one operators assigned to processes
This chart shows the number of operators assigned to each process and the total number of assigned operators during the eight hour work shift using the OCR system model. Significant processes assigned operator levels are labeled while others are unlabeled to show background activity levels only.

Two general observations are made from Figure 8.7 concerning the idle operator levels for experiment one:

- The period up to 120 minutes shows a dip in the total number of assigned operators, a decrease in some process's assigned operators and an increase in others.

- The period after 120 minutes shows no change in assigned operator levels except for one small drop for a single process, and the corresponding small drop in the total.

Analysis of the results of this experiment has led to explanations for these two observations, some of which also apply to the latter experiments.

The changes in assigned operator levels up to 120 minutes are due to the optimising feature of the OCR system model reassigning operators from processes to and from the operator pool. This corresponds to a similar period of initial activity shown in Figure 8.4 for idle operators.

Once the system has achieved a steady state, no further optimisation is necessary other than for aberrations in document quality, image quality and text quality. Therefore, after 120 minutes the assigned operator levels stay the same, except for the small drop at 360 minutes which also appears as a small peak in Figure 8.4. This drop is attributed to an aberration in image quality.

8.4.2. Experiment Two

The experiment two results for the assigned operators are taken at the same time as the process buffer and output level results described in Section 8.2.2. The pertinent OCR system model parameters that this experiment is concerned with are listed in Section 8.2.2. These parameters and others associated with experiment two are listed in Appendix F. The assigned operator results for experiment two are shown in Figure 8.8.

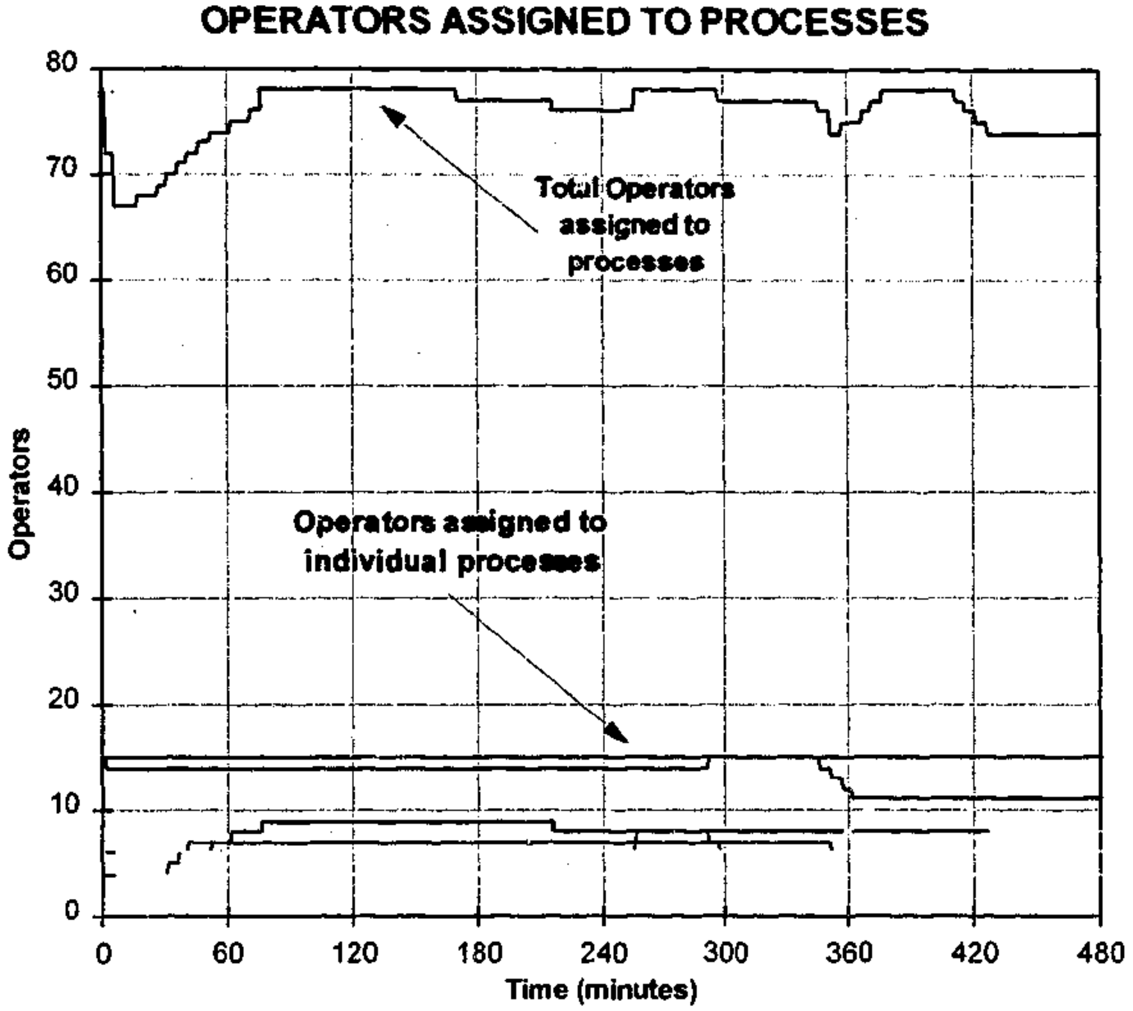


Figure 8.8 Experiment two operators assigned to processes

This chart shows the number of operators assigned to each process and the total number of assigned operators during the eight hour work shift using the OCR system model. Significant processes assigned operator levels are labeled while others are unlabeled to show background activity levels only.

General observations are made from Figure 8.8 concerning the idle operator levels for experiment one, some of which are covered by Section 8.4.1:

- As per Figure 8.7, there is initially a dip in the total number of assigned operators, a decrease in some process's assigned operator levels and an increase in others. The time taken for this initial period is only 80 minutes for experiment two, compared with the 120 minutes for experiment one.
- From 260 minutes onwards there are several changes in the number of operators assigned to processes.

Analysis of the results of this experiment has led to explanations for these two observations, some of which also apply to the experiment one and experiment three.

The decrease in the initial time taken by the optimising feature of the OCR system model is due to the extra initial documents in the text recognition buffer (refer to Figure 8.2). These extra initial documents, mid way through the system, fill the later process's buffers sooner and speeds the operator optimisation. This decrease in optimisation time corresponds to a faster decrease in idle operators in Figure 8.5.

The changes in the number of assigned operators after 260 minutes is due to process's buffers being emptied (refer to Figure 8.2). As the text recognition process buffer and following buffers are emptied, the operators which become idle at those processes are transferred to other processes. There is also increased activity in the idle operator levels during this period (refer to Figure 8.5).

Because the assigned operator level changes are small in this experiment, it can be difficult to see the relationship between process buffer levels being emptied and the changes in the assigned operator levels. Experiment three however, has more apparent changes in process buffer, idle operator and assigned operator levels which serves to highlight the relationship.

8.4.3. Experiment Three

The experiment three results for the assigned operators are taken at the same time as the process buffer and output level results described in Section 8.2.3. The pertinent OCR system model parameters which this experiment is concerned with are listed in Section 8.2.3. These parameters and others associated with experiment three are listed in Appendix F. The assigned operator results for experiment three are shown in Figure 8.9.

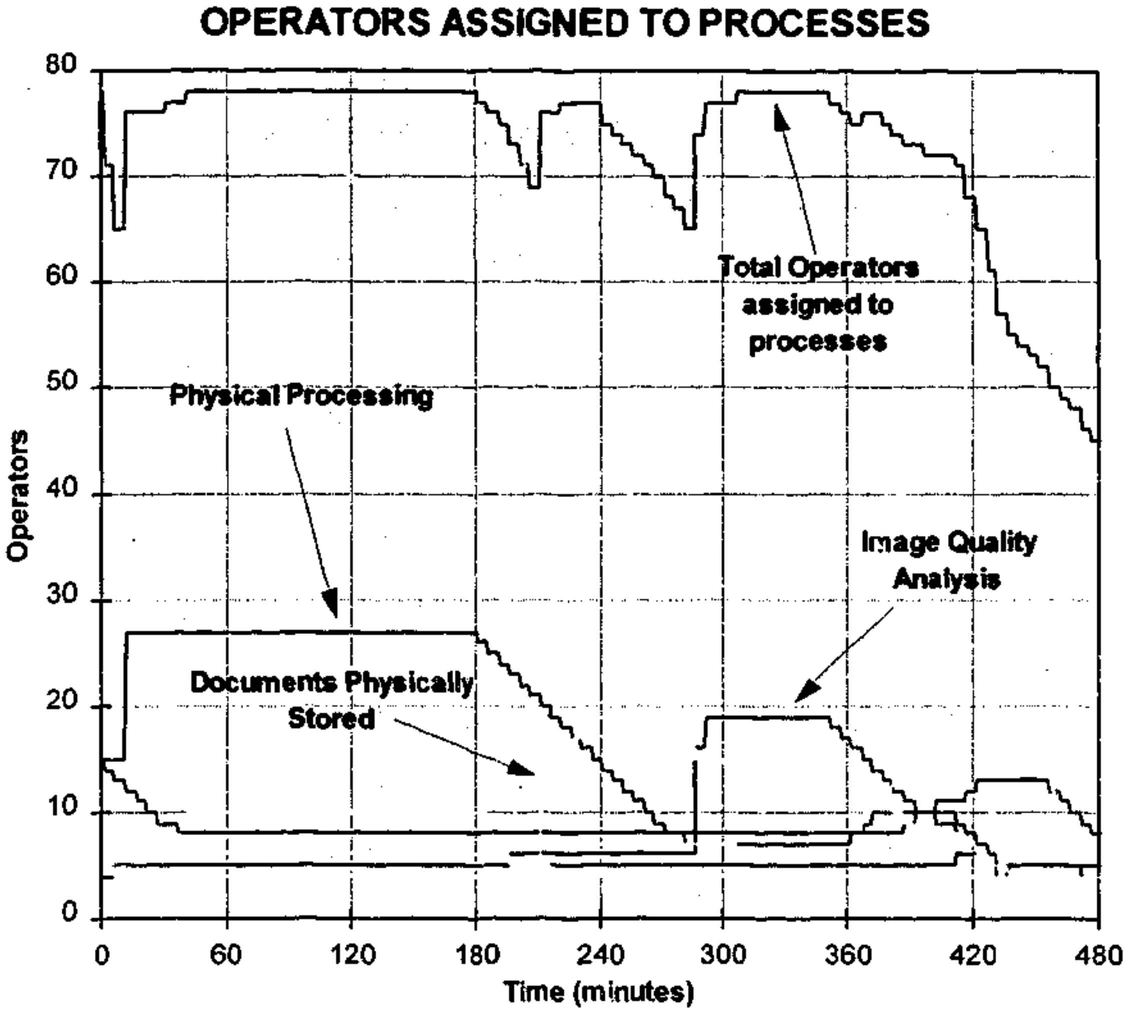


Figure 8.9 Experiment three operators assigned to processes
This chart shows the number of operators assigned to each process and the total number of assigned operators during the eight hour work shift using the OCR system model. Significant processes assigned operator levels are labeled while others are unlabeled to show background activity levels only.

Several general observations are made from Figure 8.9 concerning the assigned operator levels for experiment three, some of which are covered by Section 8.4.1 and Section 8.4.2:

- As per Figure 8.7 and Figure 8.8, there is initially a dip in the total number of assigned operators, a decrease in some process's assigned operator levels and an increase in others. The time taken for this initial period is only 50 minutes for experiment three.

- The physical processing assigned operator level rises quickly to 27 operators at 20 minutes, then declines slowly from 180 minutes onwards.
- The image quality assigned operator level rises quickly at 270 minutes to 19 operators, then declines slowly from 350 minutes onwards.
- There is a gradual decline in the total operators assigned to processes from 350 minutes onwards.

Analysis of the results of this experiment has led to explanations for these general observations, some of which also apply to the first two experiments.

The decline in physical processing buffer at 180 minutes was due to the physical processing process buffer being emptied (refer to Figure 8.3. The number of idle operators for this process also rises sharply at 180 minutes (refer to Figure 8.6).

The rise in the image quality process's assigned operator level at 270 minutes is reflected in the image-quality-process buffer gradient change at the same time (refer to Figure 8.6). The decline in this process's assigned operator level from 350 minutes onwards is due to the process buffer being emptied.

The gradual decline in total operators assigned to processes from 350 minutes onwards is due to the emptying of processes buffers as all the documents are completely converted to text and are physically stored.

8.5. Sensitivity Analysis

This section analyses the experimental results involving the sensitivity of the OCR system model's operator idle cost and processing time to variations in physical document quality, image quality and recognised text quality. These three qualities are described in Chapter 7. The operator idle cost is the sum of the time not spent working by each operator multiplied by the rate at which the operators are paid. The processing time is the total time taken to convert all the documents from their physical form into electronic text.

Three new sets of experiments are conducted. The first set varies the physical image quality from zero to 100 per cent probability of passing in increments of 10 per cent. The second and third set of experiments varies the image and recognised text qualities respectively across the same range and with the same resolution as the first set. This selection of experimental parameter ranges is chosen because it ensures an even coverage of samples through the possible ranges. If greater detail is required in a particular region, then the sample resolution for that region can be increased later and more samples taken.

Some of the OCR system model parameters settings used for these experiments are listed in Table 8.1 while others are listed in Appendix F. The setting for the initial physical processing buffer is lowered to 400 documents so that all the documents can be processed in the single eight hour work period.

The results of the operator idle costs are shown together so that the effects of the three document qualities on operator idle cost sensitivity can be compared. Similarly, the results of the processing times are shown together so that the effects of the three document qualities on processing time sensitivity can be compared.

8.5.1.Operator Idle Cost

This section analyses the effects of the three document qualities on operator idle cost sensitivity. The operator idle cost sensitivity results are shown in Figure 8.10.

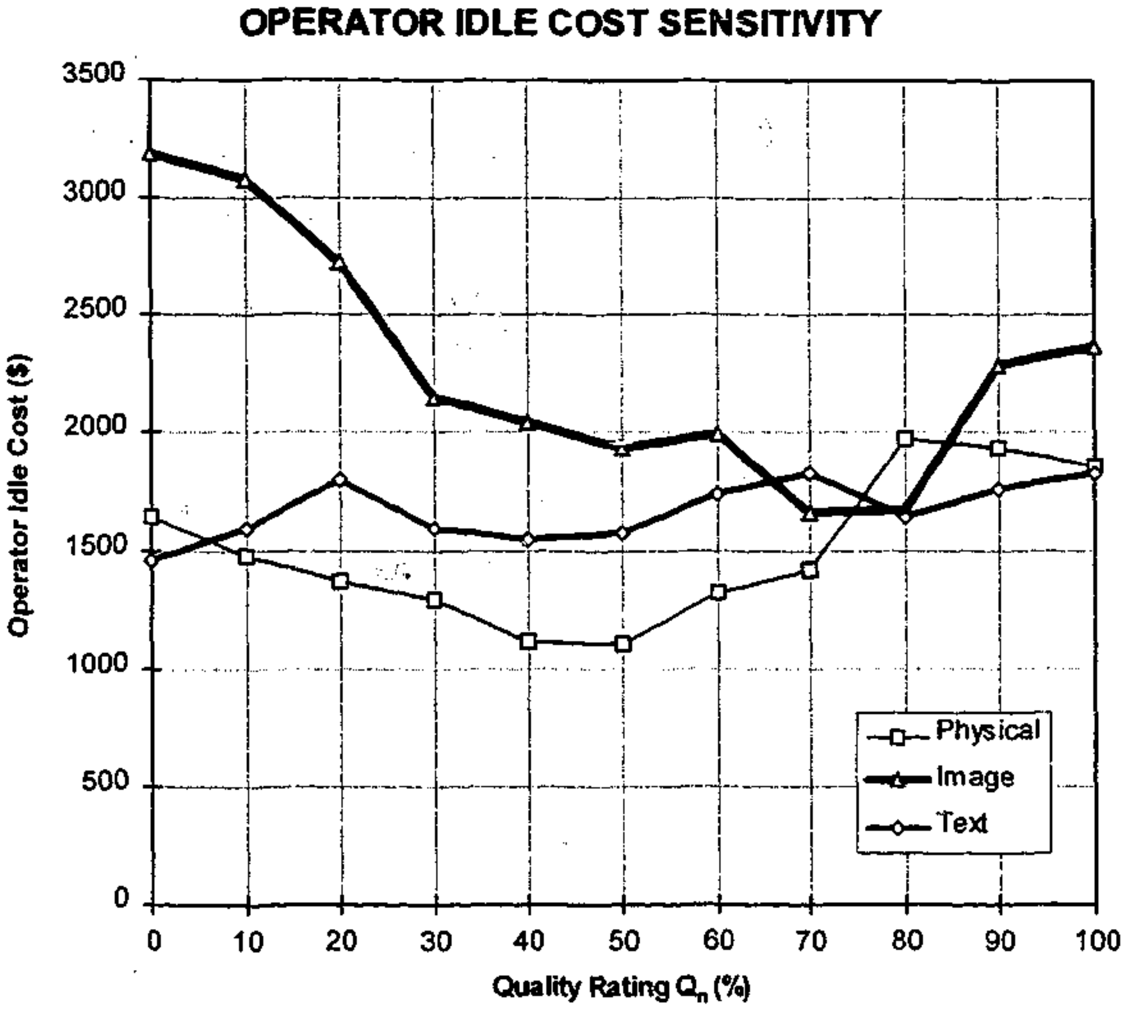


Figure 8.10 Operator idle cost sensitivity
This chart shows the sensitivity of the operator idle cost to variations in the physical document quality, image quality and recognised text quality.

Several general observations are made from Figure 8.10 concerning the effects of physical, image and text qualities on the operator idle cost.

- The image quality rating appears to have the greatest potential for increasing operator idle cost over most of the range shown.
- The physical quality rating offers the lowest operator idle cost over the range 10 to 70 per cent.
- At 75 per cent ratings, the physical, image and text quality give approximately the same idle operator cost.

- There are two slight peaks in operator idle cost for text quality ratings of 20 and 70 percent.

Analysis of the results of this experiment has led to some explanations for these general observations.

Examination of the process flow shown in Figure 7.5 reveals the reason why low image quality has a greater effect on operator idle cost that either low physical or low text quality. The process flow diagram for the OCR system model shows that documents which fail the image quality analysis are returned to the physical processing process. This feedback loop in the process flow causes low image quality documents to repeat the loop several times, thus increasing the time taken to process such documents and also the number of idle operators outside the loop.

Document which fail their physical or text quality checks follow alternative rather that feedback process paths. Therefore physical and text quality ratings do not have as profound effects upon the operator idle cost.

8.5.2.Processing Time

This section analyses the effects of the three document qualities on processing time sensitivity. The processing time sensitivity results are shown in Figure 8.10.

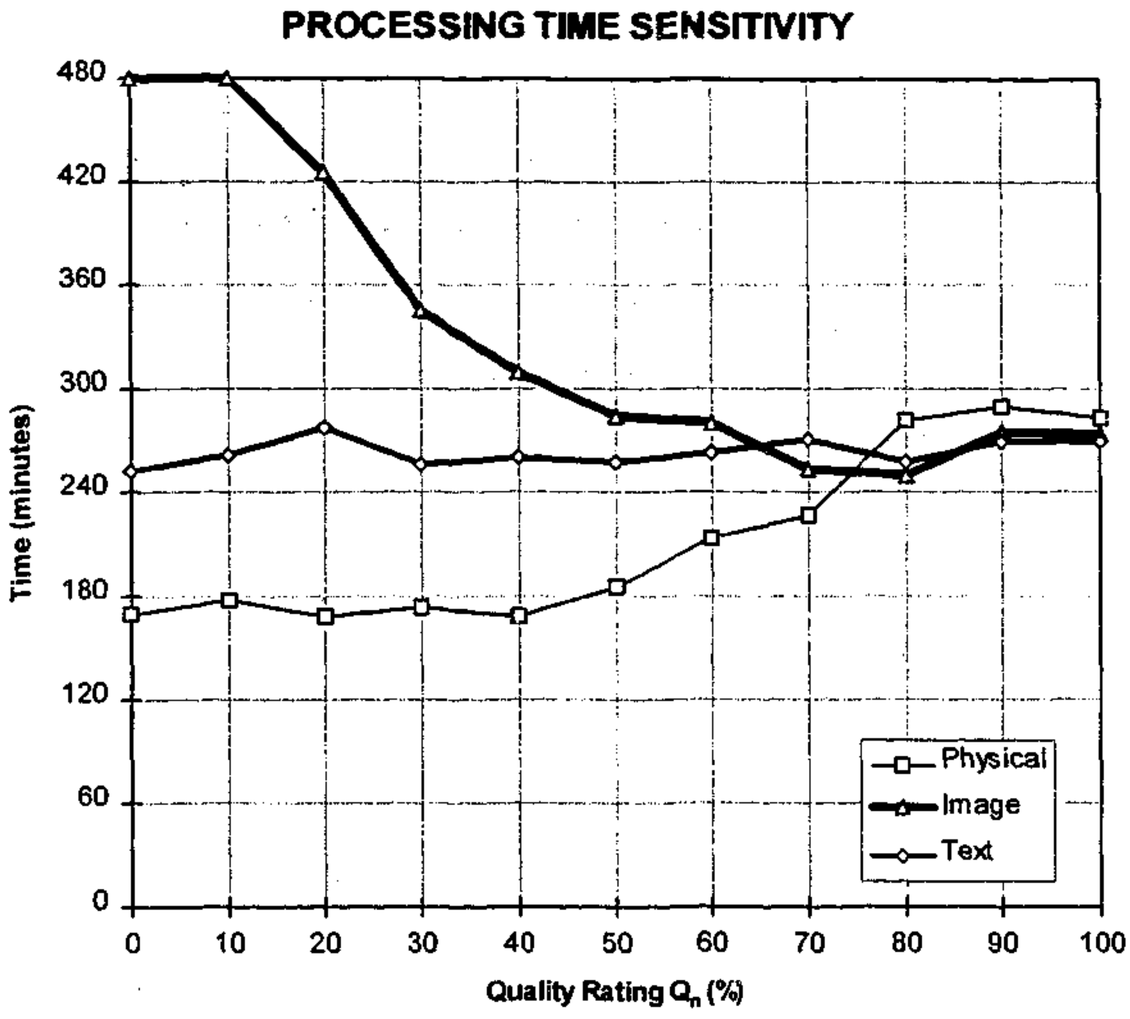


Figure 8.11 Processing time sensitivity
This chart shows the sensitivity of the operator idle cost to variations in the physical document quality, image quality and recognised text quality.

Several general observations are made from Figure 8.11 concerning the effects of physical, image and text qualities on processing time.

- The image quality rating appears to have the greatest potential for increasing processing time over most of the range shown. The processing times at 0 and 10 per cent image quality ratings were in fact greater than the eight hour shift duration.
- The physical quality rating offers the lowest processing time over the range 0 to 70 per cent.
- At 75 per cent ratings, the physical, image and text quality give approximately the same processing time.

- There are two slight peaks in processing time for text quality ratings of 20 and 70 percent.

Analysis of the results of this experiment has led to some explanations for these general observations.

The cause of the increase in processing time due to low image quality rating can be found from analysis of the OCR system model's process flow diagram (refer to Figure 7.5) and is explained in Section 8.5.1.

Several general observations are also made from Figure 8.10 and Figure 8.11 concerning the similarity between the results shown in the two figures.

- There is a good degree of correlation between the shapes of the two graphs. The image quality line appears above the text quality line in most cases, which in turn appears above the physical quality line in most cases. As the quality factor increases, the distance between the quality rating lines generally becomes less.
- A crossover point where the lines showing physical, image and text quality meet occurs at the same quality rating point, 75 per cent, on both graphs.
- The two peaks for text quality rating in operator idle time correspond to the two peaks for text quality rating in processing time.

Analysis of the similarity between the sensitivity results shown in Figure 8.10 and Figure 8.11 has led to some explanations for these general observations.

The good degree of correlation between the shapes of the two graphs is due to the OCR system model's interdependence of the operator idle cost and processing time. This also explains two matching peaks in both graphs for text quality.

The matching crossover points on both graphs at 75 per cent quality ratings is due partly to this interdependence and partly to other factors related to the programming of the OCR system model.

8.6. SUMMARY OF ANALYSIS

This section summarises the analysis of the OCR system model experimentation work as applied to a simulated high volume document processing environment. It also lists areas of the OCR system model where additional analysis could prove insightful.

The analysis of the OCR system model experimentation work is summarised into the following general observations:

- Analysis of the process buffer and output levels shows those processes which could most benefit from additional operators. This is the basis by which operator optimisation procedure determines which processes release operators the operator pool (described in Chapter 7) and which processes request operators from the operator pool.

- Analysis of the idle operator levels shows the effects of the operator optimisation procedures on number of idle operators and the rate at which those idle operators are reduced.

- Analysis of the assigned operator levels complements the analysis of the idle operators by showing the speed with which the OCR system model compensates for variations in load for the individual processes in the system.

- Sensitivity analysis of the operator idle cost and processing time to variations in the document quality, image quality and text quality ratings showed the sensitivity of the model to perturbations in the image quality. Because the image-quality-analysis process provides feedback to earlier processes (refer to Figure 7.5), low image quality is more likely to cause variations in operator idle cost and processing time than either low document quality or low text quality.

There are several areas where additional analysis of the OCR system model could provide further knowledge in the area. Examination of the effects of the operator transfer delay time upon the model performance could lead further performance gains. The operator transfer delay controls the time taken for operators to move between the operator pool and processes. A similar examination of the operator number transfer factor could also lead to further performance gains. The operator number transfer factor controls the proportion of available operators in the operator pool which can be assigned to processes.

The analysis of the OCR system model experimentation presented in this chapter provides a detailed insight into the operation of digital image processing systems in the high volume document environment. It also demonstrates the potential of the model for improving the performance of OCR document processing systems in that environment. The OCR system model is unique amongst those models surveyed in the literature in Chapter 2 and represents a significant advancement in the knowledge of digital image document processing.

CHAPTER 9

CONCLUSION

CHAPTER CONTENTS PAGE

9. CONCLUSION

This thesis reports original research carried out on digital image processing in the high volume document environment. The research focuses on experimentation and modelling of the document and image processing with particular emphasis on performance optimisation using the developed models. The performance optimisation is successfully proven on both low level OCR systems and higher level document processing systems. The conclusions drawn from the research work are consistent with the general and specific objectives stated in Chapter 1.

A summary of the general achievements of the research is presented which highlights the significant developments. The original contributions of the research to the knowledge of digital image document processing are described. The directions of possible future work in the research area is indicated, including further possible development of apparatus and research tools.

9.1. SUMMARY OF GENERAL ACHIEVEMENT

This section summarises the general achievements of the research work. Chapter 2 reported research on digital image document processing. It presented an historical perspective of OCR systems, examined current state of the art OCR systems and key enabling technologies, and presented an overview of current digital image document processing theories.

The research work reported in this thesis progressed in two phases. The first phase concentrated on low level OCR, while the second focused on high level digital image document processing. The first phase of the research work included development of analytical tools and models for optimisation of the low level OCR processes in the high volume document environment. The LitBase analytical tools (Chapter 3) which were developed specifically for this research have been shown to locate articles using search terms with an accuracy of 98.8%. The initial experimental work reported in Chapter 4 into OCR performance classes and characteristic variables provided conclusive evidence of the existence of a link between the performance classes of OCR systems in the high volume document environment. The effects of the characteristic variables on the performance classes are quantified for these OCR systems.

Extensive analysis of the preliminary experimental results (Chapter 5) led to the definition of the relationship between the OCR performance classes and characteristic variables and the development of the accuracy-resulution-text_size-speed (ARTS) curve. The ARTS curve is a multi-dimensional surface model for visualisation of the relationship between the OCR performance classes and characteristic variables. Elementary mathematical models representing the ARTS curve were developed and are shown to predict the performance of an OCR system given a particular set of characteristic variables.

The ARTS curve and associated mathematical models were developed into an analysis tool for OCR system performance optimisation (OSPO). OSPO is shown to be able to tune the characteristic variables of an OCR system to optimise the performance classes in the high volume environment. The prototype OCR systems reported in Chapter 6 substantiate the findings obtained from analysis of the experimental results in Chapter 4. The tools and models developed from Chapter 5 were shown to reliably operate on the prototype OCR systems, even when those systems are subjected to non-synthesised real world documents.

The preceding research work, models and tools developed successfully conclude the first phase of the research work. The first phase presents an original technique and model for optimising the performance of the OCR process in the high volume environment. The second phase of the research work extends the optimisation and modelling work to include the whole digital image document processing system of which the OCR process examined in the first phase is a key component.

The optimisation of the digital image document processing system begins by development of a series of increasingly sophisticated document processing system models. The development of these models is reported in Chapter 7 and culminates in a document processing model which is shown to successfully model high volume document processing systems which are subjected to real world variables and constraints. The model experimentation and analysis (Chapter 8) show the ability of the models to successfully optimise the performance of document processing systems under conditions typical of the high volume environment. The document processing models incorporate the OCR optimisation model developed in Chapters 4 and 5. By incorporating the low level OCR model within the high level document processing model, it is shown that optimisation of the combined model is superior to optimisation of the low level and high level models individually.

The general achievement of the research work which is reported in the preceding chapters is that optimisation of real world digital image processing systems in the high volume document environment is possible using sufficiently sophisticated models. This general achievement includes several original and significant contributions to the knowledge of digital image document processing.

9.2. ORIGINAL CONTRIBUTION

The contribution of the research reported in this thesis to the knowledge of digital image document processing in the high volume environment is divided into four distinct and original contributions:

- A comprehensive set of experimental results which define the effects of certain characteristic variables on the performance classes of OCR systems.
- An original model for describing the performance of OCR systems in terms of the systems characteristic variables;

- An original model for describing the performance of digital image processing systems in the high volume document environment;
- A series of tools for analysing OCR systems and digital image document processing systems.

The four contributions listed above represent the most significant and original of the many that are presented in this thesis.

9.3. FUTURE WORK

There is scope for future research and development work to be carried out in the area of research reported in this thesis. Further development of the apparatus and research tools is possible to provide greater experimental precision and accuracy. Several possible avenues of future research work are reported which build upon this thesis' work.

9.3.1. Apparatus and Research Tool Development

Future experimentation work using the present apparatus and research tools would be greatly facilitated by upgrading the document processing hardware. The processing rates for the LitBase system reported in Chapter 3 would be significantly improved by the addition of one or more faster document scanning devices and a more powerful OCR processing module. A more detailed and extensive analysis from LitBase would result from implementing these improvements.

Refinements to the OSPO optimisation tool to allow the display of three dimensional surface data would enhance the visualisation of the optimisation data. The ability to set some performance classes and characteristic variables to a particular axis while varying others would allow the operator to interactively visualise the relationship between the performance classes and characteristic variables.

9.3.2. Research Directions

There are several potentially useful directions that the research work reported in this thesis could take. The following list represents the higher priority directions of future research which are apparent from this research:

- Further refinement of the document processing models to produce a model which better represents the high volume document processing systems and improves the optimisation of these systems;
- Adaptation of the developed OCR system and document processing system optimisation techniques to similar areas such as cursive script recognition and other language recognition;
- Refinement of the ARTS model and OCR optimising system to incorporate characteristic variables such as degradation and noise which were previously treated as constraints.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] S. Mitchel, "The two-headed giant," *Australian Personal Computer*, pp. 78-82, January 1997.

[2] D. S. Doermann, "Document image understanding: integrating recovery and interpretation," PhD Dissertation, University of Maryland, Maryland, U.S.A., 1993.

[3] S. Impedovo, L. Ottaviano, S. Occhinegro, "Optical character recognition - a survey," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 5, no. 1&2, pp. 1-24, 1991.

[4] G. Nagy, "At the frontiers of OCR," *Proceedings of the IEEE*, vol. 80, no. 7, pp. 1193-1100, July 1992.

[5] S. Mori, C. Y. Suen, K. Yamamoto, "Historical review of OCR research and development," *Proceedings of the IEEE*, vol. 80, no. 7, pp. 1029-1058, July 1992.

[6] M. Bokser, "Omnidocument technologies," *Proceedings of the IEEE*, vol. 80, no. 7, pp. 1066-1078, July 1992.

[7] H. S. Baird, "The skew angle of printed documents," *Proceedings of the Conference of the Society of Photographic Scientists and Engineers*, pp. 14-21, 1987.

[8] H. S. Baird, "Document image defect models," *Structured Document Image Analysis*, pp. 546-556, 1992.

[9] H. S. Baird, "Document image defect models and their uses," *Proceedings of the Second International Conference on Document Analysis and Recognition*, vol. 1, pp. 62-67, 1993.

[10] B. Griffin, K. Spriggs, G. Vains, W. Nageswaran, "OCR performance in a high volume commercial environment," *Proceedings of Digital Image Computing: Techniques and Applications*, vol. 2, pp. 525-532, 1993.

[11] T. K. Ho, H. S. Baird, "Evaluation of OCR accuracy using synthetic data," *Proceedings of the Fourth Annual Symposium on Document Analysis and Information Retrieval*, pp. 413-422, 1995.

[12] D. Doermann, S. Yao, "Generating synthetic data for text string analysis," *Proceedings of the Fourth Annual Symposium on Document Analysis and Information Retrieval*, pp. 449-467, 1995.

[13] B. Griffin, K. Spriggs, Y. Ibrahim, G. Vains, "OCR performance optimisation in a high volume commercial environment," *Proceedings of Digital Image Computing: Techniques and Applications*, pp. 485-490, 1995.

[14] R. Casey, D. Ferguson, K. Mohiuddin, E. Walach, "Intelligent forms processing system," Machine Vision and Applications, vol. 5, pp. 143-155, 1992.

[15] S. N. Srihari, "High-performance reading machines," *Proceedings of the IEEE*, vol. 80, no. 7, pp. 1121-1132, July 1992.

[16] T. Sakai, "A history and evolution of documen information processing," *Proceedings of the Second International Conference on Document Analysis and Recognition*, vol. 1, pp. 377-384, 1993.

[17] W. S. Davis, A. McCormack, *The Information Age*, Addison Wesley, Reading, 1979.

[18] J. Tovey, *The Technique of Weaving*, Batsford, London, 1975.

[19] J. Kanai, T. A. Nartker, S. V. Rice, G. Nagy, "Performance metrics for document understanding systems," *Proceedings of the Second International Conference on Document Analysis and Recognition*, vol. 1, pp. 424-427, 1993.

[20] K. O. Grover (ed.), *Information Science Research Institute 1993 Annual Research Report*, University of Nevada, Las Vegas, 1993.

[21] L. R. Blando, J. Kanai, T. A. Nartker, "Prediction of OCR accuracy using simple shape features," *Proceedings of the Third International Conference on Document Analysis and Recognition*, vol. 1, pp. 319-322, 1995.

[22] J. Kreich, "Robust recognition of documents," *Proceedings of the Second International Conference on Document Analysis and Recognition*, vol. 1, pp. 444-447, 1993.

[23] D. P. Loprsti, J. S. Sandberg, "Certifiable optical character recognition," *Proceedings of the Second International Conference on Document Analysis and Recognition*, vol. 1, pp. 432-435, 1993.

[24] D. J. Lee, S. W. Lee, "A new methodology for grey-scale character segmentation and recognition," *Proceedings of the Third International Conference on Document Analysis and Recognition*, vol. 1, pp. 524-537, 1995.

[25] L. Wang, T. Pavlidis, "Direct grey-scale extraction of features for character recognition," *IEEE: Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 10, October 1993.

[26] K. O. Grover (ed.), *Information Science Research Institute 1994 Annual Research Report*, University of Nevada, Las Vegas, 1994.

[27] J. A. Vlontzos, S. Y. Kung, "Hidden Markov Models for character recognition," *IEEE Transactions on Image Processing*, vol. 1, no. 4, pp. 539-543, October 1992.

[28] Y. Zhao, X. Zhuang, L. Atlas, L. Anderson, "Parameter estimation & restoration of noisy images using Gibbs distributions in Hidden Markov Models," *CVGIP: Graphical Models and Image Processing*, vol. 54, no. 3, pp. 187-197, May 1992.

[29] C. Yen, S. S. Kuo, "Degraded grey-scale text recognition using pseudo-2D Hidden Markov Models and N-best hypotheses," *CVGIP: Graphical Models and Image Processing*, vol. 57, no. 2, pp. 131-145, March 1995.

[30] J. C. Anigbogu, A. Belaid, "Application of Hidden Markov Models to multifont text recognition," *Proceedings of the First International Conference on Document Analysis and Recognition*, vol. 2, pp. 785-793, 1991.

[31] O. E. Agazzi, S. Kuo, "Joint normalization and recognition of degraded document images using pseudo-2D Hidden Markov Models," *Proceedings of the Second International Conference on Document Analysis and Recognition*, vol. 1, pp.155-158, 1993.

[32] C. Pearce, "Dynamic hypertext links for highly degraded data in TELLTALE," *Proceedings of the Fourth Annual Symposium on Document Analysis and Information Retrieval*, pp. 89-106, 1995.

[33] C. Fang, J. J. Hull, "A word-level deciphering algorithm for degraded document recognition," *Proceedings of the Fourth Annual Symposium on Document Analysis and Information Retrieval*, pp. 191-202, 1995.

[34] M. A. Jones, G. A. Story, B. W. Ballard, "Integrating multiple knowledge sources in a Baysian OCR post-processor," *Proceedings of the First International Conference on Document Analysis and Recognition*, vol. 2, pp. 925-933, 1991.

[35] H. Bunke, R. Liviero, "Classification and postprocessing of documents using an error-correcting parser," *Proceedings of the Third International Conference on Document Analysis and Recognition*, vol. 1, pp. 222-226, 1995.

[36] K. Kigo, "Improving speed of Japanese OCR through linguistic preprocessing," *Proceedings of the Second International Conference on Document Analysis and Recognition*, vol. 1, pp.214-217, 1993.

[37] R. Hoch, T. Kieninger, "On virtual partitioning of large dictionaries for contextural post-processing to improve character recognition," *Proceedings of the Second International Conference on Document Analysis and Recognition*, vol. 1, pp. 226-231, 1993.

[38] R. M. K. Sinha, B. Prasada, G. F. Houle, M. Sabourin, "Hybrid contextural text recognition with string matching," *IEEE: Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 9, September 1993.

[39] R. N. Bozinovic, S. N. Srihari, "Off-line cursive script word recognition," *IEEE: Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 1, pp. 68-83, January 1989.

[40] M. Parizeau, R. Plamondon, "A fuzzy-syntactic approach to allograph modeling for cursive script recognition," *IEEE: Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 7, pp. 702-712, July 1995.

[41] K. Fukushima, N. Wake, "Handwritten alphanumeric character recognition by the neocognitron," *IEEE Transactions on Neural Networks*, vol. 2, no. 3, pp 355-365, May 1991.

[42] E. J. Bellegarda, J. R. Bellegarda, D. Nahamoo, K. S. Nathan, "A fast statistical mixture algorithm for on-line handwriting recognition," *IEEE:*

Transactions on Pattern Analysis and Machine Intelligence, vol. 16, no. 12, December 1994.

[43] J. C. Simon, "Off-line cursive word recognition," *Proceedings of the IEEE*, vol. 80, no. 7, pp 1150-1161, July 1992.

[44] C. J. C. Burges, J. I. Ben, J. S. Denker, Y. Lecun, C. R. Nohl, "Off line recognition of handwritten postal words using neural networks," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 7, no. 4, pp. 689-704, 1993.

[45] G. Seni, R. K. Srihari, N. Nasrabadi, "Large vocabulary recognition of on-line handwritten cursive words," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 7, pp. 757-762, July 1996.

[46] H. Fujisawa, K. Marukawa, "Full-text search and document recognition of Japanese text," *Proceedings of the Fourth Annual Symposium on Document Analysis and Information Retrieval*, pp. 55-80, 1995.

[47] H. Guo, X. Ding, Z. Zhang, F. Guo, Y. Wu, "Realization of a high performance bilingual Chinese-English OCR system," *Proceedings of the Third International Conference on Document Analysis and Recognition*, vol. 2, pp. 978-981, 1995.

[48] T. Hisamitsu, K. Marukawa, Y. Shima, H. Fujisawa, Y. Nitta, "Optimul techniques in OCR error correction for Japanese texts," *Proceedings of the Third International Conference on Document Analysis and Recognition*, vol. 2, pp. 1014-1017, 1995.

[49] R. Romero, R. Berger, R. Thibadeau, D. Touretzky, "Neural network classifiers for optical Chinese character recognition," *Proceedings of the Fourth Annual Symposium on Document Analysis and Information Retrieval*, pp. 385-398, 1995.

[50] C. H. Tung, Y. J. Chen, H. J. Lee, "Performance analysis of an OCR system via an artificial handwritten Chinese character generator," *Proceedings of the Second International Conference on Document Analysis and Recognition*, vol. 1, pp. 315-318, 1993.

[51] R. Bradford, T. Nartker, "Error correlation in contempory OCR systems," *Proceedings of the First International Conference on Document Analysis and Recognition*, vol. 1, pp. 516-524, 1991.

[52] R. G. Garcia, Y. A. Dimitriadis, F. M. Pastor, J. L. Coronado, "Error detection in character recognition using pseudosyallable analysis," *Proceedings of the Third International Conference on Document Analysis and Recognition*, vol. 1, pp. 446-449, 1995.

[53] A. W. Senior, "Off-line handwriting recognition: a review and experiments," Unpublished technical report, Cambridge University Engineering Department, Cambridge, England, 1992.

[54] C. H. Chen, J. L. DeCurtins, "Word recognition in a segmentation-free approach to OCR," *Proceedings of the Second International Conference on Document Analysis and Recognition*, vol. 1, pp. 573-576, 1993.

[55] M. A. O'Hair, M. Kabrisky, "Recognizing whole words as symbols," *Proceedings of the First International Conference on Document Analysis and Recognition*, vol. 1, pp. 350-358, 1991.

[56] T. K. Ho, J. J. Hull, S. N. Srihari, "Word recognition with multi-level contextural knowledge," *Proceedings of the First International Conference on Document Analysis and Recognition*, vol. 2, pp. 905-915, 1991.

[57] S. Rai, S. Khan, "Unconstrained handprinted character recognition using simple shape features," *Proceedings of Digital Image Computing: Techniques and Applications*, pp. 152-157, 1991.

[58] T. Wakahara, H. Murase, K. Odaka, "On-line handwriting recognition," *Proceedings of the IEEE*, vol. 80, no. 7, pp. 1181-1194, July 1992.

[59] T. W. Pai, T. M. Wu, G. H. Chang, P. Y. Ting, "An intelligent Chinese official document processing system," *Proceedings of the Third International Conference on Document Analysis and Recognition*, vol. 2, pp. 974-977, 1995.

[60] D. Avendon, J. R. Levy, *Electronic Imaging Systems - Design, Applications, and Management*, McGraw-Hill, New York, 1994.

[61] R. Yiacoumi, "Toshiba DVD-ROM drive and Cinemaster," *Australian Personal Computer*, p. 33, March 1997.

[62] L. J. Jr. Galbiati, *Machine Vision and Digital Image Processing Fundamentals*, Prentice-Hall, London, 1990.

[63] M. Sonka, V. Hlavac, R. Boyle, *Image Processing, Analysis and Machine Vision*, Chapman & Hall, London, 1993.

[64] L. O'Gorman, R. Kasturi, *Document Image Analysis Systems*, IEEE Computer Society Press, Los Alamitos, 1995.

[65] M. Kamel, A. Zhao, "Extraction of binary characters/graphics images from greyscale document images," *CVGIP: Graphical Models and Image Processing*, vol. 55, no. 3, pp. 203-217, May 1993.

[66] G. Srikantan, D. S. Lee, J. T. Favata, "Comparison of normalization methods for character recognition," *Proceedings of the Third International Conference on Document Analysis and Recognition*, vol. 2, pp. 719-722, 1995.

[67] T. Pavlidis, J. Zhou, "Page segmentation and classification," *CVGIP: Graphical Models and Image Processing*, vol. 54, no. 6, pp. 484-496, November 1992.

[68] H. Fujisawa, Y. Nakano, K. Kurino, "Segmentation methods for character recognition: from segmentation to document structure," *Proceedings of the IEEE*, vol. 80, no. 7, pp. 1079-1092, July 1992.

[69] R. G. Casey, "Character segmentation in document OCR: progress and hope," *Proceedings of the Fourth Annual Symposium on Document Analysis and Information Retrieval*, pp. 13-39, 1995.

[70] T. Saitoh, M. Tachikawa, T. Yamaai, "Document image segmentation and text area ordering," *Proceedings of the Second International Conference on Document Analysis and Recognition*, vol. 1, pp. 323-329, 1993.

[71] S. I. Olsen, "Estimation of noise in images: an evaluation," *CGVIP: Graphical Models and Image Processing*, vol. 55. no. 4, pp. 319-323, July 1993.

[72] R. M. Harlick, S. R. Sternberg, X. Zhuang, "Image analysis using mathematical morphology," *IEEE: Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 4, pp. 532-550, July 1987.

[73] S. Liang, M. Ahmadi, M. Shridhar, "A morphological approach to text string extraction from regular periodic overlapping text/background," *CVGIP:*

Graphical Models and Image Processing, vol. 56, no. 5, pp. 402-413, September 1994.

[74] D. Wang, S. N. Srihari, "Analysis of form images," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 8, no. 5, pp. 1031-1052, 1994.

[75] P. Wayner, "Optimul character recognition," *Byte*, pp. 203-210, December 1993.

[76] H. S. Baird, R. Fossey, "A 100-font classifier," *Proceedings of the First International Conference on Document Analysis and Recognition*, vol. 1, pp. 332-340, 1991.

[77] T. Kanungo, R. M. Haralick, I. Phillips, "Global and local document degradation models," *Proceedings of the Second International Conference on Document Analysis and Recognition*, vol. 1, pp. 730-733, 1993.

[78] W. B. Croft, S. Harding, K. Taghva, J. Borsack, "An evaluation of information retrieval accuracy with simulated OCR output," Unpublished technical report, Computer Science Department, University of Massachusetts, Amherst, 1993.

[79] M. Helander (ed.), *Handbook of Human-Computer Interaction*, North-Holland, Amsterdam, 1992.

[80] J. Preece (ed.), *A Guide to Usability: Human Factors in Computing*, Addison Wesley, Wokingham, 1993.

[81] C. M. Brown, *Human-Computer Interface Design Guidelines*, Ablex Publishing Corporation, New Jersey, 1988.

[82] C. Y. Suen, "Character recognition by computer and applications," *Handbook of Pattern Recognition and Image Processing*, pp. 569-586, 1986.

[83] T. M. Ha, H. Bunke, "Model-based analysis and understanding of check forms," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 8, no. 5, pp. 1053-1080, 1994.

[84] L. Mui, A. Gupta, P. S. Wang, "An adaptive modular neural network with application to unconstrained character recognition, " *International Journal of Pattern Recognition and Artificial Intelligence*, vol.8, no. 5, pp. 1189-1204, 1994.

[85] E. Mittendorf, P. Schauble, P. Sheridan, "Applying probabilistic term weighting to OCR text in the case of a large alphabetic library catalogue," Unpublished technical report, Swiss Federal Institute of Technology, Zurich, Switzerland, 1995.

[86] A. Dengel, R. Bleisinger, R. Hoch, F. Fein, F. Hones, "From paper to office document standard representation," *Computer*, vol. 25, no. 7, pp. 63-67, July 1992.

[87] S. L. Taylor, R. Fritzson, J. A. Pastor, "Extraction of data from preprinted forms," *Machine Vision and Applications*, vol. 5, pp. 211-222, 1992.

[88] F. Nouboud, R. Plamondon, "On-line recognition of handprinted characters: survey and beta tests," *Pattern Recognition*, vol. 23, no. 9, pp. 1031-1044, 1990.

[89] B. Griffin, "OCR Project Report: Keyfield Data Extraction for Document Imaging Systems," Unpublished technical report, Monash University, Churchill, 1994.

GLOSSARY

GLOSSARY

Term:	**Description:**
ADF	Automatic Document Feeder
ART	Accuracy, Resolution, Text size
ARTS	Accuracy, Resolution, Text size, Speed
ASC	Australian Securities Commission
ASCII	American Standard Code for Information Interchange
ASWEC	Australian SoftWare Engineering Conference
Bit	A single binary digit
Byte	Eight binary digits
CCITT	Comite Consultatif International de Telegraphie et Telephonie
CEDAR	Center of Excellence for Document Analysis and Recognition
CD	Compact Disc
CDR	Compact Disc - Recordable
CD ROM	Compact Disc - Read Only Memory
Character	A single letter, number, or symbol.
CITRI	Collaborative Information Technology Research Institute
CPS	Characters Per Second
DIAC	Digital Imaging Applications Center
DICTA	Digital Image Computing: Techniques and Applications
DOS	Disk Operating System
DPI	Dots Per Inch
DVD	Digital Versatile Disc
GB	GigaByte - 1000 million bytes
GUI	Graphical User Interface
HMM	Hidden Markov Model
HP	Hewlett Packard
HV	High Volume
ISO	Internation Standards Organisation

ISRI	Information Science Research Institute
MB	MegaByte - a million bytes
MHz	MegaHertz - a million cycles per second
MICR	Magnetic Ink Character Recognition
MO	Magneto Optical
MS	MicroSoft
MUGC	Monash University Gippsland Campus
NIPC	National Information Processing Centre
OCR	Optical Character Recognition
OPSO	OCR System Performance Optimiser
Point	1/72nd of an inch
POWR	Predictive Optical Word Recognition
PC	Personal Computer
RAM	Random Access Memory
RMIT	Royal Melbourne Institute of Technology
ROM	Read Only Memory
RTS	Resolution, Text size, Speed
SDAIR	Symposium on Document Analysis and Information Retrieval
WORM	Write Once Read Many

APPENDIX A

OSPO SOURCE CODE

```
VERSION 2.00
Begin Form AboutOSPO
   BackColor       =   &H00C0C0C0&
   BorderStyle     =   3  'Fixed Double
   Caption         =   "About OPSO"
   ClientHeight    =   3780
   ClientLeft      =   1905
   ClientTop       =   2730
   ClientWidth     =   7695
   Height          =   4185
   Left            =   1845
   LinkTopic       =   "Form1"
   ScaleHeight     =   3780
   ScaleWidth      =   7695
   Top             =   2385
   Width           =   7815
   Begin TextBox Text2
      Alignment       =   2  'Center
      BackColor       =   &H00C0C0C0&
      BorderStyle     =   0  'None
      Height          =   255
      Left            =   2400
      MultiLine       =   -1  'True
      TabIndex        =   3
      Text            =   "Copyright 1995 Brian Griffin"
      Top             =   2760
      Width           =   2775
   End
   Begin TextBox Text1
      Alignment       =   2  'Center
      BackColor       =   &H00C0C0C0&
      BorderStyle     =   0  'None
      Height          =   255
      Left            =   2880
      MultiLine       =   -1  'True
      TabIndex        =   4
      Text            =   "Version 1.0 beta"
      Top             =   2400
      Width           =   1695
   End
   Begin CommandButton Command1
      Caption         =   "OK"
      Default         =   -1  'True
      Height          =   375
      Left            =   3120
      TabIndex        =   0
      Top             =   3240
      Width            =   1215
   End
   Begin Label Label2
      Alignment        =   2  'Center
      BackColor        =   &H00C0C0C0&
      Caption          =   "OCR System Performance
Optimiser"
      FontBold         =   -1  'True
      FontItalic       =   -1  'True
      FontName         =   "Brush Script MT"
      FontSize         =   24
      FontStrikethru   =   0   'False
      FontUnderline    =   0   'False
      ForeColor        =   &H000000FF&
      Height           =   615
      Left             =   360
      TabIndex         =   2
      Top              =   1560
      Width            =   7095
   End
   Begin Label Label1
      Alignment        =   2  'Center
      BackColor        =   &H00C0C0C0&
      Caption          =   "OSPO"
      FontBold         =   -1  'True
      FontItalic       =   -1  'True
      FontName         =   "Brush Script MT"
      FontSize         =   60
      FontStrikethru   =   0   'False
      FontUnderline    =   0   'False
      ForeColor        =   &H000000FF&
      Height           =   1215
      Left             =   360
      TabIndex         =   1
      Top              =   240
      Width            =   7095
   End
End
Sub Command1_Click ()

     AboutOSPO.Hide

     OSPO.Show

End Sub
```

```
VERSION 2.00
Begin Form OSPO
   BackColor       =   &H00C0C0C0&
   Caption         =   "OSPO - OCR System Performance
Optimiser"
   ClientHeight    =   8685
   ClientLeft      =   1095
   ClientTop       =   1800
   ClientWidth     =   11010
   Height          =   9375
   Left            =   1035
   LinkTopic       =   "Form1"
   ScaleHeight     =   8685
   ScaleWidth      =   11010
   Top             =   1170
   Width           =   11130
   Begin SSFrame Frame3D3
      Caption         =   "Parameter Values"
      Font3D          =   0  'None
      ForeColor       =   &H00000000&
      Height          =   2175
      Left            =   240
      TabIndex        =   37
      Top             =   3960
      Width           =   3735
      Begin TextBox Text13
         Height          =   285
         Left            =   1680
         TabIndex        =   41
         Text            =   "*"
         Top             =   720
         Width           =   1815
      End
      Begin TextBox Text12
         Height          =   285
         Left            =   1680
         TabIndex        =   40
         Text            =   "200"
         Top             =   1080
         Width           =   1815
      End
      Begin TextBox Text11
         Height          =   285
         Left            =   1680
         TabIndex        =   39
         Text            =   "*"
      Top             =   1440
      Width           =   1815
   End
   Begin TextBox Text10
      Height          =   285
      Left            =   1680
      TabIndex        =   38
      Text            =   "30"
      Top             =   1800
      Width           =   1815
   End
   Begin Label Label23
      Alignment       =   2  'Center
      BackColor       =   &H00C0C0C0&
      Caption         =   "Value"
      FontBold        =   -1  'True
      FontItalic      =   0   'False
      FontName        =   "MS Sans Serif"
      FontSize        =   8.25
      FontStrikethru  =   0   'False
      FontUnderline   =   -1  'True
      Height          =   255
      Left            =   1560
      TabIndex        =   47
      Top             =   360
      Width           =   855
   End
   Begin Label Label15
      Alignment       =   1  'Right Justify
      BackColor       =   &H00C0C0C0&
      Caption         =   "Accuracy (%)"
      Height          =   255
      Left            =   120
      TabIndex        =   46
      Top             =   720
      Width           =   1455
   End
   Begin Label Label14
      Alignment       =   1  'Right Justify
      BackColor       =   &H00C0C0C0&
      Caption         =   "Resolution (DPI)"
      Height          =   255
      Left            =   120
      TabIndex        =   45
      Top             =   1080
      Width           =   1455
   End
```

```
      Begin Label Label13
         Alignment       =   1  'Right Justify
         BackColor       =   &H00C0C0C0&
         Caption         =   "Text Size (pts)"
         Height          =   255
         Left            =   120
         TabIndex        =   44
         Top             =   1440
         Width           =   1455
      End
      Begin Label Label12
         Alignment       =   1  'Right Justify
         BackColor       =   &H00C0C0C0&
         Caption         =   "Speed (cps)"
         Height          =   255
         Left            =   120
         TabIndex        =   43
         Top             =   1800
         Width           =   1455
      End
      Begin Label Label11
         Alignment       =   1  'Right Justify
         BackColor       =   &H00C0C0C0&
         Caption         =   "PARAMETER"
         FontBold        =   -1  'True
         FontItalic      =   0   'False
         FontName        =   "MS Sans Serif"
         FontSize        =   8.25
         FontStrikethru  =   0   'False
         FontUnderline   =   -1  'True
         Height          =   255
         Left            =   120
         TabIndex        =   42
         Top             =   360
         Width           =   1455
      End
   End
   Begin SSFrame Frame3D1
      Caption         =   "System Data"
      Font3D          =   0  'None
      Height          =   3735
      Left            =   240
      TabIndex        =   1
      Top             =   120
      Width           =   3735
      Begin TextBox Text9
         Height          =   285
   Left             =  1440
   TabIndex         =  36
   Text             =  "ARTS v1.32"
   Top              =  720
   Width            =  2055
End
Begin TextBox Text8
   Height           =  285
   Left             =  1680
   TabIndex         =  18
   Text             =  "*"
   Top              =  3240
   Width            =  1815
End
Begin TextBox Text7
   Height           =  285
   Left             =  1680
   TabIndex         =  17
   Text             =  "*"
   Top              =  2880
   Width            =  1815
End
Begin TextBox Text6
   Height           =  285
   Left             =  1680
   TabIndex         =  16
   Text             =  "*"
   Top              =  2520
   Width            =  1815
End
Begin TextBox Text5
   Height           =  285
   Left             =  1680
   TabIndex         =  15
   Text             =  ".0314"
   Top              =  2160
   Width            =  1815
End
Begin TextBox Text4
   Height           =  285
   Left             =  1680
   TabIndex         =  14
   Text             =  "80"
   Top              =  1800
   Width            =  1815
End
Begin TextBox Text3
```

```
      Height              =   285
      Left                =   1680
      TabIndex            =   13
      Text                =   "1800"
      Top                 =   1440
      Width               =   1815
   End
   Begin TextBox Text2
      Height              =   285
      Left                =   1680
      TabIndex            =   12
      Text                =   "49"
      Top                 =   1080
      Width               =   1815
   End
   Begin TextBox Text1
      Height              =   285
      Left                =   1440
      TabIndex            =   11
      Text                =   "FormReader 2"
      Top                 =   360
      Width               =   2055
   End
   Begin Label Label10
      Alignment           =   1  'Right Justify
      BackColor           =   &H00C0C0C0&
      Caption             =   "Model Name : "
      Height              =   255
      Left                =   120
      TabIndex            =   35
      Top                 =   720
      Width               =   1335
   End
   Begin Label Label9
      Alignment           =   2  'Center
      BackColor           =   &H00C0C0C0&
      Caption             =   "g"
      Height              =   255
      Left                =   1440
      TabIndex            =   10
      Top                 =   3240
      Width               =   255
   End
   Begin Label Label8
      Alignment           =   2  'Center
      BackColor           =   &H00C0C0C0&
      Caption             =   "f"
      Height             =    255
      Left               =    1440
      TabIndex           =    9
      Top                =    2880
      Width              =    255
   End
   Begin Label Label7
      Alignment          =    2  'Center
      BackColor          =    &H00C0C0C0&
      Caption            =    "e"
      Height             =    255
      Left               =    1440
      TabIndex           =    8
      Top                =    2520
      Width              =    255
   End
   Begin Label Label6
      Alignment          =    2  'Center
      BackColor          =    &H00C0C0C0&
      Caption            =    "d"
      Height             =    255
      Left               =    1440
      TabIndex           =    7
      Top                =    2160
      Width              =    255
   End
   Begin Label Label5
      Alignment          =    2  'Center
      BackColor          =    &H00C0C0C0&
      Caption            =    "c"
      Height             =    255
      Left               =    1440
      TabIndex           =    6
      Top                =    1800
      Width              =    255
   End
   Begin Label Label4
      Alignment          =    2  'Center
      BackColor          =    &H00C0C0C0&
      Caption            =    "b"
      Height             =    255
      Left               =    1440
      TabIndex           =    5
      Top                =    1440
      Width              =    255
   End
   Begin Label Label3
```

```
      Alignment       =   2  'Center
      BackColor       =   &H00C0C0C0&
      Caption         =   "a"
      Height          =   255
      Left            =   1440
      TabIndex        =   4
      Top             =   1080
      Width           =   255
   End
   Begin Label Label2
      Alignment       =   1  'Right Justify
      BackColor       =   &H00C0C0C0&
      Caption         =   "Co-efficients : "
      Height          =   255
      Left            =   120
      TabIndex        =   3
      Top             =   1080
      Width           =   1335
   End
   Begin Label Label1
      Alignment       =   1  'Right Justify
      BackColor       =   &H00C0C0C0&
      Caption         =   "System Name : "
      Height          =   255
      Left            =   120
      TabIndex        =   2
      Top             =   360
      Width           =   1335
   End
End
Begin GRAPH Graph1
   AsciiData       =   "1"
   AsciiFFamily    =   "1~1~1~1"
   AsciiXPos       =   "4"
   AutoInc         =   0  'Off
   BottomTitle     =   "Text Size (pts)"
   GraphCaption    =   "System Performance Plot"
   GraphTitle      =   "System Performance Plot"
   GraphType       =   6  'Line
   GridStyle       =   3  'Horizontal and Vertical
   Height          =   8415
   Left            =   4200
   LeftTitle       =   "Accuracy (%)"
   NumPoints       =   21
   RandomData      =   0  'Off
   TabIndex        =   0
   Top             =   120
   Width              =   6615
   YAxisMax           =   100
   YAxisPos           =   1  'Left
   YAxisStyle         =   2  'User-defined
   YAxisTicks         =   10
End
Begin CommonDialog CMDialog1
   Left               =   600
   Top                =   2760
End
Begin SSFrame Frame3D2
   Caption            =   "Plot Parameter Ranges"
   Font3D             =   0  'None
   ForeColor          =   &H00000000&
   Height             =   2295
   Left               =   240
   TabIndex           =   26
   Top                =   6240
   Width              =   3735
   Begin TextBox Text24
      Height             =   285
      Left               =   2640
      TabIndex           =   19
      Text               =   "1000"
      Top                =   1800
      Width              =   855
   End
   Begin TextBox Text23
      Height             =   285
      Left               =   2640
      TabIndex           =   20
      Text               =   "24"
      Top                =   1440
      Width              =   855
   End
   Begin TextBox Text22
      Height             =   285
      Left               =   2640
      TabIndex           =   21
      Text               =   "400"
      Top                =   1080
      Width              =   855
   End
   Begin TextBox Text21
      Height             =   285
      Left               =   2640
      TabIndex           =   22
      Text            =   "100"
      Top             =   720
      Width           =   855
   End
   Begin TextBox Text20
      Height          =   285
      Left            =   1680
      TabIndex        =   23
      Text            =   "10"
      Top             =   1800
      Width           =   855
   End
   Begin TextBox Text19
      Height          =   285
      Left            =   1680
      TabIndex        =   24
      Text            =   "4"
      Top             =   1440
      Width           =   855
   End
   Begin TextBox Text18
      Height          =   285
      Left            =   1680
      TabIndex        =   25
      Text            =   "100"
      Top             =   1080
      Width           =   855
   End
   Begin TextBox Text17
      Height          =   285
      Left            =   1680
      TabIndex        =   28
      Text            =   "0"
      Top             =   720
      Width           =   855
   End
   Begin Label Label22
      Alignment       =   1  'Right Justify
      BackColor       =   &H00C0C0C0&
      Caption         =   "PARAMETER"
      FontBold        =   -1  'True
      FontItalic      =   0   'False
      FontName        =   "MS Sans Serif"
      FontSize        =   8.25
      FontStrikethru  =   0   'False
      FontUnderline   =   -1  'True
      Height          =   255
      Left            =   120
      TabIndex        =   34
      Top             =   360
      Width           =   1455
   End
   Begin Label Label21
      Alignment       =   1  'Right Justify
      BackColor       =   &H00C0C0C0&
      Caption         =   "Speed (cps)"
      Height          =   255
      Left            =   120
      TabIndex        =   33
      Top             =   1800
      Width           =   1455
   End
   Begin Label Label20
      Alignment       =   1  'Right Justify
      BackColor       =   &H00C0C0C0&
      Caption         =   "Text Size (pts)"
      Height          =   255
      Left            =   120
      TabIndex        =   32
      Top             =   1440
      Width           =   1455
   End
   Begin Label Label19
      Alignment       =   1  'Right Justify
      BackColor       =   &H00C0C0C0&
      Caption         =   "Resolution (DPI)"
      Height          =   255
      Left            =   120
      TabIndex        =   31
      Top             =   1080
      Width           =   1455
   End
   Begin Label Label18
      Alignment       =   1  'Right Justify
      BackColor       =   &H00C0C0C0&
      Caption         =   "Accuracy (%)"
      Height          =   255
      Left            =   120
      TabIndex        =   30
      Top             =   720
      Width           =   1455
   End
   Begin Label Label17
      Alignment       =   2  'Center
      BackColor         =   &H00C0C0C0&
      Caption           =   "Minimum"
      FontBold          =   -1  'True
      FontItalic        =   0   'False
      FontName          =   "MS Sans Serif"
      FontSize          =   8.25
      FontStrikethru    =   0   'False
      FontUnderline     =   -1  'True
      Height            =   255
      Left              =   1680
      TabIndex          =   29
      Top               =   360
      Width             =   855
   End
   Begin Label Label16
      Alignment         =   2  'Center
      BackColor         =   &H00C0C0C0&
      Caption           =   "Maximum"
      FontBold          =   -1  'True
      FontItalic        =   0   'False
      FontName          =   "MS Sans Serif"
      FontSize          =   8.25
      FontStrikethru    =   0   'False
      FontUnderline     =   -1  'True
      Height            =   255
      Index             =   1
      Left              =   2640
      TabIndex          =   27
      Top               =   360
      Width             =   855
   End
End
Begin Menu File
   Caption         =   "&File"
   Begin Menu New
      Caption           =   "&New System"
   End
   Begin Menu Open
      Caption           =   "&Open System..."
   End
   Begin Menu Save
      Caption           =   "Save System"
   End
   Begin Menu SaveAs
      Caption           =   "Save System As..."
   End
   Begin Menu Null2
```

```
         Caption             =  "-"
      End
      Begin Menu Print
         Caption             =  "&Print"
      End
      Begin Menu Null1
         Caption             =  "-"
      End
      Begin Menu Exit
         Caption             =  "E&xit"
      End
   End
   Begin Menu Options
      Caption             =  "&Options"
      Begin Menu PlotDimensions
         Caption             =  "&Plot Dimensions"
      End
   End
   Begin Menu Help
      Caption             =  "&Help"
      Begin Menu Contents
         Caption             =  "&Contents"
      End
      Begin Menu Search
         Caption             =  "&Search For Help On..."
      End
      Begin Menu About
         Caption             =  "&About OSPO"
      End
   End
End
Sub About_Click ()
    AboutOSPO.Show
End Sub

Sub Contents_Click ()
    CMDialog1.Action = 6
End Sub

Sub Exit_Click ()
    End
End Sub

Sub Form_Load ()

    OSPO.Hide
```

```
Graph1.ThisPoint = 1
Graph1.GraphData = 2
Graph1.XPosData = 4

Graph1.ThisPoint = 2
Graph1.GraphData = 2
Graph1.XPosData = 5

Graph1.ThisPoint = 3
Graph1.GraphData = 3
Graph1.XPosData = 6

Graph1.ThisPoint = 4
Graph1.GraphData = 5
Graph1.XPosData = 7

Graph1.ThisPoint = 5
Graph1.GraphData = 11
Graph1.XPosData = 8

Graph1.ThisPoint = 6
Graph1.GraphData = 49
Graph1.XPosData = 9

Graph1.ThisPoint = 7
Graph1.GraphData = 86
Graph1.XPosData = 10

Graph1.ThisPoint = 8
Graph1.GraphData = 93
Graph1.XPosData = 11

Graph1.ThisPoint = 9
Graph1.GraphData = 97
Graph1.XPosData = 12

Graph1.ThisPoint = 10
Graph1.GraphData = 98
Graph1.XPosData = 13

Graph1.ThisPoint = 11
Graph1.GraphData = 99
Graph1.XPosData = 14

Graph1.ThisPoint = 12
```

```
        Graph1.GraphData = 99
        Graph1.XPosData = 15

        Graph1.ThisPoint = 13
        Graph1.GraphData = 99
        Graph1.XPosData = 16

        Graph1.ThisPoint = 14
        Graph1.GraphData = 99
        Graph1.XPosData = 17

        Graph1.ThisPoint = 15
        Graph1.GraphData = 99
        Graph1.XPosData = 18

        Graph1.ThisPoint = 16
        Graph1.GraphData = 99
        Graph1.XPosData = 19

        Graph1.ThisPoint = 17
        Graph1.GraphData = 99
        Graph1.XPosData = 20

        Graph1.ThisPoint = 18
        Graph1.GraphData = 99
        Graph1.XPosData = 21

        Graph1.ThisPoint = 19
        Graph1.GraphData = 99
        Graph1.XPosData = 22

        Graph1.ThisPoint = 20
        Graph1.GraphData = 99
        Graph1.XPosData = 23

        Graph1.ThisPoint = 21
        Graph1.GraphData = 99
        Graph1.XPosData = 24


        AboutOSPO.Show

    End Sub

    Sub Open_Click ()
        CMDialog1.Action = 1
    End Sub
```

```
Sub Print_Click ()
    CMDialog1.Action = 5
End Sub

Sub SaveAs_Click ()
    CMDialog1.Action = 2
End Sub
```

APPENDIX B

FORMREADER SOURCE CODE

```
'FormReader v3.0 beta
'Copywrite 1992-1996 Brian Griffin

'Global variable declarations

Global SoundData As String

'External function declarations

Declare Function Say Lib "C:\MONOLOGW\FB_SPCH.DLL" (ByVal
lpEnglishString As String) As Integer


VERSION 2.00
Begin Form About
   BorderStyle     =   3  'Fixed Double
   Caption         =   "About Form Reader"
   ClientHeight    =   3915
   ClientLeft      =   1650
   ClientTop       =   2460
   ClientWidth     =   6240
   Height          =   4320
   Left            =   1590
   LinkTopic       =   "Form2"
   MaxButton       =   0   'False
   MinButton       =   0   'False
   ScaleHeight     =   3915
   ScaleWidth      =   6240
   Top             =   2115
   Width           =   6360
   Begin CommandButton Command1
      Caption         =   "OK"
      Default         =   -1  'True
      Height          =   375
      Left            =   2520
      TabIndex        =   4
      Top             =   3360
      Width           =   1215
   End
   Begin PictureBox Picture1
      AutoSize        =   -1  'True
      BorderStyle     =   0  'None
      Height          =   2310
      Left            =   120
      Picture         =   ABOUT.FRX:0000
      ScaleHeight     =   2310
      ScaleWidth      =   5985
      TabIndex        =   0
      Top             =   120
      Width           =   5985
   End
   Begin Label Label3
      Alignment       =   2  'Center
      Caption         =   "Version 3.0 beta"
      Height          =   255
      Left            =   120
      TabIndex        =   3
      Top             =   2760
      Width           =   5895
   End
   Begin Label Label2
      Alignment       =   2  'Center
      Caption         =   "Form Reader"
      Height          =   255
      Left            =   120
      TabIndex        =   2
      Top             =   2520
      Width           =   5895
   End
   Begin Label Label1
      Alignment       =   2  'Center
      Caption         =   "Copyright 1992 - 1996 Brian
Griffin."
      Height          =   255
      Left            =   120
      TabIndex        =   1
      Top             =   3000
      Width           =   5895
   End
End
Sub Command1_Click ()
    About.Hide
End Sub

Sub Form_Load ()

    'Initialise sound system

    FormRead.MMControl1.Notify = False
    FormRead.MMControl1.Wait = True
    FormRead.MMControl1.Shareable = False
    FormRead.MMControl1.DeviceType = "WaveAudio"

    'Play FormReader opening sound file
    FormRead.MMControl1.Command = "Close"
    FormRead.MMControl1.FileName =
"C:\VB\FORMRED3\SOUNDS\FORMREAD.WAV"
    FormRead.MMControl1.Command = "Open"
    FormRead.MMControl1.Command = "play"

End Sub


VERSION 2.00
Begin Form FormRead
   Caption         =   "FormReader 3.0"
   ClientHeight    =   4170
   ClientLeft      =   1680
   ClientTop       =   2340
   ClientWidth     =   5910
   Height          =   4860
   Left            =   1620
   LinkTopic       =   "Form1"
   ScaleHeight     =   4170
   ScaleWidth      =   5910
   Top             =   1710
   Width           =   6030
   Begin TextBox Text4
      DataField       =   "ACN"
      DataSource      =   "Data1"
      Height          =   375
      Left            =   3480
      TabIndex        =   3
      Text            =   "Text4"
      Top             =   1080
      Width           =   1695
   End
   Begin TextBox Text3
      DataField       =   "Company Name"
      DataSource      =   "Data1"
      Height          =   375
      Left            =   3480
      ScrollBars      =   1  'Horizontal
      TabIndex        =   4
      Text            =   "Text3"
      Top             =   480
      Width           =   1695
   End
   Begin Data Data1
      Caption         =   "Data1"
   Connect         =   ""
   DatabaseName    =   "C:\VB\FORMRED3\ASCBASE1.MDB"
   Exclusive       =   0   'False
   Height          =   270
   Left            =   1800
   Options         =   0
   ReadOnly        =   -1  'True
   RecordSource    =   "CoData"
   Top             =   2760
   Width           =   2775
End
Begin TextBox Text2
   Height          =   375
   Left            =   600
   TabIndex        =   2
   Text            =   "Text2"
   Top             =   1080
   Width           =   2295
End
Begin TextBox Text1
   Height          =   375
   Left            =   600
   TabIndex        =   1
   Text            =   "Text1"
   Top             =   480
   Width           =   2295
End
Begin MMControl MMControl1
   Height          =   375
   Left            =   600
   TabIndex        =   0
   Top             =   3360
   Visible         =   0   'False
   Width           =   3540
End
Begin CommonDialog CMDialog1
   Left            =   480
   Top             =   2640
End
Begin Menu Menu_File
   Caption         =   "&File"
   Begin Menu Menu_Open
      Caption         =   "&Open..."
   End
   Begin Menu Menu_Save
      Caption         =   "&Save..."
   End
```

```
      Begin Menu Menu_Null1
         Caption          =   "-"
      End
      Begin Menu Menu_Print
         Caption          =   "&Print..."
      End
      Begin Menu Menu_Null2
         Caption          =   "-"
      End
      Begin Menu Menu_Exit
         Caption          =   "E&xit"
      End
   End
   Begin Menu Menu_Options
      Caption          =   "&Options"
      Begin Menu Menu_Sound
         Caption          =   "&Sound..."
      End
      Begin Menu Menu_Paths
         Caption          =   "&Path..."
      End
   End
   Begin Menu Menu_Help
      Caption          =   "&Help"
      Begin Menu Menu_About
         Caption          =   "&About..."
      End
   End
End

Sub Form_Load ()

    'About.Show 1

    'Initialise OmniPage
    ChDir "C:\OPRO"
    z% = Shell("C:\OPRO\OP -DDE", 1)
    z% = DoEvents()
    Text2.LinkMode = NONE
    Text2.LinkTopic = "OmniPro|OmniPro"
    Text2.LinkItem = ""
    Text2.LinkMode = MANUAL

End Sub

Sub Menu_About_Click ()
```

```
    About.Show 1

End Sub

Sub Menu_Exit_Click ()

    End

End Sub

Sub Menu_Open_Click ()

    CMDialog1.Filter = "Image file (*.tif)|*.tif|Template
(*.tmp)|*.tmp|Database (*.mdb)|*.mdb|Configuration
(*.con)|*.con|All files (*.*)|*.*"
    ChDir "C:\VB\FormRed3"

    CMDialog1.Action = 1

    If Right$(CMDialog1.Filename, 3) = "TIF" Then
        z% = Shell("C:\ImagePac\ImagePac
/sDocumentDisplay", 1)
        z% = DoEvents()
        Text1.LinkMode = 0
        Text1.LinkTopic = "ImagePac|DocumentDisplay"
        Text1.LinkItem = ""
        Text1.LinkMode = 2
        Text1.LinkExecute "[display x " +
CMDialog1.Filename
    End If

End Sub

Sub Menu_Paths_Click ()

    PathOpt.Show 0

End Sub

Sub Menu_Print_Click ()

    CMDialog1.Action = 5

End Sub

Sub Menu_Save_Click ()
```

```
    CMDialog1.Action = 2

End Sub

Sub MMControl1_Done (NotifyCode As Integer)

        FormRead.MMControl1.Command = "Close"
        If Len(SoundData) > 0 Then
            Char = Left$(SoundData, 1)
            SoundData = Right$(SoundData, Len(SoundData)
- 1)
            If Char = "." Then Char = "point"
            FormRead.MMControl1.FileName =
"C:\VB\FORMREAD\SOUNDS\" + Char + ".WAV"
            FormRead.MMControl1.Command = "Open"
            FormRead.MMControl1.Command = "play"
        End If

End Sub

Sub Sayx (SoundDataAdd As String)

    SoundData = SoundData + SoundDataAdd

    If Len(SoundData) = Len(SoundDataAdd) Then
        Char = Left$(SoundData, 1)
        SoundData = Right$(SoundData, Len(SoundData) - 1)
        If Char = "." Then Char = "POINT"
        FormRead.MMControl1.FileName =
"C:\VB\FORMRED3\SOUNDS\" + Char + ".WAV"
        FormRead.MMControl1.Command = "Open"
        FormRead.MMControl1.Command = "Play"
    End If

End Sub

Sub Text1_KeyPress (KeyAscii As Integer)

    If KeyAscii = 13 Then
        KeyAscii = 0
        If Text1.Text <> "" Then z% = Say(Text1.Text)
    End If

End Sub

Sub Text2_DblClick ()
```

```
    'OCR Document

    Text2.LinkExecute "[imagesource = (1)]"
    Text2.LinkExecute "[imagefiles(" + CMDialog1.Filename
+ ")]"
    Text2.LinkExecute "[setform(ASCII)]"
    Text2.LinkExecute "[setoutput (C:\VB\OCRTEST.TXT)]"
    Text2.LinkExecute "[scan(0)]"

End Sub


VERSION 2.00
Begin Form PathOpt
   BackColor       =   &H00C0C0C0&
   BorderStyle     =   3  'Fixed Double
   Caption         =   "Path Options"
   ClientHeight    =   2385
   ClientLeft      =   2025
   ClientTop       =   3090
   ClientWidth     =   5760
   Height          =   2790
   Left            =   1965
   LinkTopic       =   "Form1"
   MaxButton       =   0   'False
   MinButton       =   0   'False
   ScaleHeight     =   2385
   ScaleWidth      =   5760
   Top             =   2745
   Width           =   5880
   Begin TextBox ImageSource
      Height          =   285
      Left            =   2160
      TabIndex        =   11
      Top             =   120
      Width           =   3495
   End
   Begin TextBox DBaseDest
      Height          =   285
      Left            =   2160
      TabIndex        =   10
      Top             =   1560
      Width           =   3495
   End
   Begin TextBox DBaseSource
      Height          =   285
      Left            =   2160
   TabIndex        =   9
   Top             =   1200
   Width           =   3495
End
Begin TextBox TextDest
   Height          =   285
   Left            =   2160
   TabIndex        =   8
   Top             =   840
   Width           =   3495
End
Begin TextBox ImageDest
   Height          =   285
   Left            =   2160
   TabIndex        =   7
   Top             =   480
   Width           =   3495
End
Begin CommandButton OK
   Caption         =   "OK"
   Default         =   -1  'True
   Height          =   375
   Left            =   4440
   TabIndex        =   1
   Top             =   1920
   Width           =   1215
End
Begin CommandButton Cancel
   Caption         =   "Cancel"
   Height          =   375
   Left            =   120
   TabIndex        =   0
   Top             =   1920
   Width           =   1215
End
Begin Label Label5
   Alignment       =   1  'Right Justify
   BackColor       =   &H00C0C0C0&
   Caption         =   "Database Destination"
   Height          =   255
   Left            =   120
   TabIndex        =   6
   Top             =   1560
   Width           =   1935
End
Begin Label Label4
   Alignment       =   1  'Right Justify
```

```
         BackColor        =    &H00C0C0C0&
         Caption          =    "Database Source"
         Height           =    255
         Left             =    120
         TabIndex         =    5
         Top              =    1200
         Width            =    1935
      End
      Begin Label Label3
         Alignment        =    1  'Right Justify
         BackColor        =    &H00C0C0C0&
         Caption          =    "Text Destination"
         Height           =    255
         Left             =    120
         TabIndex         =    4
         Top              =    840
         Width            =    1935
      End
      Begin Label Label2
         Alignment        =    1  'Right Justify
         BackColor        =    &H00C0C0C0&
         Caption          =    "Image Destination"
         Height           =    255
         Left             =    120
         TabIndex         =    3
         Top              =    480
         Width            =    1935
      End
      Begin Label Label1
         Alignment        =    1  'Right Justify
         BackColor        =    &H00C0C0C0&
         Caption          =    "Image Source"
         Height           =    255
         Left             =    120
         TabIndex         =    2
         Top              =    120
         Width            =    1935
      End
   End
   Sub Cancel_Click ()
       PathOpt.Hide
   End Sub

   Sub OK_Click ()
       ImageSourceDir = ImageSource.Text
       ImageDestDir = ImageDest.Text
       TextDestDir = TextDest.Text
```

```
    DBaseSourceDir = DBaseSource.Text
    DBaseDestDir = DBaseDest.Text
    PathOpt.Hide
End Sub
```

APPENDIX C

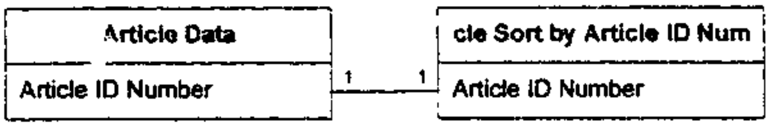
LITBASE SOURCE CODE

Columns

Name	Type	Size
Article ID Number	Number (Double)	8
Author(s)	Text	100
Title	Text	100
Publication	Text	100
Volume & pp	Text	100
Year	Number (Double)	8
Key Words	Text	255
Abstract	Memo	-
Image Available	Yes/No	1
Text Available	Yes/No	1
User 1	Text	50
User 2	Text	50
User 3	Text	50
User 4	Number (Double)	8
User 5	Number (Double)	8
User 6	Number (Double)	8
User 7	Yes/No	1
User 8	Yes/No	1
User 9	Yes/No	1
User 10	Memo	-

Relationships

Reference



Attributes: One to One, Not Enforced

Table Indexes

Name	Number of Fields
PrimaryKey	1

Fields: Article ID Number, Ascending

Columns

Name	Type	Size
ID	Number (Long)	4
User Field 1 Name	Text	50
User Field 2 Name	Text	50
User Field 3 Name	Text	50
User Field 4 Name	Text	50
User Field 5 Name	Text	50
User Field 6 Name	Text	50
User Field 7 Name	Text	50
User Field 8 Name	Text	50
User Field 9 Name	Text	50
User Field 10 Name	Text	50

Table Indexes

Name	Number of Fields
PrimaryKey	1
Fields:	ID, Ascending

Columns

Name	Type	Size
ID	Number (Long)	4
User Field 1	Yes/No	1
User Field 2	Yes/No	1
User Field 3	Yes/No	1
User Field 4	Yes/No	1
User Field 5	Text	255
User Field 6	Text	50
User Field 7	Text	50
User Field 8	Text	50
User Field 9	Number (Double)	8
User Field 10	Number (Double)	8

Table Indexes

Name	Number of Fields
PrimaryKey	1
Fields:	ID, Ascending

SQL

```
SELECT DISTINCTROW [Article Data].Title, [Article Data].[Author(s)], [Article Data].Publication, [Article
Data].[Volume & pp], [Article Data].Year, [Article Data].[Article ID Number]
FROM [Article Data]
ORDER BY [Article Data].Title, [Article Data].[Author(s)];
```

Columns

Name	Type	Size
Title	Text	100
Author(s)	Text	100
Publication	Text	100
Volume & pp	Text	100
Year	Number (Double)	8
Article ID Number	Number (Double)	8

SQL

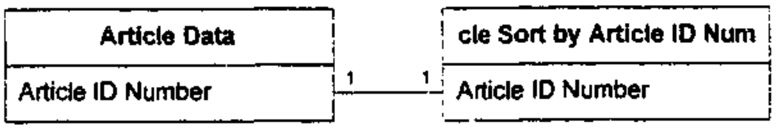
```
SELECT DISTINCTROW [Article Data].Title, [Article Data].[Author(s)], [Article Data].Publication, [Article
Data].[Volume & pp], [Article Data].Year, [Article Data].[Article ID Number]
FROM [Article Data]
ORDER BY [Article Data].[Article ID Number];
```

Columns

Name	Type	Size
Title	Text	100
Author(s)	Text	100
Publication	Text	100
Volume & pp	Text	100
Year	Number (Double)	8
Article ID Number	Number (Double)	8

Relationships

Reference



Attributes: One to One, Not Enforced

SQL

```
SELECT DISTINCTROW [Article Data].Title, [Article Data].[Author(s)], [Article Data].Publication, [Article
Data].[Volume & pp], [Article Data].Year, [Article Data].[Article ID Number]
FROM [Article Data]
ORDER BY [Article Data].[Author(s)];
```

Columns

Name	Type	Size
Title	Text	100
Author(s)	Text	100
Publication	Text	100
Volume & pp	Text	100
Year	Number (Double)	8
Article ID Number	Number (Double)	8

C:\WINAPPS\ACCESS\DOCS\LITBASE2.MDB
Query: Article Sort by Publication

Friday, 28 February 1997
Page: 1

SQL

```
SELECT DISTINCTROW [Article Data].Title, [Article Data].[Author(s)], [Article Data].Publication, [Article
Data].[Volume & pp], [Article Data].Year, [Article Data].[Article ID Number]
FROM [Article Data]
ORDER BY [Article Data].Publication;
```

Columns

Name	Type	Size
Title	Text	100
Author(s)	Text	100
Publication	Text	100
Volume & pp	Text	100
Year	Number (Double)	8
Article ID Number	Number (Double)	8

SQL

```
SELECT DISTINCTROW [Article Data].Title, [Article Data].[Author(s)], [Article Data].Publication, [Article
Data].[Volume & pp], [Article Data].Year, [Article Data].[Article ID Number]
FROM [Article Data]
ORDER BY [Article Data].Title, [Article Data].[Author(s)];
```

Columns

Name	Type	Size
Title	Text	100
Author(s)	Text	100
Publication	Text	100
Volume & pp	Text	100
Year	Number (Double)	8
Article ID Number	Number (Double)	8

SQL

```
SELECT DISTINCTROW [Article Data].Title, [Article Data].[Author(s)], [Article Data].Publication, [Article
Data].[Volume & pp], [Article Data].Year, [Article Data].[Article ID Number]
FROM [Article Data]
ORDER BY [Article Data].Year DESC;
```

Columns

Name	Type	Size
Title	Text	100
Author(s)	Text	100
Publication	Text	100
Volume & pp	Text	100
Year	Number (Double)	8
Article ID Number	Number (Double)	8

Objects

Section: Detail0

Section: FormFooter2

Section: FormHeader1

Command Button: AboutOKButton

Label: Text12

Label: Text13

Label: Text35

Label: Text38

Label: Text39

Code

```
Option Compare Database   'Use database order for string comparisons

Sub AboutOKButton_Click ()
On Error GoTo Err_AboutOKButton_Click


    DoCmd Close

Exit_AboutOKButton_Click:
    Exit Sub

Err_AboutOKButton_Click:
    MsgBox Error$
    Resume Exit_AboutOKButton_Click

End Sub

```

Objects

Section: Detail0

Section: FormFooter2

Section: FormHeader1

Text Box: Abstract

Text Box: Article ID Number

Text Box: Author(s)

Command Button: Button39

Command Button: Button40

Command Button: Button41

Command Button: Button42

Command Button: Button44

Option Group: Field46

Text Box: Field64

Text Box: Field66

Text Box: Field68

Text Box: Field70

Text Box: Field72

Text Box: Field74

Check Box: Field77

Check Box: Field81

Check Box: Field83

Text Box: Field85

Text Box: Field87

Check Box: Image Available

Text Box: Key Words

Command Button: OCRArticleButton

Text Box: Publication

Command Button: ReturnButton

Check Box: Text Available

Label: Text12

Label: Text13

Label: Text15

Label: Text17

Label: Text19

Label: Text21

Label: Text23

Label: Text25

Label: Text27

Label: Text29

Label: Text47

Label: Text53

Label: Text55

Label: Text65

Label: Text67

Label: Text69

Label: Text71

Label: Text73

Label: Text75

Label: Text78

Label: Text82

Label: Text84

Label: Text86

Text Box: Title

Text Box: Volume & pp

Text Box: Year

Code

```
Option Compare Database   'Use database order for string comparisons

Sub Button37_Click ()
    MsgBox "Scanner not connected", 0, "Error"
End Sub

```

```
Sub Button39_Click ()
On Error GoTo Err_Button39_Click


    DoCmd GoToRecord , , A_FIRST

Exit_Button39_Click:
    Exit Sub

Err_Button39_Click:
    MsgBox Error$
    Resume Exit_Button39_Click

End Sub

Sub Button40_Click ()
On Error GoTo Err_Button40_Click


    DoCmd GoToRecord , , A_PREVIOUS

Exit_Button40_Click:
    Exit Sub

Err_Button40_Click:
    MsgBox Error$
    Resume Exit_Button40_Click

End Sub

Sub Button41_Click ()
On Error GoTo Err_Button41_Click


    DoCmd GoToRecord , , A_NEXT

Exit_Button41_Click:
    Exit Sub

Err_Button41_Click:
    MsgBox Error$
    Resume Exit_Button41_Click

End Sub

Sub Button42_Click ()
On Error GoTo Err_Button42_Click


    DoCmd GoToRecord , , A_LAST

Exit_Button42_Click:
    Exit Sub

Err_Button42_Click:
    MsgBox Error$
    Resume Exit_Button42_Click

End Sub

Sub Button43_Click ()
On Error GoTo Err_Button43_Click


    Screen.PreviousControl.SetFocus
    DoCmd FindNext

Exit_Button43_Click:
    Exit Sub

Err_Button43_Click:
    MsgBox Error$
    Resume Exit_Button43_Click

End Sub

Sub Button44_Click ()
On Error GoTo Err_Button44_Click


    DoCmd DoMenuItem A_FORMBAR, A_EDITMENU, 10, , A_MENU_VER20

Exit_Button44_Click:
    Exit Sub

Err_Button44_Click:
    MsgBox Error$
    Resume Exit_Button44_Click

End Sub

Sub Field34_Click ()
    MsgBox "Scanner not connected", 0, "Error"
End Sub

Sub Field35_Click ()

End Sub

Sub OCRArticleButton_Click ()
On Error GoTo Err_OCRArticleButton_Click

    Dim x As Integer
    Dim AppName As String

    AppName = "C:\WINAPPS\OMNIPRO\OMNIPAGE.EXE"
    x = Shell(AppName, 1)

Exit_OCRArticleButton_Click:
    Exit Sub
```

```

Err_OCRArticleButton_Click:
    MsgBox Error$
    Resume Exit_OCRArticleButton_Click

End Sub

Sub ReturnButton_Click ()
On Error GoTo Err_ReturnButton_Click


    DoCmd Close

Exit_ReturnButton_Click:
    Exit Sub

Err_ReturnButton_Click:
    MsgBox Error$
    Resume Exit_ReturnButton_Click

End Sub

```

Objects

Section: Detail0

Section: FormFooter2

Section: FormHeader1

Text Box: Abstract

Text Box: Article ID Number

Text Box: Author(s)

Command Button: Button41

Command Button: Button42

Command Button: Button43

Command Button: Button44

Command Button: Button45

Object Frame: Embedded49

Check Box: Field58

Check Box: Field60

Command Button: FullTextSearchButton

Label: ImageAvailable

Command Button: ImageDisplayButton

Text Box: Key Words

Command Button: PrintArticleButton

Text Box: Publication

Option Group: Record Operations

Command Button: ReturnButton2

Label: Text12

Label: Text13

Label: Text15

Label: Text17

Label: Text19

Label: Text21

Label: Text23

Label: Text25

Label: Text27

Label: Text29

Label: Text47

Label: TextAvailable

Command Button: TextDisplayButton

Text Box: Title

Text Box: Volume & pp

Text Box: Year

Code

```
Option Compare Database   'Use database order for string comparisons

Sub Button41_Click ()
On Error GoTo Err_Button41_Click


    DoCmd GoToRecord , , A_FIRST

Exit_Button41_Click:
    Exit Sub

Err_Button41_Click:
    MsgBox Error$
    Resume Exit_Button41_Click

End Sub

Sub Button42_Click ()
On Error GoTo Err_Button42_Click


    DoCmd GoToRecord , , A_PREVIOUS

Exit_Button42_Click:
    Exit Sub

Err_Button42_Click:
    MsgBox Error$
    Resume Exit_Button42_Click

End Sub

Sub Button43_Click ()
On Error GoTo Err_Button43_Click


```

```
    DoCmd GoToRecord , , A_NEXT

Exit_Button43_Click:
    Exit Sub

Err_Button43_Click:
    MsgBox Error$
    Resume Exit_Button43_Click

End Sub

Sub Button44_Click ()
On Error GoTo Err_Button44_Click


    DoCmd GoToRecord , , A_LAST

Exit_Button44_Click:
    Exit Sub

Err_Button44_Click:
    MsgBox Error$
    Resume Exit_Button44_Click

End Sub

Sub Button45_Click ()
On Error GoTo Err_Button45_Click


    DoCmd DoMenuItem A_FORMBAR, A_EDITMENU, 10, , A_MENU_VER20

Exit_Button45_Click:
    Exit Sub

Err_Button45_Click:
    MsgBox Error$
    Resume Exit_Button45_Click

End Sub

Sub Button51_Click ()
On Error GoTo Err_Button51_Click


    DoCmd Print

Exit_Button51_Click:
    Exit Sub

Err_Button51_Click:
    MsgBox Error$
    Resume Exit_Button51_Click

End Sub
```

```

Sub FullTextSearchButton_Click ()
On Error GoTo Err_FullTextSearchButton_Click

    Dim x As Integer
    Dim AppName As String

    AppName = "C:\WINAPPS\ISYS\IQW.EXE /Z"
    x = Shell(AppName, 1)

    SendKeys "%q"

Exit_FullTextSearchButton_Click:
    Exit Sub

Err_FullTextSearchButton_Click:
    MsgBox Error$
    Resume Exit_FullTextSearchButton_Click

End Sub

Sub ImageDisplayButton_Click ()
On Error GoTo Err_ImageDisplayButton_Click

    Dim x As Integer
    Dim AppName As String

    If [Image Available] Then
        AppName = "C:\WINAPPS\IMAGEPAC\IMAGEPAC.EXE /z"
        x = Shell(AppName, 1)
        SendKeys "%fo"
        If [Article ID Number] < 10 Then
            SendKeys "00" + LTrim$(Str$([Article ID Number])) + "p001.tif"
        ElseIf [Article ID Number] < 100 Then
            SendKeys "0" + LTrim$(Str$([Article ID Number])) + "p001.tif"
        ElseIf [Article ID Number] < 1000 Then
            SendKeys LTrim$(Str$([Article ID Number])) + "p001.tif"
        End If
        SendKeys "~"
    Else
        MsgBox "Image not available", 0, "LitBase"
    End If


Exit_ImageDisplayButton_Click:
    Exit Sub

Err_ImageDisplayButton_Click:
    MsgBox Error$
    Resume Exit_ImageDisplayButton_Click

End Sub

Sub PrintArticleButton_Click ()
On Error GoTo Err_PrintArticleButton_Click
```

```


    DoCmd DoMenuItem A_FORMBAR, A_EDITMENU, A_SELECTRECORD_V2, , A_MENU_VER20
    DoCmd Print A_SELECTION

Exit_PrintArticleButton_Click:
    Exit Sub

Err_PrintArticleButton_Click:
    MsgBox Error$
    Resume Exit_PrintArticleButton_Click

End Sub

Sub PrintArticleDataButt_Click ()
On Error GoTo Err_PrintArticleDataButt_Click


    DoCmd Print

Exit_PrintArticleDataButt_Click:
    Exit Sub

Err_PrintArticleDataButt_Click:
    MsgBox Error$
    Resume Exit_PrintArticleDataButt_Click

End Sub

Sub ReturnButton2_Click ()
On Error GoTo Err_ReturnButton2_Click


    DoCmd Close

Exit_ReturnButton2_Click:
    Exit Sub

Err_ReturnButton2_Click:
    MsgBox Error$
    Resume Exit_ReturnButton2_Click

End Sub

Sub TextDisplayButton_Click ()
On Error GoTo Err_TextDisplayButton_Click

    Dim x As Integer

    If [Text Available] Then
        If [Article ID Number] < 10 Then
            x = Shell("Write.EXE c:\winapps\access\docs\text\00" +
LTrim$(Str$([Article ID Number])) + ".txt", 1)
        ElseIf [Article ID Number] < 100 Then
```

```
            x = Shell("Write.EXE c:\winapps\access\docs\text\0" +
LTrim$(Str$([Article ID Number])) + ".txt", 1)
        ElseIf [Article ID Number] < 1000 Then
            x = Shell("Write.EXE c:\winapps\access\docs\text\" +
LTrim$(Str$([Article ID Number])) + ".txt", 1)
        End If
        SendKeys "%n"
    Else
        MsgBox "Text not available", 0, "LitBase"
    End If

Exit_TextDisplayButton_Click:
    Exit Sub

Err_TextDisplayButton_Click:
    MsgBox Error$
    Resume Exit_TextDisplayButton_Click

End Sub

```

Objects

Section: Detail0

Section: FormFooter2

Section: FormHeader1

Command Button: AuthorButton

Command Button: Button39

Object Frame: Embedded44

Option Group: Field41

Option Group: Field52

Command Button: IDNumberButton

Command Button: PrintAuthorButton

Command Button: PrintIDNumberButton

Command Button: PrintPublicationButton

Command Button: PrintTitleButton

Command Button: PrintYearButton

Command Button: PublicationButton

Command Button: ReturnButton3

Label: Text12

Label: Text13

Label: Text42

Label: Text53

Command Button: YearButton

Code

```
Option Compare Database   'Use database order for string comparisons

Sub AuthorButton_Click ()
On Error GoTo Err_AuthorButton_Click

    Dim DocName As String

    DocName = "Article Sort by Author"
    DoCmd OpenQuery DocName, A_NORMAL, A_EDIT

Exit_AuthorButton_Click:
    Exit Sub

Err_AuthorButton_Click:
    MsgBox Error$
    Resume Exit_AuthorButton_Click

End Sub

Sub Button39_Click ()
On Error GoTo Err_Button39_Click

    Dim DocName As String

    DocName = "Article Sort by Title"
    DoCmd OpenQuery DocName, A_NORMAL, A_EDIT

Exit_Button39_Click:
    Exit Sub

Err_Button39_Click:
    MsgBox Error$
    Resume Exit_Button39_Click

End Sub

Sub Button47_Click ()
On Error GoTo Err_Button47_Click

    Dim DocName As String

    DocName = "Article Sort by Article ID Number"
    DoCmd OpenReport DocName, A_PREVIEW

Exit_Button47_Click:
    Exit Sub

Err_Button47_Click:
    MsgBox Error$
    Resume Exit_Button47_Click

End Sub

Sub IDNumberButton_Click ()
On Error GoTo Err_IDNumberButton_Click

    Dim DocName As String

    DocName = "Article Sort by Article ID Number"
    DoCmd OpenQuery DocName, A_NORMAL, A_EDIT

Exit_IDNumberButton_Click:
    Exit Sub

Err_IDNumberButton_Click:
    MsgBox Error$
    Resume Exit_IDNumberButton_Click

End Sub

Sub PrintAuthorButton_Click ()
On Error GoTo Err_PrintAuthorButton_Click

    Dim DocName As String

    DocName = "Article Sort by Author"
    DoCmd OpenReport DocName, A_NORMAL

Exit_PrintAuthorButton_Click:
    Exit Sub

Err_PrintAuthorButton_Click:
    MsgBox Error$
    Resume Exit_PrintAuthorButton_Click

End Sub

Sub PrintIDNumberButton_Click ()
On Error GoTo Err_PrintIDNumberButton_Click

    Dim DocName As String

    DocName = "Article Sort by Article ID Number"
    DoCmd OpenReport DocName, A_NORMAL

Exit_PrintIDNumberButton_Click:
    Exit Sub

Err_PrintIDNumberButton_Click:
    MsgBox Error$
    Resume Exit_PrintIDNumberButton_Click

End Sub

Sub PrintPublicationButt_Click ()
On Error GoTo Err_PrintPublicationButt_Click

    Dim DocName As String

    DocName = "Article Sort by Publication"
    DoCmd OpenReport DocName, A_NORMAL

Exit_PrintPublicationButt_Click:
    Exit Sub

Err_PrintPublicationButt_Click:
    MsgBox Error$
    Resume Exit_PrintPublicationButt_Click

End Sub

```

```
Sub PrintTitleButton_Click ()
On Error GoTo Err_PrintTitleButton_Click

    Dim DocName As String

    DocName = "Article Sort by Title Report"
    DoCmd OpenReport DocName, A_NORMAL

Exit_PrintTitleButton_Click:
    Exit Sub

Err_PrintTitleButton_Click:
    MsgBox Error$
    Resume Exit_PrintTitleButton_Click

End Sub

Sub PrintYearButton_Click ()
On Error GoTo Err_PrintYearButton_Click

    Dim DocName As String

    DocName = "Article Sort by Year"
    DoCmd OpenReport DocName, A_NORMAL

Exit_PrintYearButton_Click:
    Exit Sub

Err_PrintYearButton_Click:
    MsgBox Error$
    Resume Exit_PrintYearButton_Click

End Sub

Sub PublicationButton_Click ()
On Error GoTo Err_PublicationButton_Click

    Dim DocName As String

    DocName = "Article Sort by Publication"
    DoCmd OpenQuery DocName, A_NORMAL, A_EDIT

Exit_PublicationButton_Click:
    Exit Sub

Err_PublicationButton_Click:
    MsgBox Error$
    Resume Exit_PublicationButton_Click

End Sub

Sub ReturnButton3_Click ()
On Error GoTo Err_ReturnButton3_Click


```

```
    DoCmd Close

Exit_ReturnButton3_Click:
    Exit Sub

Err_ReturnButton3_Click:
    MsgBox Error$
    Resume Exit_ReturnButton3_Click

End Sub

Sub YearButton_Click ()
On Error GoTo Err_YearButton_Click

    Dim DocName As String

    DocName = "Article Sort by Year"
    DoCmd OpenQuery DocName, A_NORMAL, A_EDIT

Exit_YearButton_Click:
    Exit Sub

Err_YearButton_Click:
    MsgBox Error$
    Resume Exit_YearButton_Click

End Sub

```

Objects

Section: Detail0

Section: FormFooter2

Section: FormHeader1

Command Button: AboutLitBaseButton

Command Button: ArticleEntryButton

Command Button: ArticleSearchButton

Command Button: ArticleSortButton

Object Frame: Embedded39

Command Button: ExitButton

Command Button: HelpButton

Label: Text12

Label: Text13

Command Button: UserFieldsButton2

Code

```
Option Compare Database   'Use database order for string comparisons

Sub AboutLitBaseButton_Click ()
On Error GoTo Err_AboutLitBaseButton_Click

    Dim DocName As String
    Dim LinkCriteria As String

    DocName = "AboutLitBase"
    DoCmd OpenForm DocName, , , LinkCriteria

Exit_AboutLitBaseButton_Click:
    Exit Sub

Err_AboutLitBaseButton_Click:
    MsgBox Error$
    Resume Exit_AboutLitBaseButton_Click

End Sub

Sub ArticleEntryButton_Click ()
On Error GoTo Err_ArticleEntryButton_Click

    Dim DocName As String
    Dim LinkCriteria As String

    DocName = "Article Entry"
    DoCmd OpenForm DocName, , , LinkCriteria

Exit_ArticleEntryButton_Click:
    Exit Sub

Err_ArticleEntryButton_Click:
    MsgBox Error$
    Resume Exit_ArticleEntryButton_Click

End Sub

Sub ArticleSearchButton_Click ()
On Error GoTo Err_ArticleSearchButton_Click

    Dim DocName As String
    Dim LinkCriteria As String

    DocName = "Article Search"
    DoCmd OpenForm DocName, , , LinkCriteria

Exit_ArticleSearchButton_Click:
    Exit Sub

Err_ArticleSearchButton_Click:
    MsgBox Error$
    Resume Exit_ArticleSearchButton_Click

End Sub

Sub ArticleSortButton_Click ()
On Error GoTo Err_ArticleSortButton_Click

    Dim DocName As String
    Dim LinkCriteria As String

    DocName = "Article Sort"
    DoCmd OpenForm DocName, , , LinkCriteria

Exit_ArticleSortButton_Click:
    Exit Sub

Err_ArticleSortButton_Click:
    MsgBox Error$
    Resume Exit_ArticleSortButton_Click

End Sub

Sub Button44_Click ()
On Error GoTo Err_Button44_Click

    Dim DocName As String
    Dim LinkCriteria As String

    DocName = "User Fields"
    DoCmd OpenForm DocName, , , LinkCriteria

Exit_Button44_Click:
    Exit Sub

Err_Button44_Click:
    MsgBox Error$
    Resume Exit_Button44_Click

End Sub

Sub ExitButton_Click ()
On Error GoTo Err_ExitButton_Click


    DoCmd Quit

Exit_ExitButton_Click:
    Exit Sub

Err_ExitButton_Click:
    MsgBox Error$
    Resume Exit_ExitButton_Click

End Sub

Sub HelpButton_Click ()
On Error GoTo Err_HelpButton_Click

    Dim DocName As String
    Dim LinkCriteria As String

    DocName = "AboutLitBase"
    DoCmd OpenForm DocName, , , LinkCriteria

Exit_HelpButton_Click:
    Exit Sub

Err_HelpButton_Click:
    MsgBox Error$
    Resume Exit_HelpButton_Click

End Sub

Sub TitleSort_Button_Click ()
On Error GoTo Err_TitleSort_Button_Click

    Dim DocName As String

    DocName = "Article Sort by Title Report"
    DoCmd OpenReport DocName, A_PREVIEW

Exit_TitleSort_Button_Click:
    Exit Sub
```

```

Err_TitleSort_Button_Click:
    MsgBox Error$
    Resume Exit_TitleSort_Button_Click

End Sub

Sub UserFieldsButton2_Click ()
On Error GoTo Err_UserFieldsButton2_Click

    Dim DocName As String
    Dim LinkCriteria As String

    DocName = "User Fields"
    DoCmd OpenForm DocName, , , LinkCriteria

Exit_UserFieldsButton2_Click:
    Exit Sub

Err_UserFieldsButton2_Click:
    MsgBox Error$
    Resume Exit_UserFieldsButton2_Click

End Sub

```

Objects

Section: Detail0

Section: FormFooter1

Section: FormHeader0

Line: Line31

Command Button: RetumButton2

Label: Text1

Label: Text11

Label: Text13

Label: Text15

Label: Text16

Label: Text18

Label: Text26

Label: Text27

Label: Text29

Label: Text3

Label: Text30

Label: Text5

Label: Text7

Label: Text9

Text Box: uf1

Text Box: uf10

Text Box: uf2

Text Box: uf3

Text Box: uf4

Text Box: uf5

Text Box: uf6

Text Box: uf7

Text Box: uf8

Text Box: uf9

Code

```
Option Compare Database   'Use database order for string comparisons

Sub ReturnButton2_Click ()
On Error GoTo Err_ReturnButton2_Click


    DoCmd Close

Exit_ReturnButton2_Click:
    Exit Sub

Err_ReturnButton2_Click:
    MsgBox Error$
    Resume Exit_ReturnButton2_Click


End Sub

```

Objects

Group Level 0

Control Source:	Article ID Number	Group Footer:	No
Group Header:	No	Group Interval:	1
Group On:	Each Value	Keep Together:	No
Sort Order:	Ascending		

Section: Detail1

Section: PageFooter2

Section: PageHeader0

Section: ReportFooter4

Section: ReportHeader3

Text Box: Article ID Number

Text Box: Author(s)

Text Box: Field19

Text Box: Field20

Text Box: GrandTotal_Article ID Number

Line: Line21

Line: Line22

Line: Line23

Line: Line24

Text Box: Publication

Label: Text11

Label: Text13

Label: Text16

Label: Text18

Label: Text25

Label: Text5

Label: Text7

Label: Text9

Text Box: Title

Text Box: Volume & pp

Text Box: Year

Code

```
Option Compare Database   'Use database order for string comparisons

```

__Objects__

Group Level 0

Control Source:	Author(s)	Group Footer:	No
Group Header:	No	Group Interval:	1
Group On:	Each Value	Keep Together:	No
Sort Order:	Ascending		

Section: Detail1

Section: PageFooter2

Section: PageHeader0

Section: ReportFooter4

Section: ReportHeader3

Text Box: Article ID Number

Text Box: Author(s)

Text Box: Field19

Text Box: Field20

Text Box: GrandTotal_Article ID Number

Line: Line21

Line: Line22

Line: Line23

Line: Line24

Text Box: Publication

Label: Text11

Label: Text13

Label: Text16

Label: Text18

Label: Text25

Label: Text5

Label: Text7

Label: Text9

Text Box: Title

Text Box: Volume & pp

Text Box: Year

Code

```
Option Compare Database   'Use database order for string comparisons

```

Objects

Group Level 0

Control Source:	Publication	Group Footer:	No
Group Header:	No	Group Interval:	1
Group On:	Each Value	Keep Together:	No
Sort Order:	Ascending		

Group Level 1

Control Source:	Title	Group Footer:	No
Group Header:	No	Group Interval:	1
Group On:	Each Value	Keep Together:	No
Sort Order:	Ascending		

Section: Detail1

Section: PageFooter2

Section: PageHeader0

Section: ReportFooter4

Section: ReportHeader3

Text Box: Article ID Number

Text Box: Author(s)

Text Box: Field19

Text Box: Field20

Text Box: GrandTotal_Article ID Number

Line: Line21

Line: Line22

Line: Line23

Line: Line24

Text Box: Publication

Label: Text11

Label: Text13

Label: Text16

Label: Text18

Label: Text25

Label: Text5

Label: Text7

Label: Text9

Text Box: Title

Text Box: Volume & pp

Text Box: Year

<u>Code</u>

```
Option Compare Database   'Use database order for string comparisons

```

Objects

Group Level 0

Control Source:	Title	Group Footer:	No
Group Header:	No	Group Interval:	1
Group On:	Each Value	Keep Together:	No
Sort Order:	Ascending		

Section: Detail1

Section: PageFooter2

Section: PageHeader0

Section: ReportFooter4

Section: ReportHeader3

Text Box: Article ID Number

Text Box: Author(s)

Text Box: Field19

Text Box: Field20

Text Box: GrandTotal_Article ID Number

Line: Line21

Line: Line22

Line: Line23

Line: Line24

Text Box: Publication

Label: Text11

Label: Text13

Label: Text16

Label: Text18

Label: Text25

Label: Text5

Label: Text7

Label: Text9

Text Box: Title

Text Box: Volume & pp

Text Box: Year

Code

```
Option Compare Database   'Use database order for string comparisons

```

Objects

Group Level 0

Control Source:	Year	Group Footer:	No
Group Header:	No	Group Interval:	1
Group On:	Each Value	Keep Together:	No
Sort Order:	Descending		

Group Level 1

Control Source:	Title	Group Footer:	No
Group Header:	No	Group Interval:	1
Group On:	Each Value	Keep Together:	No
Sort Order:	Ascending		

Section: Detail1

Section: PageFooter2

Section: PageHeader0

Section: ReportFooter4

Section: ReportHeader3

Text Box: Article ID Number

Text Box: Author(s)

Text Box: Field19

Text Box: Field20

Text Box: GrandTotal_Article ID Number

Line: Line21

Line: Line22

Line: Line23

Line: Line24

Text Box: Publication

Label: Text11

Label: Text13

Label: Text16

Label: Text18

Label: Text25

Label: Text5

Label: Text7

Label: Text9

Text Box: Title

Text Box: Volume & pp

Text Box: Year

Code

```
Option Compare Database   'Use database order for string comparisons

```

Actions

Name	Condition	Action	Argument	Value
		OpenForm	Form Name:	MainMenu
			View:	Form
			Filter Name:	
			Where Condition:	
			Data Mode:	Edit
			Window Mode:	Normal
		OpenForm	Form Name:	AboutLitBase
			View:	Form
			Filter Name:	
			Where Condition:	
			Data Mode:	Edit
			Window Mode:	Normal

APPENDIX D

APPARATUS SCHEDULE

Table D.1 Hardware apparatus schedule

This table lists the hardware apparatus used for the work reported in the thesis. The apparatus are grouped according to apparatus type. The code is used to refer to the apparatus from the thesis. The manufacturer and model of the apparatus is listed, as well as a concise specification or description of the piece of apparatus.

[illegible]	[illegible]	[illegible]	[illegible]
Document Scanners	H1	Hewlett Packard Scanjet	Flatbed, 300 DPI optical resolution, 8 bit grey.
	H2	Hewlett Packard Scanjet IIC and Automatic Document Feeder	Flatbed, 400 DPI optical resolution, 24 bit color, 20 s/page scanning speed.
	H3	Microtech Pagewiz and Automatic Document Feeder	Roller feed, 300 DPI optical resolution, 4 bit grey, 6 s/page scanning speed.
Compact Disc Recorder	H4	Phillips CDD521	352.8 KB/s transfer rate, 650MB storage capacity
Compact Disc JukeBox	H5	Pioneer DRM604X	6 CD, 3.25 GB capacity, 612 KB/s transfer rate
Image Display Monitors	H6	NEC 6FG	21 inch display, 1280 x 1024 pixel screen resolution
	H7	Videocom CA-1718	17 inch display, 1280 x 1024 pixel screen resolution
Computing Hardware	H8	Intel 80486 DX	33 MHz processor speed, 16 MB RAM
	H9	Intel 80486 DX	50 MHz processor speed, 20 MB RAM
	H10	Intel 80486 DX2	66 MHz processor speed, 32 MB RAM
	H11	Intel Pentium	90 MHz processor speed, 32 MB RAM
Printers	H12	Epson LQ400	180 DPI, 24 pin dot matrix
	H13	Hewlett Packard PaintJet	180 DPI ink jet
	H14	Toshiba PageLaser6	300 DPI laser
	H15	Hewlett Packard LaserJet 3D	300 DPI laser
	H16	LexMark 4029 042	600 DPI laser

Table D.2 Software apparatus schedule

This table lists the software apparatus used for the work reported in the thesis. The apparatus are grouped according to apparatus type. The code is used to refer to the apparatus from the thesis. The manufacturer and model of the apparatus is listed, as well as a concise specification or description of the piece of apparatus.

[illegible]	[illegible]	[illegible]	[illegible]
Operating Systems	S1	Microsoft DOS v6.22	Disk operating system
	S2	Microsoft Windows for Workgroups v3.11	Graphical user interface operating system
OCR Software	S3	Caere Corp. OmniPage Professional Edition v2.0	Omnifont OCR, Greyscale OCR, up to 4000 words per minute recognition speed.
	S4	Calera Recognition Systems WordScan Plus v1.1	Omnifont OCR
Modelling Software	S5	Microsoft Excel v5.0	Programmable spreadsheet and modelling system
	S6	Palisade @Risk	Risk analysis and modelling module for MS Excel
Programming Software	S7	Microsoft Visual Basic Professional Edition v3.0	Object oriented Windows programming
	S8	Microsoft Access v2.0	Relational database programming
	S9	Media Architects ImageKnife v1.0	Visual Basic image manipulation function library
	S10	Data Techniques ImageMan/VB v2.0	Visual Basic image manipulation and scanning control function library
	S11	Black Ice Software !mage SDK	Windows image manipulation function library
	S12	Kodak ImagePac v2.2	CCITT G4 compressed image display
	S13	Microsoft Write v3.11	ASCII text display
Image Processing Software	S14	Aldus Photostyler v1.1a	24bit image manipulation
Text Processing Software	S15	Odyssey Development ISYS v3.0	Text indexing, search and retrieval
CD-ROM Mastering Software	S16	Phillips CDWrite v1.0	ISO 9660 standard CD-ROM writing
	S17	Tata Unisys CD-Gen v2.10	ISO 9660 standard CD-ROM writing

APPENDIX E

SUMMARY OF EXPERIMENTAL DATA

Table E.1 Typeface experiment OCR accuracy data for the full character set

This table lists the OCR accuracy data for five typefaces using the full character set. The OCR accuracy is listed for each of the five samples. The mean OCR accuracy for the five samples is listed, as is the maximum variation of the samples from the mean.

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
OCR-A	96.50	96.60	97.18	95.96	96.86	96.62	0.66
OCR-B	96.30	96.59	96.64	97.69	97.33	96.91	0.78
Courier	98.58	99.71	98.71	99.23	99.37	99.12	0.59
Helvetica	97.22	96.73	97.73	97.31	96.51	97.10	0.63
Times Roman	99.35	98.79	99.05	98.18	99.28	98.93	0.75

Table E.2 Typeface experiment OCR accuracy data for the limited char. set

This table lists the OCR accuracy data for five typefaces using the limited character set. The OCR accuracy is listed for each of the five samples. The mean OCR accuracy for the five samples is listed, as is the maximum variation of the samples from the mean.

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
OCR-A	100.00	100.00	100.00	100.00	100.00	100.00	0.00
OCR-B	100.00	100.00	100.00	100.00	100.00	100.00	0.00
Courier	99.39	98.94	99.93	99.92	99.37	99.51	0.57
Helvetica	100.00	99.67	100.00	99.60	99.78	99.81	0.21
Times Roman	100.00	100.00	100.00	100.00	99.25	99.85	0.60

Table E.3 Font type experiment OCR accuracy data

This table lists the OCR accuracy data for eight font types. The OCR accuracy is listed for each of the five samples. The mean OCR accuracy for the five samples is listed, as is the maximum variation of the samples from the mean.

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Normal	99.37	99.87	99.62	98.89	99.18	99.39	0.51
Bold	97.38	97.60	97.11	97.25	96.96	97.26	0.60
Italic	96.45	96.38	96.18	96.11	96.51	96.33	0.51
Underline	84.94	85.36	85.17	84.74	85.21	85.08	0.36
Bold Italic	92.01	91.85	92.30	92.42	92.33	92.18	0.42
Bold Underline	80.67	80.79	81.38	80.82	80.69	80.87	0.38
Italic Underline	80.49	80.77	80.48	80.46	81.00	80.64	0.54
Bold Italic Underline	73.26	72.73	72.86	73.23	72.50	72.92	0.50

Table E.4 Text size experiment OCR accuracy data

This table lists the OCR accuracy data for 16 text sizes. The OCR accuracy is listed for each of the five samples. The mean OCR accuracy for the five samples is listed, as is the maximum variation of the samples from the mean.

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
4	46.12	44.72	45.67	45.18	44.57	45.25	1.43
5	55.79	55.25	56.93	56.57	56.13	56.13	1.75
6	80.99	80.08	80.88	79.04	78.92	79.98	1.08
7	83.49	86.29	86.57	84.42	84.30	85.01	1.57
8	93.93	94.34	91.78	93.59	91.27	92.98	1.73
9	94.24	97.79	96.96	94.55	95.25	95.76	1.79
10	94.63	97.22	96.42	95.18	96.06	95.90	1.37
11	97.81	97.38	99.03	96.78	98.85	97.97	1.22
12	97.66	97.33	98.76	97.53	98.37	97.93	1.76
13	96.50	97.18	99.33	98.25	99.51	98.15	1.51
14	97.79	95.37	95.99	96.39	94.36	95.98	1.79
16	96.94	96.85	98.52	98.83	99.06	98.04	1.65
18	99.46	98.77	100.00	100.00	99.26	99.49	0.70
20	100.00	98.89	99.72	98.74	97.89	99.04	1.50
22	99.99	97.79	97.93	98.87	98.11	98.53	1.61
24	99.52	100.00	97.25	100.00	100.00	99.35	1.75

Table E.5 Text size experiment OCR speed data

This table lists the OCR speed data for 15 text sizes. The OCR speed is listed for each of the five samples. The mean OCR speed for the five samples is listed, as is the maximum variation of the samples from the mean.

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
4	43.90	46.86	42.30	42.90	46.58	44.51	2.70
5	100.84	96.61	97.67	102.02	100.97	99.62	3.02
6	136.93	137.81	139.24	136.04	141.51	138.31	2.96
7	141.29	140.41	144.49	138.37	139.29	140.77	3.49
8	141.95	144.62	142.56	145.25	142.53	143.38	3.25
9	142.94	138.81	143.11	139.14	140.55	140.91	2.19
10	139.20	137.25	136.28	138.04	140.63	138.28	3.63
11	76.30	76.92	78.90	78.23	83.50	78.77	3.70
12	74.13	76.36	74.12	76.82	74.22	75.13	2.82
14	69.12	67.65	69.58	70.07	70.47	69.38	2.35
16	55.21	59.63	60.02	56.16	62.04	58.61	3.79
18	54.22	52.32	57.32	53.99	56.06	54.78	2.68
20	57.3	57.89	58.75	60.57	57.61	58.42	2.57
22	47.71	48.83	43.43	49.97	45.43	47.07	3.57
24	46.61	44.78	48.47	43.72	46.91	46.10	2.47

Table E.6 Image resolution experiment OCR accuracy data for 200 DPI

This table lists the OCR accuracy data for 15 text sizes at 200 DPI. The OCR accuracy is listed for each of the five samples. The mean OCR accuracy for the five samples is listed, as is the maximum variation of the samples from the mean.

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
4	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	3.83	0.60	4.90	5.27	4.79	3.88	2.40
6	14.65	13.70	10.70	15.05	11.90	13.20	3.05
7	17.56	17.32	15.17	11.94	16.17	15.63	3.06
8	57.74	57.78	54.75	53.16	52.93	55.27	2.78
9	93.60	90.04	91.82	91.95	91.80	91.84	1.96
10	98.54	95.51	96.50	98.94	96.98	97.29	1.94
11	96.19	99.61	97.59	96.78	96.41	97.32	1.81
12	97.06	99.07	99.18	96.51	99.14	98.19	1.49
14	98.60	98.59	97.89	96.64	97.78	97.90	1.36
16	98.58	97.91	100.00	100.00	97.91	98.88	1.09
18	99.27	98.33	97.86	98.71	97.61	98.36	1.39
20	99.92	97.89	98.33	100.00	99.65	99.16	1.11
22	97.74	99.56	99.32	97.84	98.28	98.55	1.26
24	99.86	97.93	97.61	98.54	99.12	98.61	1.39

Table E.7 Image resolution experiment OCR accuracy data for 300 DPI

This table lists the OCR accuracy data for 15 text sizes at 300 DPI. The OCR accuracy is listed for each of the five samples. The mean OCR accuracy for the five samples is listed, as is the maximum variation of the samples from the mean.

4	10.80	5.40	6.51	10.55	5.74	7.80	2.80
5	39.74	40.61	38.87	37.31	35.33	38.37	2.67
6	74.15	77.84	72.67	75.63	74.57	74.97	2.84
7	91.26	89.54	88.90	93.27	90.20	90.63	3.10
8	100.00	100.00	97.75	100.00	97.25	99.00	2.00
9	98.46	97.98	98.00	97.06	98.12	97.92	0.94
10	99.59	98.29	99.77	99.11	99.70	99.29	0.77
11	99.35	98.49	100.00	100.00	100.00	99.57	1.51
12	99.50	100.00	98.87	100.00	100.00	99.67	1.13
14	98.97	100.00	99.18	97.[illegible]0	100.00	99.17	1.30
16	100.00	98.32	98.07	98.33	97.84	98.51	1.16
18	100.00	99.37	99.17	97.83	97.85	98.84	1.17
20	100.00	97.90	97.56	99.84	99.73	99.01	1.44
22	99.29	98.28	97.61	99.28	99.09	98.71	1.39
24	99.91	98.17	97.88	99.29	99.45	98.94	1.12

Table E.8 Image resolution experiment OCR speed data

This table lists the OCR speed data for seven image resolutions. The OCR speed is listed for each of the five samples. The mean OCR speed for the five samples is listed, as is the maximum variation of the samples from the mean.

100	155.77	148.69	151.69	149.61	155.63	155.14	4.31
150	133.30	139.59	138.39	141.98	136.62	136.98	4.70
200	109.37	114.53	109.85	112.90	111.49	110.30	3.53
250	78.10	80.11	81.16	82.27	80.80	78.99	2.27
300	62.43	59.71	55.86	55.85	59.80	55.45	4.15
350	49.65	51.50	51.37	49.79	51.54	47.60	4.54
400	48.49	53.56	54.44	48.64	53.88	54.18	3.44

Table E.9 Printing device experiment OCR accuracy data for laser

This table lists the OCR accuracy data for three typefaces using a laser printer. The OCR accuracy is listed for each of the five samples. The mean OCR accuracy for the five samples is listed, as is the maximum variation of the samples from the mean.

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Courier	99.45	99.65	98.67	99.77	99.76	99.46	0.63
Helvetica	97.47	97.91	97.33	98.15	96.68	97.51	0.95
Times Roman	98.93	98.80	99.29	99.12	97.82	98.79	0.98

Table E.10 Printing device experiment OCR accuracy data for dot matrix

This table lists the OCR accuracy data for three typefaces using a dor matrix printer. The OCR accuracy is listed for each of the five samples. The mean OCR accuracy for the five samples is listed, as is the maximum variation of the samples from the mean.

[illegible]	[illegible]						
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Courier	96.94	97.37	97.88	99.23	96.63	97.61	1.47
Helvetica	98.65	95.59	96.75	96.09	97.11	96.84	1.75
Times Roman	98.56	96.61	99.26	96.66	98.48	97.91	1.46

Table E.11 Printing device experiment OCR accuracy data for ink jet

This table lists the OCR accuracy data for three typefaces using an ink jet printer. The OCR accuracy is listed for each of the five samples. The mean OCR accuracy for the five samples is listed, as is the maximum variation of the samples from the mean.

[illegible]	[illegible]						
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Courier	96.14	94.94	95.00	95.25	96.23	95.51	0.76
Helvetica	96.67	96.99	96.87	97.70	97.86	97.22	1.16
Times Roman	93.51	93.72	94.68	94.83	95.12	94.37	1.09

Table E.12 Image skew experiment OCR accuracy data

This table lists the OCR accuracy data for 16 image skew angles. The OCR accuracy is listed for each of the five samples. The mean OCR accuracy for the five samples is listed, as is the maximum variation of the samples from the mean.

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
0.00	98.33	98.69	97.72	98.38	97.37	98.10	0.69
0.96	98.23	98.33	97.86	97.93	97.41	97.95	0.59
1.89	95.83	98.88	96.24	96.32	98.13	97.08	1.88
2.81	97.39	98.23	97.41	97.84	97.08	97.59	0.92
3.76	90.77	93.79	90.66	90.73	91.96	91.58	1.79
4.32	88.72	87.66	88.07	87.37	88.22	88.01	0.72
4.60	81.01	79.51	82.12	82.50	81.67	81.36	1.50
5.03	77.52	77.82	78.96	76.32	79.89	78.10	1.89
5.44	63.22	66.85	65.44	62.59	66.76	64.97	2.41
5.61	62.73	59.25	63.24	60.04	61.29	61.31	2.75
5.70	65.00	66.64	62.60	64.78	65.93	64.99	2.64
5.95	51.68	51.50	48.26	46.96	49.32	49.54	2.68
6.19	41.13	37.57	37.87	39.63	38.32	38.90	2.13
6.41	36.35	36.48	39.35	37.43	38.20	37.56	2.35
6.58	28.66	31.22	27.82	31.44	28.94	29.62	2.18
5.75	11.67	7.07	10.52	12.09	9.83	10.24	3.09

Table E.13 Text size and image resolution experiment OCR accuracy data

This table lists the OCR accuracy data for 21 text sizes at 7 image resolutions. The mean OCR accuracy data (of the five samples) is listed for each text size and image resolution.

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
4	0.00	0.00	0.00	4.13	8.25	10.51	12.77
5	0.00	0.46	0.92	19.51	38.09	54.12	70.14
6	0.00	6.08	12.16	42.94	73.72	83.26	92.80
7	0.00	8.21	16.42	54.19	91.96	93.66	95.36
8	0.00	27.35	54.70	75.90	97.10	97.85	98.59
9	2.13	47.24	92.34	95.15	97.95	98.55	99.15
10	3.98	50.51	97.03	98.12	99.21	99.42	99.62
11	6.29	52.37	98.45	99.01	99.56	99.52	99.47
12	8.14	53.68	99.22	99.13	99.04	99.29	99.54
13	8.72	54.06	99.41	99.30	99.19	99.28	99.37
14	9.30	54.45	99.59	99.46	99.33	99.27	99.20
15	30.92	65.02	99.12	99.11	99.10	98.88	98.65
16	52.54	75.59	98.64	99.13	99.61	98.86	98.10
17	71.86	85.43	99.00	99.20	99.40	99.16	98.93
18	91.17	95.27	99.36	99.42	99.48	99.62	99.75
19	94.95	97.12	99.29	99.45	99.61	99.71	99.81
20	98.72	98.97	99.22	99.48	99.73	99.80	99.86
21	99.10	99.24	99.39	99.43	99.48	99.56	99.64
22	99.48	99.52	99.55	99.39	99.22	99.32	99.42
23	99.20	99.24	99.29	99.04	98.80	99.04	99.28
24	99.84	99.43	99.02	98.70	98.38	98.76	99.14

Table E.14 Text size and image resolution experiment OCR speed data

This table lists the OCR speed data for 21 text sizes at 7 image resolutions. The mean OCR speed data (of the five samples) is listed for each text size and image resolution.

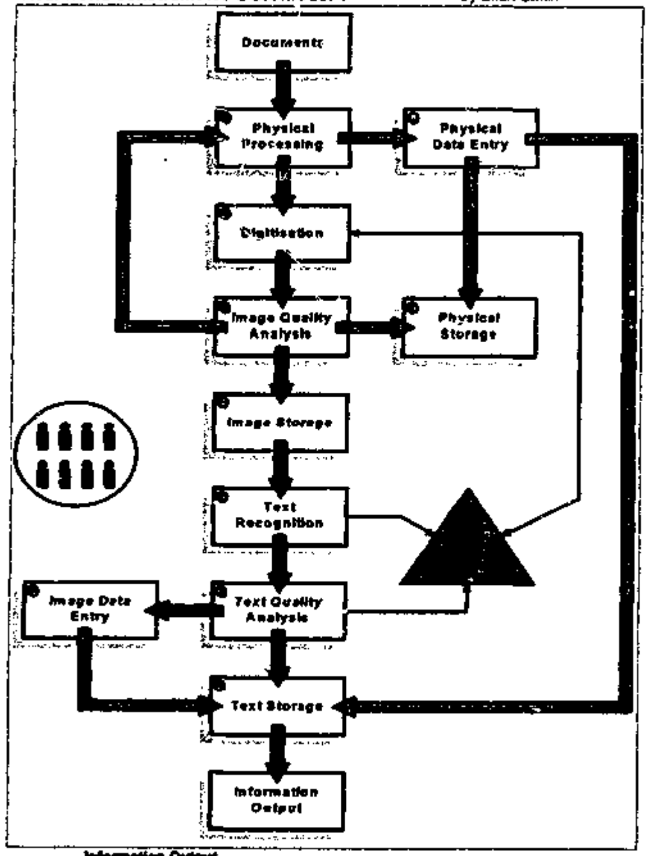
4	0.00	0.00	0.00	0.00	0.00	0.00	17.00
5	0.00	0.00	0.00	0.00	0.00	0.32	24.00
6	0.00	0.00	0.00	0.00	41.60	39.05	37.00
7	0.00	0.00	0.62	0.43	26.67	26.85	27.00
8	0.00	0.00	24.20	22.13	19.62	21.13	23.00
9	0.00	0.00	21.93	21.04	20.93	23.34	25.00
10	0.00	0.00	18.95	21.25	22.41	25.03	27.99
11	0.00	0.00	18.31	22.34	25.37	26.46	28.06
12	0.00	0.00	17.10	23.28	29.21	29.13	28.34
13	0.25	0.35	16.06	22.45	27.96	26.20	25.00
14	26.21	20.96	16.17	21.15	26.80	23.94	23.50
15	26.78	23.65	19.78	24.37	30.59	27.25	25.42
16	27.34	26.34	23.39	27.59	35.62	30.55	27.34
17	21.01	23.20	27.13	29.78	34.78	29.56	26.56
18	14.67	20.05	30.86	31.97	33.99	28.56	25.78
19	16.86	20.20	25.94	26.30	27.37	25.79	24.83
20	19.04	20.34	21.01	20.63	22.91	23.02	23.87
21	18.42	20.41	22.40	23.98	25.62	24.40	23.93
22	17.79	20.47	23.79	27.32	29.05	25.78	23.98
23	17.46	19.30	23.48	24.86	24.30	24.50	23.42
24	17.13	18.12	23.16	22.39	22.18	23.21	22.86

APPENDIX F

OCR SYSTEM MODEL SOURCE CODE AND EXAMPLES

Real world constraint HV OCR model V

by Brian Griffin



Information Output

Total Processing Time	[illegible]	minutes
Total Number of Operators	[illegible]	operators
Processing rate	[illegible]	documents/min
Processing rate per Operator	[illegible]	document [illegible] documents/hour

[illegible]	FormReader 2
[illegible]	ARTS v1.32
[illegible]	48
[illegible]	1800
[illegible]	80
[illegible]	0.314
[illegible]	
[illegible]	200
[illegible]	12
[illegible]	

Operator cost per Hour

$15

Operator Idle Cost

[illegible]

		Actual
Physical Quality Pass Rate	87.00%	85.[illegible]%
Image Quality Pass Rate	60.00%	58.33%
Text Quality Pass Rate	80.00%	77.68%

Documents Input

Initial Number of Documents 2000 documents

Physical Processing Process (1)

Process Time	5	minutes
Initial Operators	15	operators
Processing Rate	3.00	documents/min

Digitisation Process (2)

Process Time	2	minutes
Initial Operators	4	operators
Processing Rate	2.00	documents/min

Image Quality Analysis Process (3)

Process Time	3	minutes
Initial Operators	8	operators
Processing Rate	2.67	documents/min

Image Storage Process (4)

Process Time	4	minutes
Initial Operators	6	operators
Processing Rate	1.50	documents/min

Text Recognition Process (5)

Process Time	2	minutes
Initial Operators	6	operators
Processing Rate	3.00	documents/min

Text Quality Analysis Process (6)

Process Time	5	minutes
Initial Operators	15	operators
Processing Rate	3.00	documents/min

Text Storage Process (7)

Process Time	2	minutes
Initial Operators	4	operators
Processing Rate	2.00	documents/min

Physical Data Entry Process (8)

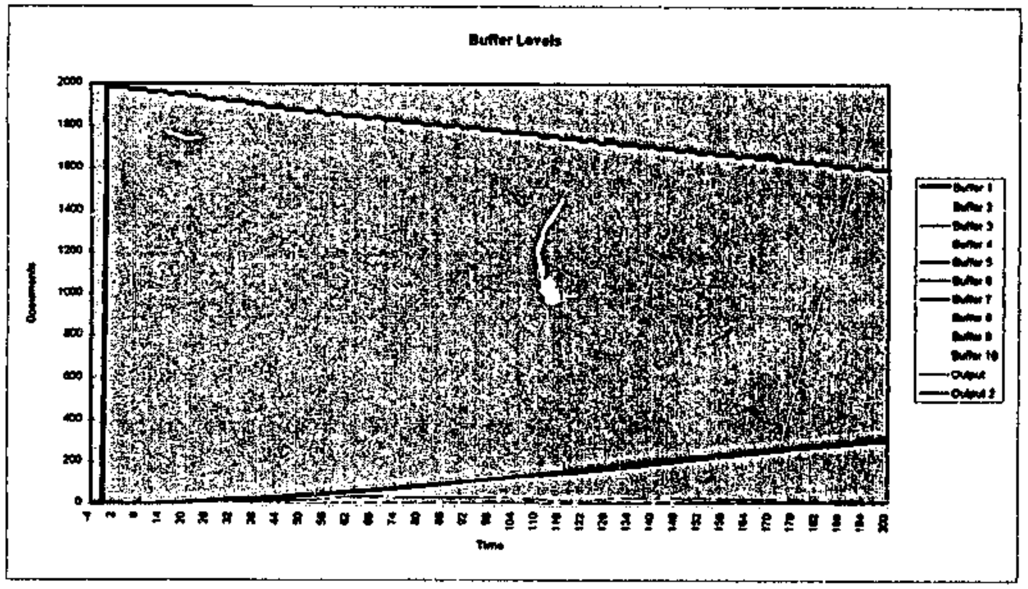
Process Time	6	minutes
Initial Operators	8	operators
Processing Rate	1.33	documents/min

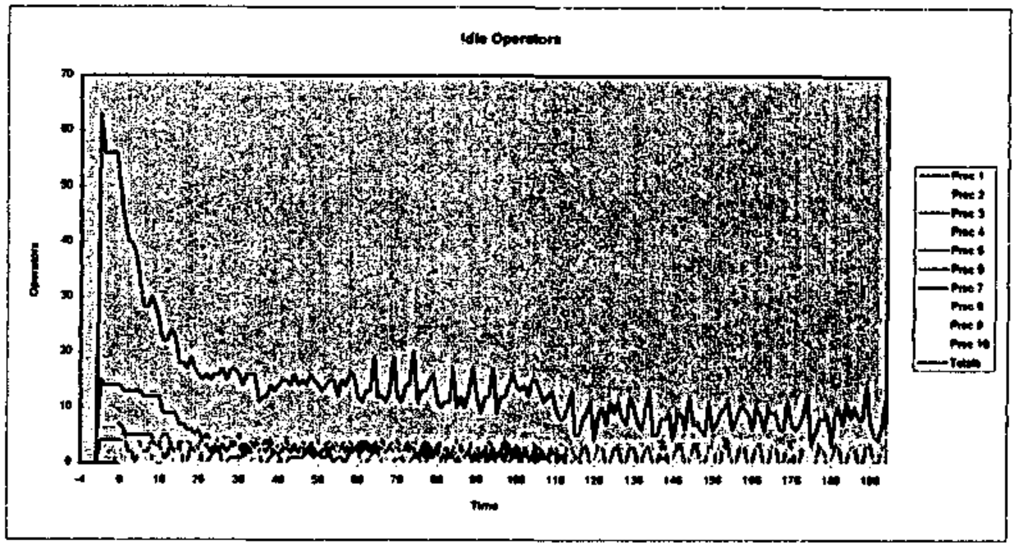
Physical Storage Process (9)

Process Time	4	minutes
Initial Operators	6	operators
Processing Rate	1.50	documents/min

Image Data Entry Process (10)

Process Time	2	minutes
Initial Operators	6	operators
Processing Rate	3.00	documents/min





Process Number	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Process Time	5	2	3	4	2	5	2	6	4	2
Initial Operators	15	4	8	6	6	15	4	8	6	6
[illegible]	2000	0	0	0	0	0	0	0	0	0
[illegible]	4	4	4	3	5	4	4	4	3	4
[illegible]	30	20	20	20	20	20	20	20	20	20

Rate	3	2	2.6667	1.5	3	3	2	1.33333
Delay	5	2	3	4	2	5	2	6
Operators	15	4	8	6	6	15	4	8
Time	Buffer 1	Buffer 2	Buffer 3	Buffer 4	Buffer 5	Buffer 6	Euffer 7	Buffer 8
-4	0	0	0	0	0	0	0	0
-3	0	0	0	0	0	0	0	0
-2	0	0	0	0	0	0	0	0
-1	0	0	0	0	0	0	0	0
0	2000	0	0	0	0	0	0	0
1	1985	0	0	0	0	0	0	0
2	1985	0	0	0	0	0	0	0
3	1985	0	0	0	0	0	0	0
4	1985	0	0	0	0	0	0	0
5	1985	13	0	0	0	0	0	2
6	1970	9	0	0	0	0	0	0
7	1970	9	4	0	0	0	0	0
8	1970	5	0	0	0	0	0	0
9	1970	5	4	0	0	0	0	0
10	1972	13	2	2	0	0	0	3
11	1957	13	4	0	0	0	2	0
12	1959	9	2	0	0	0	0	0
13	1961	9	4	0	0	0	0	0
14	1961	5	2	2	2	0	0	0
15	1962	17	4	1	0	0	0	3
16	1948	13	2	1	0	2	3	0
17	1949	12	4	1	0	0	0	0
18	1951	8	3	1	2	0	0	0
19	1952	7	5	1	1	0	0	0
20	1952	15	4	2	1	2	0	3
21	1939	14	6	0	0	1	5	0
22	1939	10	5	2	1	1	1	0
23	1939	9	7	3	1	0	1	0
24	1939	5	6	4	2	1	0	0
25	1939	17	8	4	0	1	1	2
26	1925	13	7	5	1	2	4	0
27	1925	12	9	6	1	0	3	0
28	1926	8	8	6	2	1	2	0
29	1927	7	10	5	0	1	0	0
30	1929	17	9	5	1	2	1	1
31	1915	16	11	5	1	0	5	0
32	1917	12	10	4	2	1	2	0
33	1918	11	12	3	0	1	2	0
34	1919	7	11	4	1	2	1	0
35	1919	19	13	5	1	0	2	2
36	1905	15	12	5	2	1	1	0
37	1905	14	14	5	0	1	1	0
38	1906	10	13	6	1	2	0	0
39	1906	9	15	7	1	0	1	0
40	1906	17	14	8	2	1	1	3
41	1891	16	16	8	0	1	3	0
42	1892	12	15	9	1	2	1	0
43	1893	11	17	9	1	0	2	0

1.5	3	batches/minute					
4	2	process time					
6	6	Text Store	Phys Store				Batche
Buffer 9	Buffer 10	Output	Output 2		Finished Time?		Proc 1
0	0	0	0		1000		0
0	0	0	0		1000		0
0	0	0	0		1000		0
0	0	0	0		1000		0
0	0	0	0		1000		0
0	0	0	0		1000		15
0	0	0	0		1000		0
0	0	0	0		1000		0
0	0	0	0		1000		0
0	0	0	0		1000		0
0	0	0	0		1000		15
0	0	0	0		1000		0
0	0	0	0		1000		0
0	0	0	0		1000		0
2	0	0	0		1000		0
2	0	0	0		1000		15
0	0	0	0		1000		0
0	0	2	0		1000		0
2	0	2	2		1000		0
1	0	2	4		1000		0
4	0	2	4		1000		15
4	0	2	4		1000		0
4	0	5	6		1000		0
3	0	5	7		1000		0
4	0	5	8		1000		0
6	0	5	8		1000		15
8	0	5	10		1000		0
8	0	9	11		1000		0
9	0	9	12		1000		0
10	1	10	12		1000		0
14	0	10	14		1000		15
14	0	11	15		1000		0
14	0	14	16		1000		0
14	1	15	16		1000		0
14	0	17	18		1000		0
15	0	17	19		1000		15
14	0	18	20		1000		0
14	0	21	20		1000		0
15	0	22	22		1000		0
15	0	24	23		1000		0
16	0	25	24		1000		15
17	0	27	24		1000		0
18	1	28	26		1000		0
18	1	29	27		1000		0
19	0	29	28		1000		0
22	1	30	28		1000		15
23	0	31	30		1000		0
22	1	34	31		1000		0

2000	Total Batches							
1000	Completion Time							
78	Total Operators							
started to be processed								
Proc 2	Proc 3	Proc 4	Proc 5	Proc 6	Proc 7	Proc 8	Proc 9	Proc 10
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	2	0	0
0	0	0	0	0	0	0	0	0
4	4	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
4	2	0	0	0	0	0	0	0
0	2	2	0	0	0	3	2	0
4	2	0	0	0	2	0	2	0
0	2	0	0	0	0	0	0	0
4	2	0	0	0	0	0	0	0
0	2	2	2	0	0	0	2	0
4	2	1	0	0	0	3	1	0
1	2	1	0	2	3	0	1	0
4	2	0	0	0	0	0	0	0
1	2	1	2	0	0	0	2	0
4	2	1	1	0	0	0	1	0
1	2	2	1	2	0	3	1	0
4	2	0	0	1	4	0	0	0
1	2	1	1	1	0	0	2	0
4	2	1	1	0	1	0	1	0
1	2	2	2	1	0	0	1	0
4	2	0	0	1	1	2	0	1
1	2	1	1	2	3	0	2	0
4	2	1	1	0	1	0	1	0
1	2	2	2	1	2	0	1	0
4	2	0	0	1	0	0	0	1
1	2	1	1	2	1	1	2	0
4	2	1	1	0	3	0	1	0
1	2	2	2	1	1	0	1	0
4	2	0	0	1	2	0	0	0
1	2	1	1	2	1	0	2	0
4	2	1	1	0	2	2	1	0
1	2	2	2	1	1	0	1	0
4	2	0	0	1	1	0	0	0
1	2	1	1	2	0	0	2	1
4	2	1	1	0	1	0	1	1
1	2	2	2	1	1	3	1	0
4	2	0	0	1	3	0	0	1
1	2	1	1	2	1	0	2	0

5242	Operator Idle Minutes								
Idle Operators									
Proc 1	Proc 2	Proc 3	Proc 4	Proc 5	Proc 6	Proc 7	Proc 8	Proc 9	Proc 10
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	4	8	6	6	15	4	8	6	6
0	4	7	5	5	14	4	7	5	5
0	4	7	5	5	14	4	7	5	5
0	4	7	5	5	14	4	7	5	5
0	4	7	5	5	14	4	7	5	5
0	0	7	5	5	14	4	5	5	5
0	0	6	4	5	13	4	4	4	4
0	0	2	4	5	13	4	4	4	4
0	0	2	4	5	13	4	4	4	4
0	0	0	4	5	13	4	4	4	4
0	0	2	2	5	13	4	1	2	4
0	0	0	2	5	12	2	3	0	4
0	0	0	2	5	12	2	3	0	4
0	0	0	2	5	12	4	3	0	4
0	0	0	2	3	12	4	3	0	4
0	0	0	1	3	12	4	0	1	4
0	0	0	0	5	9	1	3	0	4
0	0	0	0	5	9	1	3	0	4
0	0	0	1	3	9	4	3	0	4
0	0	0	1	2	9	4	3	0	4
0	0	0	0	3	7	4	0	0	4
0	0	0	0	4	7	0	3	0	4
0	0	0	0	4	6	0	3	0	4
0	0	0	0	3	6	3	3	0	4
0	0	0	0	2	5	3	3	0	4
0	0	0	0	3	6	3	1	0	3
0	0	0	0	4	4	0	4	0	3
0	0	0	0	3	5	0	4	0	4
0	0	0	0	2	4	1	4	0	4
0	0	0	0	3	4	2	4	0	3
0	0	0	0	4	3	3	3	0	3
0	0	0	0	3	5	0	5	0	4
0	0	0	0	2	4	0	5	0	4
0	0	0	0	3	4	1	5	0	4
0	0	0	0	4	3	1	5	0	4
0	0	0	0	3	5	1	3	0	4
0	0	0	0	2	3	1	4	0	4
0	0	0	0	3	3	2	4	0	4
0	0	0	0	4	2	3	4	0	3
0	0	0	0	3	4	3	4	0	2
0	0	0	0	2	3	2	1	0	3
0	0	0	0	3	3	0	3	0	3
0	0	0	0	4	2	0	3	0	3

		5	Operator transfer delay (minutes)					
		20	Operator number transfer factor					
		Operator transfer requests						
Totals		Proc 1	Proc 2	Proc 3	Proc 4	Proc 5	Proc 6	Proc 7
0		0	0	0	0	0	0	0
0		0	0	0	0	0	0	0
0		0	0	0	0	0	0	0
0		0	0	0	0	0	0	0
0		0	0	0	0	0	0	0
63		0	0	-1	-1	-1	-1	0
56		0	0	0	0	0	0	0
56		0	0	0	0	0	0	0
56		0	0	0	0	0	0	0
56		0	0	0	0	0	0	0
50		0	0	-1	-1	0	-1	0
44		0	0	0	0	0	0	0
40		0	0	0	0	0	0	0
40		0	0	0	0	0	0	0
38		0	0	0	0	0	0	0
33		0	0	0	0	0	-1	0
28		0	0	0	0	0	0	0
28		0	0	0	0	0	0	0
30		0	0	0	0	0	0	0
28		0	0	0	0	0	0	0
25		0	1	0	0	0	-1	0
22		0	0	0	0	0	0	0
22		0	0	0	0	0	0	0
24		0	0	0	0	0	0	0
23		0	0	0	0	0	0	0
18		0	0	0	0	0	-1	0
18		0	0	0	0	0	0	0
17		0	0	0	0	0	0	0
19		0	0	0	0	0	0	0
17		0	0	0	0	0	0	0
16		0	0	0	0	0	-1	0
15		0	0	0	0	0	0	0
16		0	0	0	0	0	0	0
15		0	0	0	0	0	0	0
16		0	0	0	0	0	0	0
16		0	0	0	0	0	0	0
17		0	0	0	0	0	0	0
15		0	0	0	0	0	0	0
17		0	0	0	0	0	0	0
17		0	0	0	0	0	0	0
16		0	0	0	0	0	-1	0
14		0	0	0	0	0	0	0
16		0	0	0	0	0	0	0
16		0	0	0	0	0	0	0
16		0	0	0	0	0	0	0
11		0	0	0	0	0	0	0
12		0	0	0	0	0	0	0
12		0	0	0	0	0	0	0

					Operators				
Proc 8	Proc 9	Proc 10	Total		Proc 1	Proc 2	Proc 3	Proc 4	Proc 5
0	0	0	0						
0	0	0	0						
0	0	0	0						
0	0	0	0						
0	0	0	0		15	4	8	6	6
-1	-1	-1	-7		15	4	7	5	5
0	0	0	0		15	4	7	5	5
0	0	0	0		15	4	7	5	5
0	0	0	0		15	4	7	5	5
0	0	0	0		15	4	7	5	5
-1	-1	-1	-6		15	4	6	4	5
0	0	0	0		15	4	6	4	5
0	0	0	0		15	4	6	4	5
0	0	0	0		15	4	6	4	5
0	0	0	0		15	4	6	4	5
0	0	0	-1		15	4	6	4	5
0	0	0	0		15	4	6	4	5
0	0	0	0		15	4	6	4	5
0	0	0	0		15	4	6	4	5
0	0	0	0		15	4	6	4	5
0	0	0	0		15	5	6	4	5
0	0	0	0		15	5	6	4	5
0	0	0	0		15	5	6	4	5
0	0	0	0		15	5	6	4	5
0	0	0	0		15	5	6	4	5
0	0	0	-1		15	5	6	4	5
0	0	0	0		15	5	6	4	5
0	0	0	0		15	5	6	4	5
0	0	0	0		15	5	6	4	5
0	0	0	0		15	5	6	4	5
0	0	0	-1		15	5	6	4	5
0	0	0	0		15	5	6	4	5
0	0	0	0		15	5	6	4	5
0	0	0	0		15	5	6	4	5
0	0	0	0		15	5	6	4	5
0	0	0	0		15	5	6	4	5
0	0	0	0		15	5	6	4	5
0	0	0	0		15	5	6	4	5
0	0	0	0		15	5	6	4	5
0	0	0	0		15	5	6	4	5
0	0	0	-1		15	5	6	4	5
0	0	0	0		15	5	6	4	5
0	0	0	0		15	5	6	4	5
0	0	0	0		15	5	6	4	5
0	0	0	0		15	5	6	4	5
0	0	0	0		15	5	6	4	5
0	0	0	0		15	5	6	4	5
0	0	0	0		15	5	6	4	5

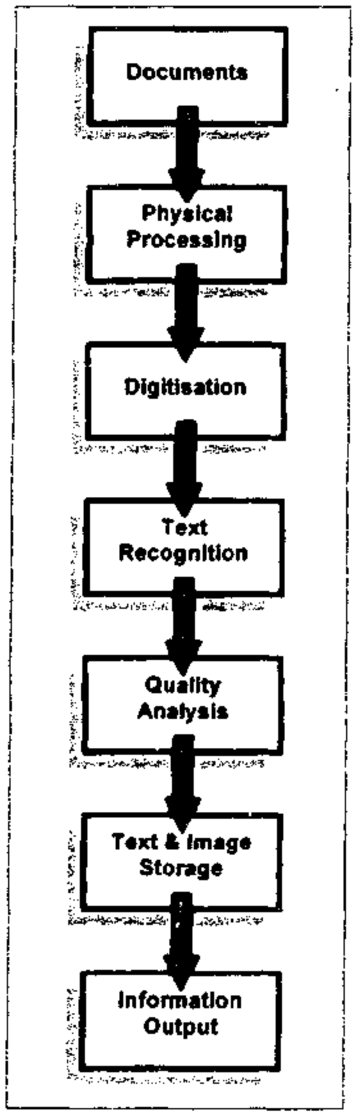
Proc 6	Proc 7	Proc 8	Proc 9	Proc 10	Total		Operators Available
							0
							0
							0
							0
15	4	8	6	6	78		0
14	4	7	5	5	71		7
14	4	7	5	5	71		7
14	4	7	5	5	71		7
14	4	7	5	5	71		7
14	4	7	5	5	71		7
13	4	6	4	4	65		13
13	4	6	4	4	65		13
13	4	6	4	4	65		13
13	4	6	4	4	65		13
13	4	6	4	4	65		13
12	4	6	4	4	64		14
12	4	6	4	4	64		14
12	4	6	4	4	64		14
12	4	6	4	4	64		14
12	4	6	4	4	64		14
11	4	6	4	4	64		14
11	4	6	4	4	64		14
11	4	6	4	4	64		14
11	4	6	4	4	64		14
11	4	6	4	4	64		14
10	4	6	4	4	63		15
10	4	6	4	4	63		15
10	4	6	4	4	63		15
10	4	6	4	4	63		15
10	4	6	4	4	63		15
9	4	6	4	4	62		16
9	4	6	4	4	62		16
9	4	6	4	4	62		16
9	4	6	4	4	62		16
9	4	6	4	4	62		16
9	4	6	4	4	62		16
9	4	6	4	4	62		16
9	4	6	4	4	62		16
9	4	6	4	4	62		16
9	4	6	4	4	62		16
8	4	6	4	4	61		17
8	4	6	4	4	61		17
8	4	6	4	4	61		17
8	4	6	4	4	61		17
8	4	6	4	4	61		17
8	4	6	4	4	61		17
8	4	6	4	4	61		17
8	4	6	4	4	61		17

	Random Pass Functions				Fail Functions		
	Q1	Q3	Q6		Q1	Q3	Q6
	0	0	0		0	0	0
	0	0	0		0	0	0
	0	0	0		0	0	0
	0	0	0		0	0	0
	0	0	0		0	0	0
	13	0	0		2	0	0
	0	0	0		0	0	0
	0	0	0		0	0	0
	0	0	0		0	0	0
	0	0	0		0	0	0
	12	2	0		3	2	0
	0	0	0		0	0	0
	0	0	0		0	2	0
	0	0	0		0	2	0
	0	2	0		0	0	0
	12	1	0		3	1	0
	0	1	0		0	1	0
	0	1	0		0	1	0
	0	0	0		0	2	0
	0	1	0		0	1	0
	12	2	0		3	0	0
	0	0	2		0	2	0
	0	2	0		0	0	0
	0	2	0		0	0	0
	0	2	0		0	0	0
	13	2	1		2	0	1
	0	1	1		0	1	0
	0	2	1		0	0	0
	0	1	0		0	1	0
	0	1	0		0	1	1
	14	0	1		1	2	0
	0	1	2		0	1	0
	0	0	0		0	2	0
	0	1	1		0	1	0
	0	1	1		0	1	0
	13	2	2		2	0	0
	0	1	0		0	1	0
	0	2	1		0	0	0
	0	1	0		0	1	1
	0	2	1		0	0	1
	12	2	0		3	0	0
	0	2	0		0	0	1
	0	1	1		0	1	0
	0	1	1		0	1	1

	Quality Percentages						
	Q1	Q3	Q6				
	85.50%	58.33%	77.88%	Averages			
	FALSE	FALSE	FALSE				
	FALSE	FALSE	FALSE				
	FALSE	FALSE	FALSE				
	FALSE	FALSE	FALSE				
	FALSE	FALSE	FALSE				
	86.67%	FALSE	FALSE				
	FALSE	FALSE	FALSE				
	FALSE	FALSE	FALSE				
	FALSE	FALSE	FALSE				
	FALSE	FALSE	FALSE				
	80.00%	50.00%	FALSE				
	FALSE	FALSE	FALSE				
	FALSE	0.00%	FALSE				
	FALSE	0.00%	FALSE				
	FALSE	100.00%	FALSE				
	80.00%	50.00%	FALSE				
	FALSE	50.00%	FALSE				
	FALSE	50.00%	FALSE				
	FALSE	0.00%	FALSE				
	FALSE	50.00%	FALSE				
	80.00%	100.00%	FALSE				
	FALSE	0.00%	100.00%				
	FALSE	100.00%	FALSE				
	FALSE	100.00%	FALSE				
	FALSE	100.00%	FALSE				
	86.67%	100.00%	50.00%				
	FALSE	50.00%	100.00%				
	FALSE	100.00%	100.00%				
	FALSE	50.00%	FALSE				
	FALSE	50.00%	0.00%				
	93.33%	0.00%	100.00%				
	FALSE	50.00%	100.00%				
	FALSE	0.00%	FALSE				
	FALSE	50.00%	100.00%				
	FALSE	50.00%	100.00%				
	86.67%	100.00%	100.00%				
	FALSE	50.00%	FALSE				
	FALSE	100.00%	100.00%				
	FALSE	50.00%	0.00%				
	FALSE	100.00%	50.00%				
	80.00%	100.00%	FALSE				
	FALSE	100.00%	0.00%				
	FALSE	50.00%	100.00%				
	FALSE	50.00%	50.00%				

High volume OCR model II

by Brian Griffin



Documents Input

Initial Number of documents 100 documents

Buffers (2,3,4) already loaded (40,4,12 documents)

Physical Processing Stage (1)

Process Time 5 minutes

Operators/Machines 15 operators

Processing Rate 3.00 documents/min

Digitisation Stage (2)

Process Time 2 minutes

Operators/Machines 4 operators

Processing Rate 2.00 documents/min

Text Recognition Stage (3)

Process Time 3 minutes

Operators/Machines 8 operators

Processing Rate 2.67 documents/min

Quality Analysis Stage (4)

Process Time 4 minutes

Operators/Machines 6 operators

Processing Rate 1.50 documents/min

Text and Image Storage Stage (5)

Process Time 2 minutes

Operators/Machines 6 operators

Processing Rate 3.00 documents/min

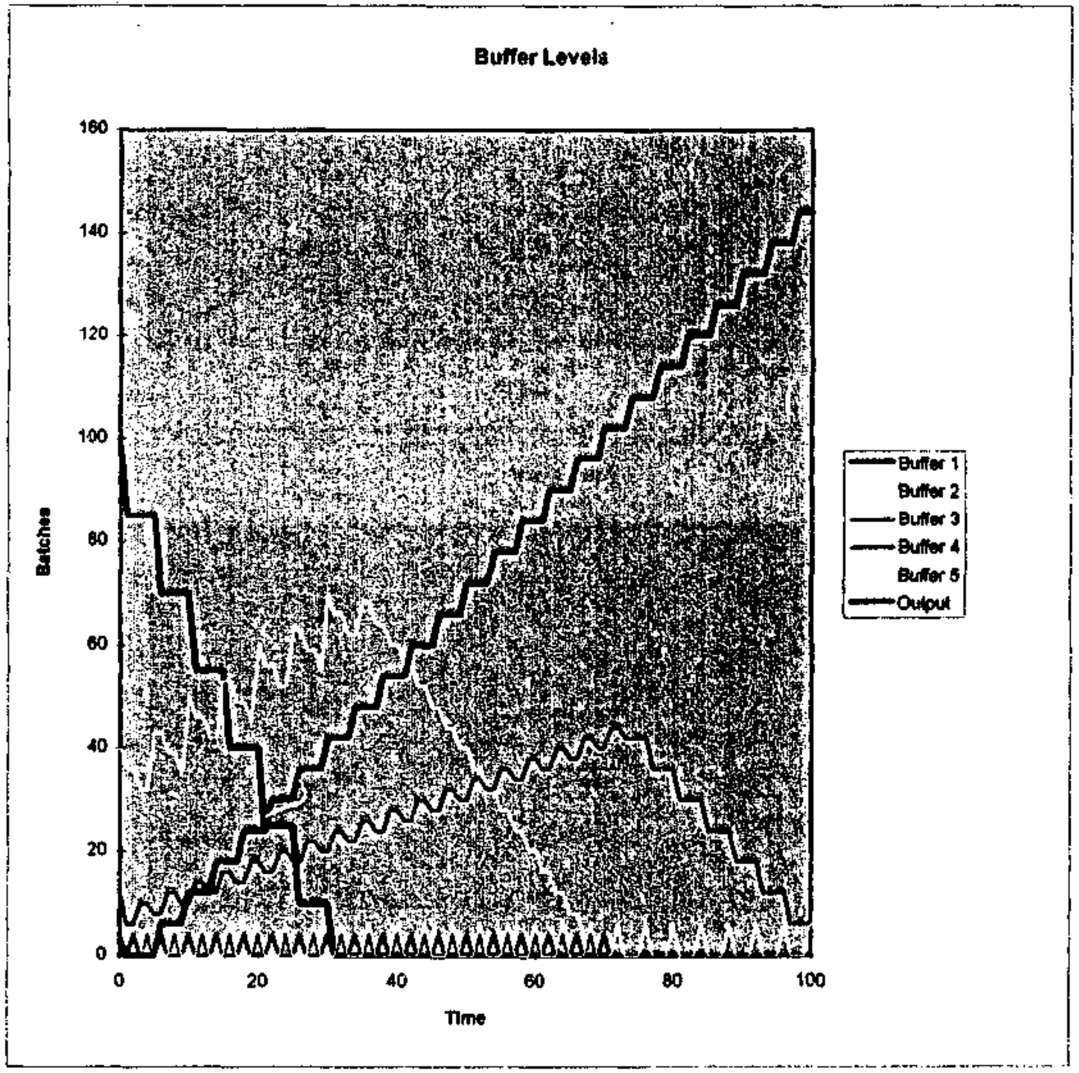
Information Output

Total Processing Time 106 minutes

Total Number of Operators 39 operators

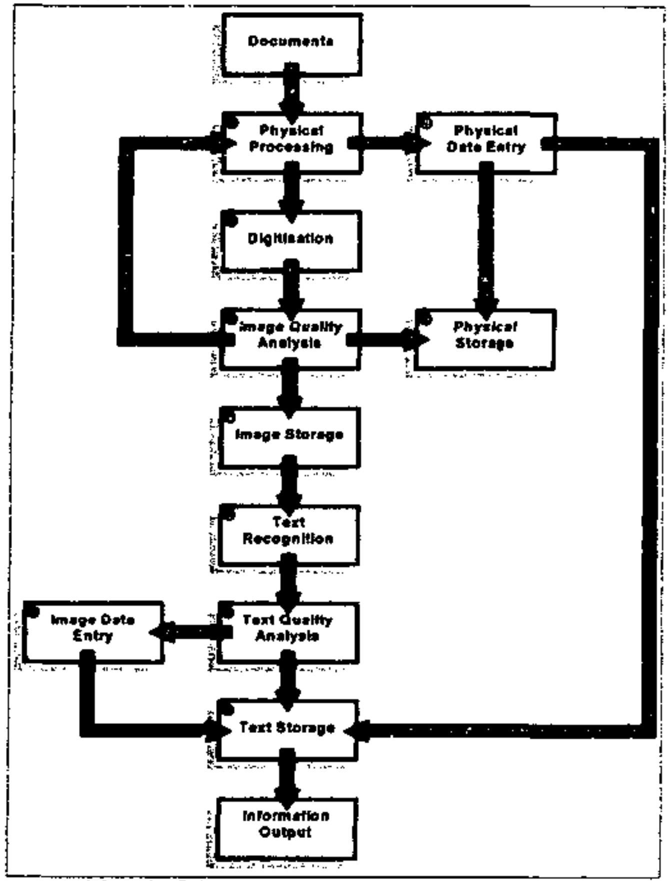
Processing rate 0.94 documents/min

Processing rate per Operator 0.0242 documents/mi 1.4514 documents/hour



Non-sequential HV OCR model III

By Brian Griffin



Information Output
Total Processing Time [redacted] minutes
Total Number of Operators [redacted] operators
Processing rate [redacted] documents/min
Processing rate per Operator [redacted] document [redacted] documents/hour

Physical Quality Pass Rate
87.00%

Image Quality Pass Rate
80.00%

Text Quality Pass Rate
80.00%

Operator cost per Hour
$15

Operator Idle Cost
[redacted]

Documents Input
Initial Number of documents 200 documents

Physical Processing Stage (1)
Process Time 5 minutes
Initial Operators 15 operators
Processing Rate 3.00 documents/min

Digitisation Stage (2)
Process Time 2 minutes
Initial Operators 4 operators
Processing Rate 2.00 documents/min

Image Quality Analysis Stage (3)
Process Time 3 minutes
Initial Operators 8 operators
Processing Rate 2.67 documents/min

Image Storage Stage (4)
Process Time 4 minutes
Initial Operators 6 operators
Processing Rate 1.50 documents/min

Text Recognition Stage (5)
Process Time 2 minutes
Initial Operators 6 operators
Processing Rate 3.00 documents/min

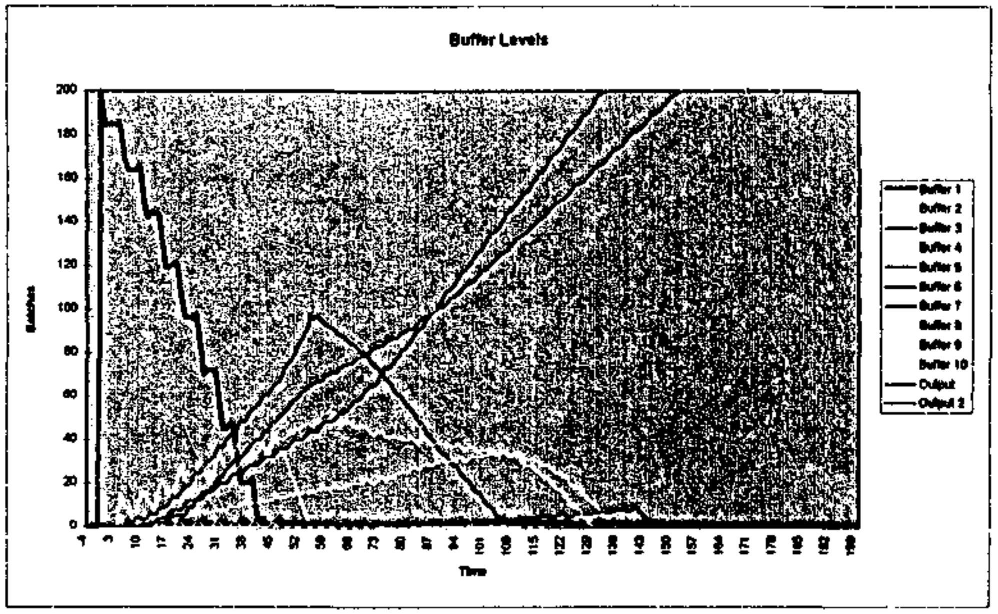
Text Quality Analysis Stage (6)
Process Time 5 minutes
Initial Operators 15 operators
Processing Rate 3.00 documents/min

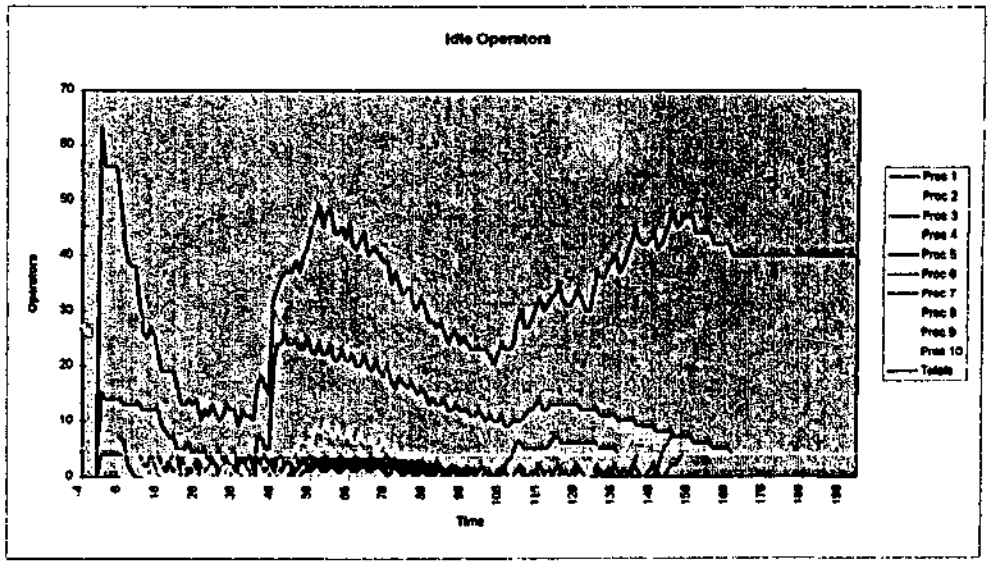
Text Storage Stage (7)
Process Time 2 minutes
Initial Operators 4 operators
Processing Rate 2.00 documents/min

Physical Data Entry Stage (8)
Process Time 6 minutes
Initial Operators 8 operators
Processing Rate 1.33 documents/min

Physical Storage Stage (9)
Process Time 4 minutes
Initial Operators 6 operators
Processing Rate 1.50 documents/min

Image Data Entry Stage (10)
Procecs Time 2 minutes
Initial Operators 6 operators
Processing Rate 3.00 documents/min

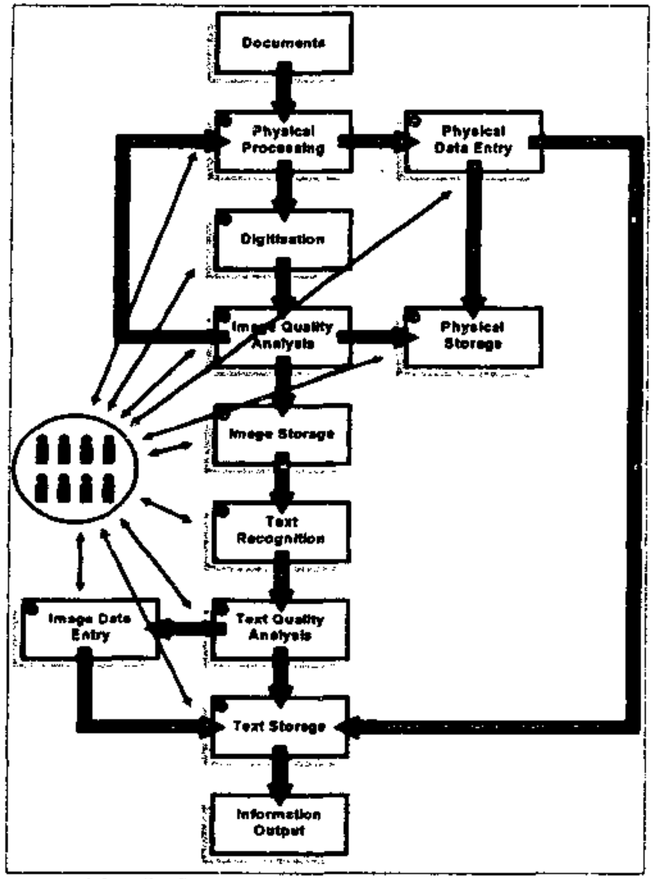

Buffer Levels


Idle Operators

[illegible]	5	2	3	4	2	5	2	6	4	2
[illegible]	15	4	8	6	6	15	4	6	6	6
[illegible]	200	0	0	0	0	0	0	0	0	0
[illegible]	4	4	4	4	4	4	4	4	4	4
[illegible]	20	20	20	20	20	20	20	20	20	20

Self optimising HV OCR model IV

By Brian Griffin



Information Output
Total Processing Time [redacted] minutes
Total Number of Operators operators
Processing rate documents/min
Processing rate per Operator document [redacted] documents/hour

Operator cost per Hour
$15

Operator Idle Cost
[redacted]

		Actual
Physical Quality Pass Rate	67.00%	69.23%
Image Quality Pass Rate	60.00%	66.34%
Text Quality Pass Rate	80.00%	82.88%

Documents Input
Initial Number of documents 200 documents

Physical Processing Stage (1)
Process Time 5 minutes
Initial Operators 15 operators
Processing Rate 3.00 documents/min

Digitisation Stage (2)
Process Time 2 minutes
Initial Operators 4 operators
Processing Rate 2.00 documents/min

Image Quality Analysis Stage (3)
Process Time 3 minutes
Initial Operators 8 operators
Processing Rate 2.67 documents/min

Image Storage Stage (4)
Process Time 4 minutes
Initial Operators 6 operators
Processing Rate 1.50 documents/min

Text Recognition Stage (5)
Process Time 2 minutes
Initial Operators 6 operators
Processing Rate 3.00 documents/min

Text Quality Analysis Stage (6)
Process Time 5 minutes
Initial Operators 15 operators
Processing Rate 3.00 documents/min

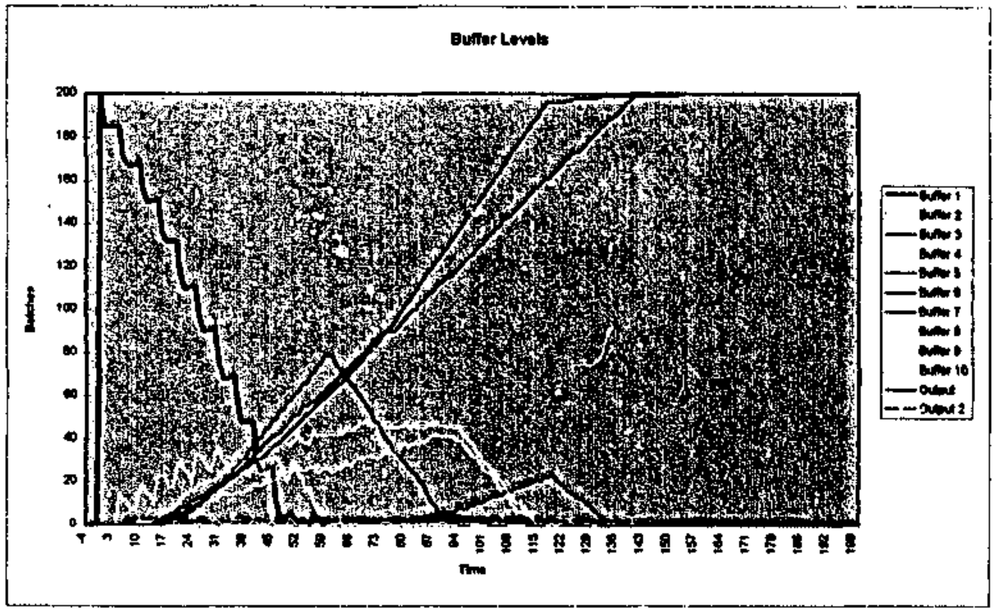
Text Storage Stage (7)
Process Time 2 minutes
Initial Operators 4 operators
Processing Rate 2.00 documents/min

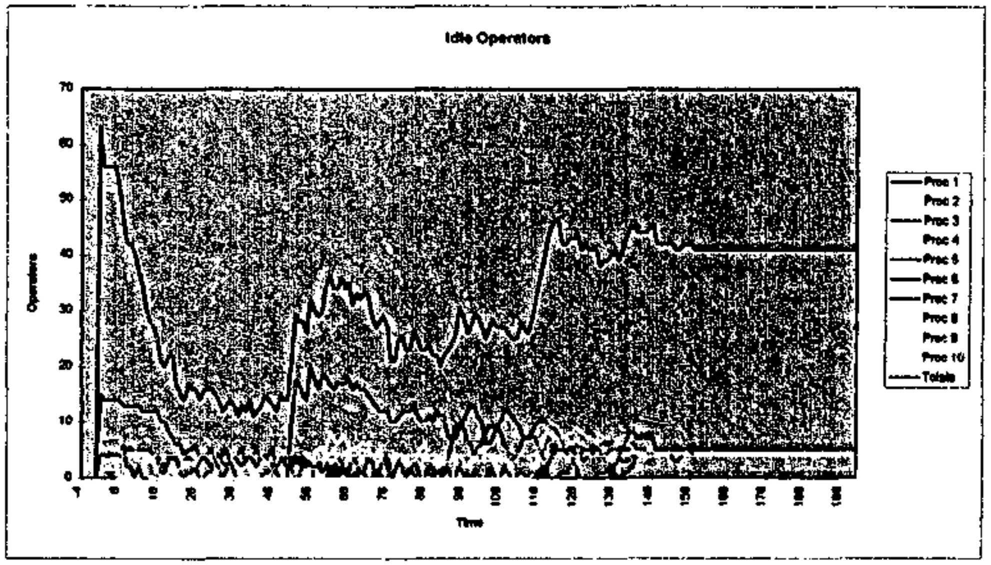
Physical Data Entry Stage (8)
Process Time 6 minutes
Initial Operators 8 operators
Processing Rate 1.33 documents/min

Physical Storage Stage (9)
Process Time 4 minutes
Initial Operators 6 operators
Processing Rate 1.50 documents/min

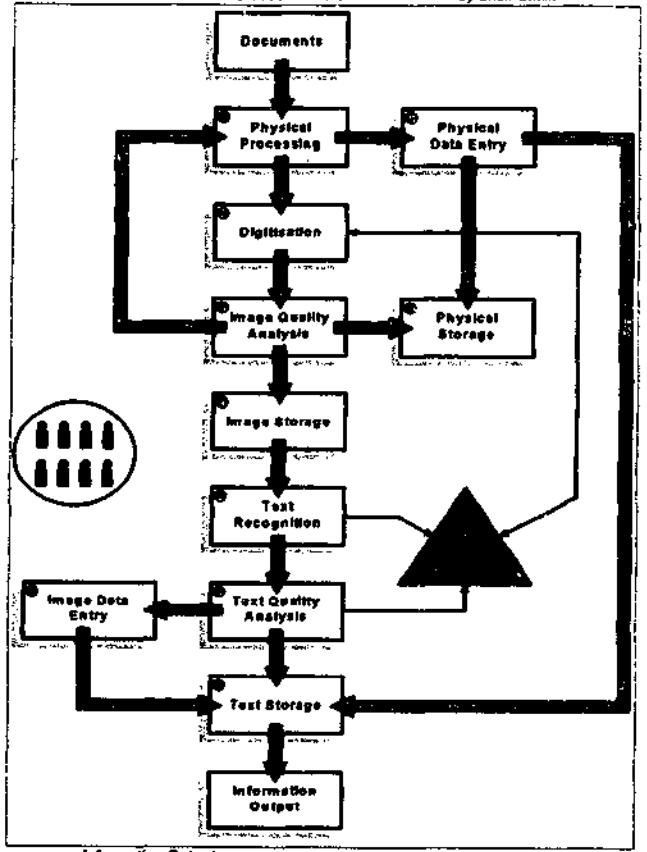
Image Data Entry Stage (10)
Process Time 2 minutes
Initial Operators 6 operators
Processing Rate 3.00 documents/min

	5	2	3	4	2	5	2	6	4	2
	15	4	8	6	6	15	4	8	6	6
	200	0	0	0	0	0	0	0	0	0
	4	4	4	3	5	4	4	4	3	4
	30	20	20	20	20	20	20	20	20	20





Real world constraint HV OCR model V by Brian Griffin



Information Output
Total Processing Time [redacted] minutes
Total Number of Operators [redacted] operators
Processing rate [redacted] documents/min
Processing rate per Operator [redacted] document [redacted] documents/hour

	FormReader 2
	ARTS v1.32
	49
	1800
	80
	0.314
	200
	12

Operator cost per Hour
$15

Operator Idle Cost
[redacted]

		Actual
Physical Quality Pass Rate	67.00%	65.50%
Image Quality Pass Rate	60.00%	58.33%
Text Quality Pass Rate	80.00%	77.86%

Documents Input
Initial Number of Documents 2000 documents

Physical Processing Process (1)
Process Time 5 minutes
Initial Operators 15 operators
Processing Rate 3.00 documents/min

Digitisation Process (2)
Process Time 2 minutes
Initial Operators 4 operators
Processing Rate 2.00 documents/min

Image Quality Analysis Process (3)
Process Time 3 minutes
Initial Operators 8 operators
Processing Rate 2.67 documents/min

Image Storage Process (4)
Process Time 4 minutes
Initial Operators 6 operators
Processing Rate 1.50 documents/min

Text Recognition Process (5)
Process Time 2 minutes
Initial Operators 6 operators
Processing Rate 3.00 documents/min

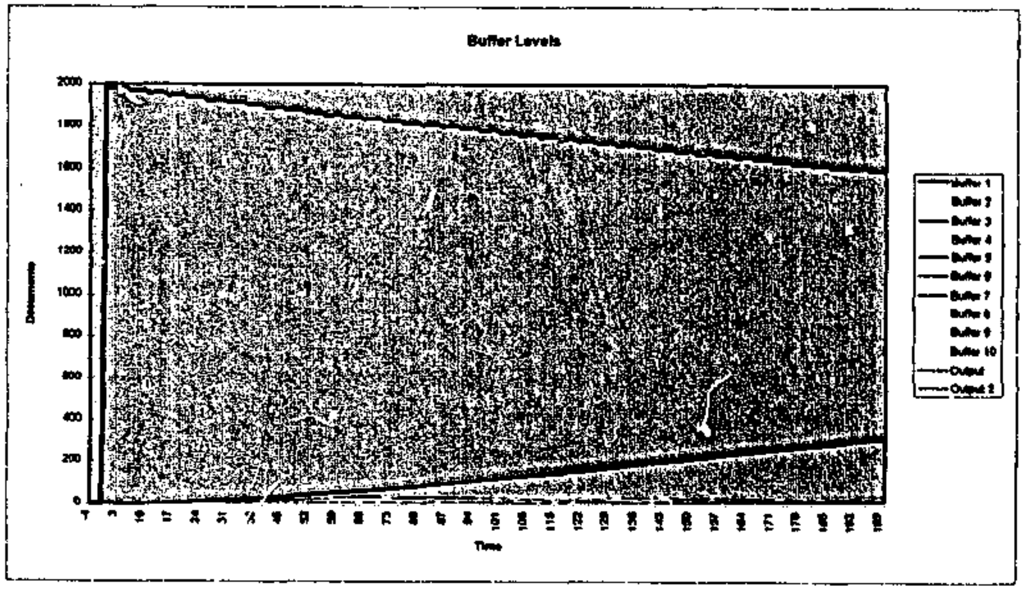
Text Quality Analysis Process (6)
Process Time 5 minutes
Initial Operators 15 operators
Processing Rate 3.00 documents/min

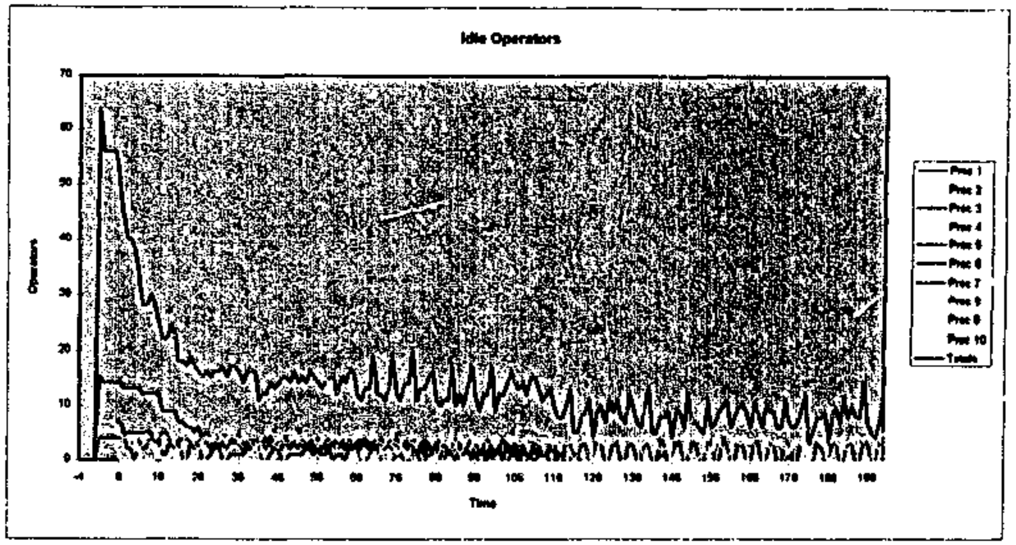
Text Storage Process (7)
Process Time 2 minutes
Initial Operators 4 operators
Processing Rate 2.00 documents/min

Physical Data Entry Process (8)
Process Time 6 minutes
Initial Operators 8 operators
Processing Rate 1.33 documents/min

Physical Storage Process (9)
Process Time 4 minutes
Initial Operators 6 operators
Processing Rate 1.50 documents/min

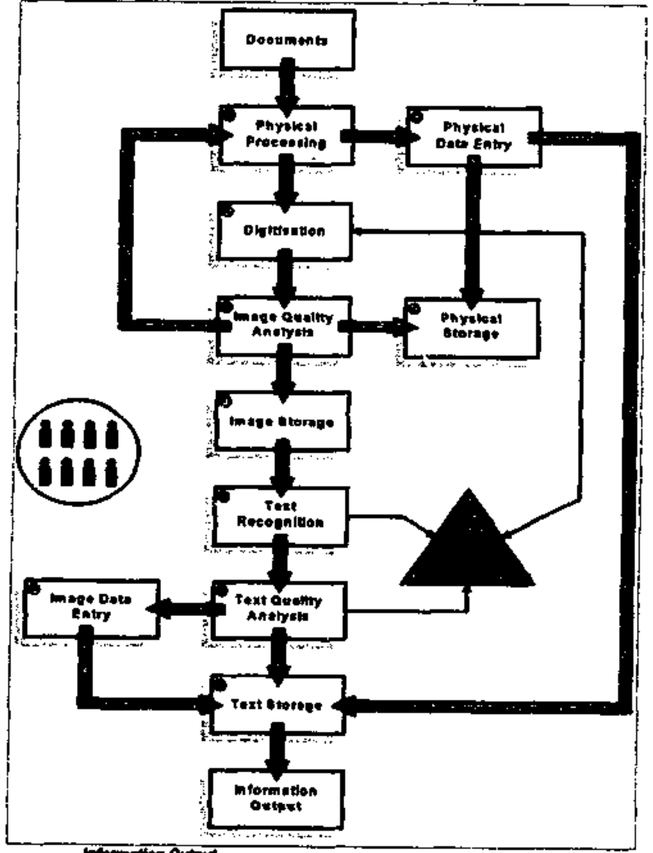
Image Data Entry Process (10)
Process Time 2 minutes
Initial Operators 6 operators
Processing Rate 3.00 documents/min


Buffer Levels


Idle Operators

	5	2	3	4	2	5	2	6	4	2
	15	4	8	6	8	15	4	8	6	6
	2000	0	0	0	0	0	0	0	0	0
	4	4	4	3	5	4	4	4	3	4
	30	20	20	20	20	20	20	20	20	20

Real world constraint HV OCR model V by Brian Griffin



Information Output
Total Processing Time [illegible] minutes
Total Number of Operators [illegible] operators
Processing rate [illegible] documents/min
Processing rate per Operator [illegible] document [illegible] documents/hour

[illegible]	[illegible]
[illegible]	FormReader 2
[illegible]	ARTS v1.32
[illegible]	49
[illegible]	1800
[illegible]	80
[illegible]	0.314
[illegible]	
[illegible]	200
[illegible]	12
[illegible]	

Operator cost per Hour
$15

Operator Idle Cost
[illegible]

		Actual
Physical Quality Pass Rate	87.00%	88.87%
Image Quality Pass Rate	60.00%	57.50%
Text Quality Pass Rate	80.00%	77.13%

Documents Input
Initial Number of Documents 2500 documents

Physical Processing Process (1)
Process Time 5 minutes
Initial Operators 15 operators
Processing Rate 3.00 documents/min

Digitisation Process (2)
Process Time 2 minutes
Initial Operators 4 operators
Processing Rate 2.00 documents/min

Image Quality Analysis Process (3)
Process Time 3 minutes
Initial Operators 8 operators
Processing Rate 2.67 documents/min

Image Storage Process (4)
Process Time 4 minutes
Initial Operators 6 operators
Processing Rate 1.50 documents/min

Text Recognition Process (5)
Process Time 2 minutes
Initial Operators 6 operators
Processing Rate 3.00 documents/min

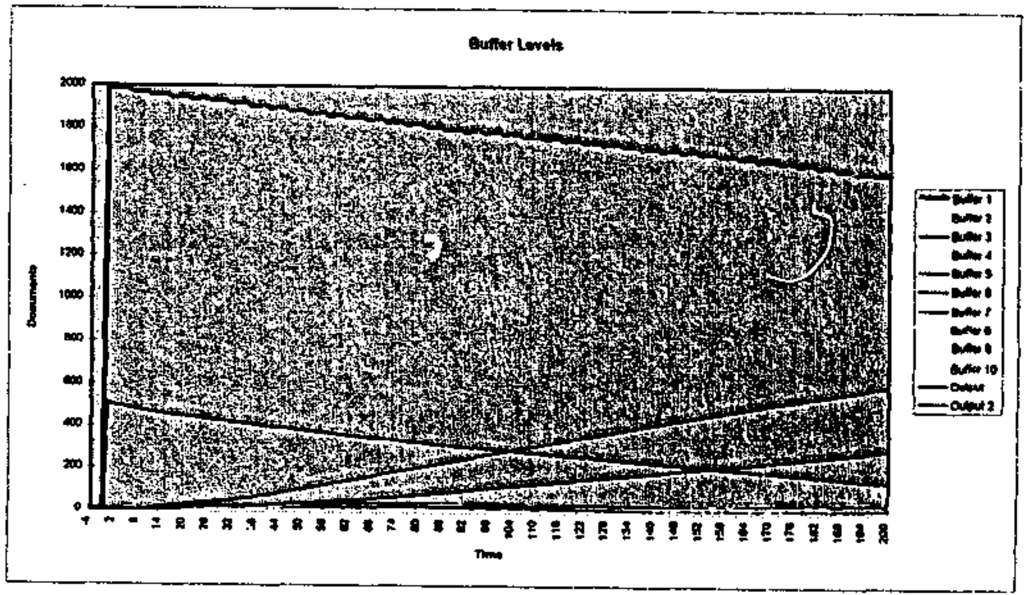
Text Quality Analysis Process (6)
Process Time 5 minutes
Initial Operators 15 operators
Processing Rate 3.00 documents/min

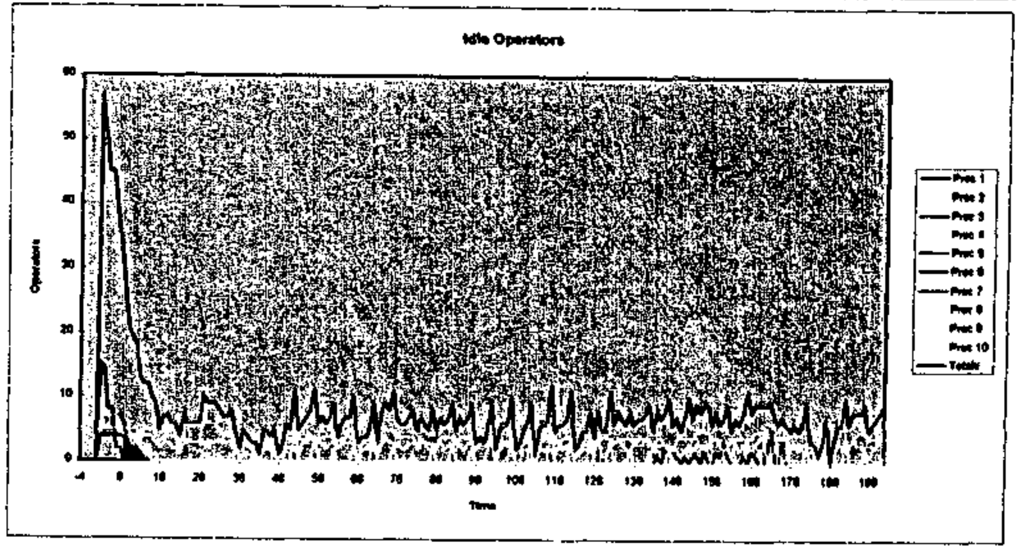
Text Storage Process (7)
Process Time 2 minutes
Initial Operators 4 operators
Processing Rate 2.00 documents/min

Physical Data Entry Process (8)
Process Time 6 minutes
Initial Operators 8 operators
Processing Rate 1.33 documents/min

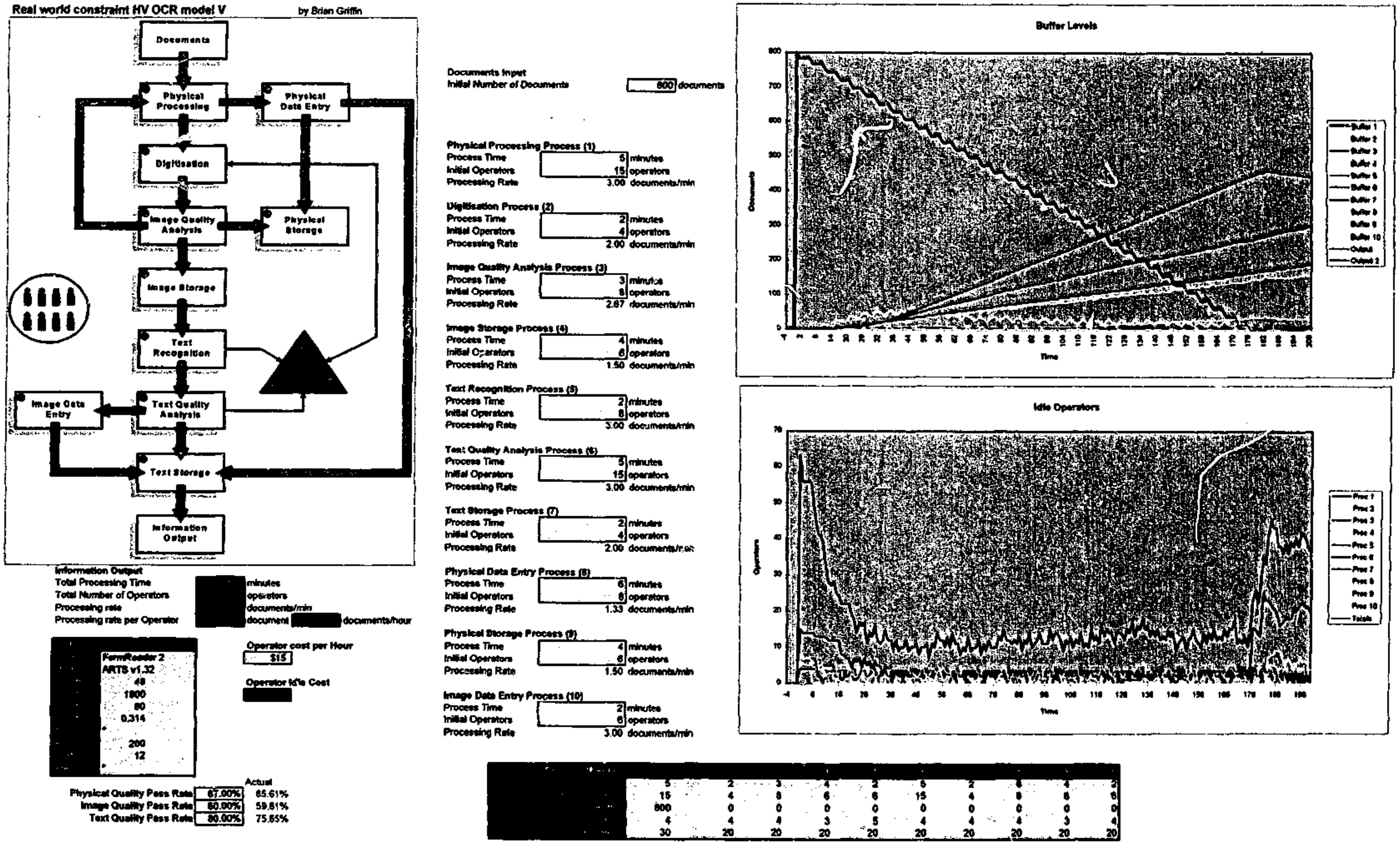
Physical Storage Process (9)
Process Time 4 minutes
Initial Operators 6 operators
Processing Rate 1.50 documents/min

Image Data Entry Process (10)
Process Time 2 minutes
Initial Operators 6 operators
Processing Rate 3.00 documents/min





[illegible]										
[illegible]	5	2	3	4	2	5	2	6	4	2
[illegible]	15	4	8	6	6	15	4	6	#	6
[illegible]	2000	0	0	0	500	0	0	0	0	0
[illegible]	4	4	4	3	5	4	4	4	3	4
[illegible]	30	20	20	20	20	20	20	20	20	20



Real world constraint HV OCR model V by Brian Griffin

Information Output
Total Processing Time minutes
Total Number of Operators operators
Processing rate documents/min
Processing rate per Operator document documents/hour

FormReader 2
ARTS v1.32
48
1800
60
0,314
200
12

Operator cost per Hour
$15

Operator Idle Cost

		Actual
Physical Quality Pass Rate	67.00%	65.61%
Image Quality Pass Rate	60.00%	59.61%
Text Quality Pass Rate	80.00%	75.65%

Documents Input
Initial Number of Documents 800 documents

Physical Processing Process (1)
Process Time 5 minutes
Initial Operators 15 operators
Processing Rate 3.00 documents/min

Digitisation Process (2)
Process Time 2 minutes
Initial Operators 4 operators
Processing Rate 2.00 documents/min

Image Quality Analysis Process (3)
Process Time 3 minutes
Initial Operators 8 operators
Processing Rate 2.67 documents/min

Image Storage Process (4)
Process Time 4 minutes
Initial Operators 6 operators
Processing Rate 1.50 documents/min

Text Recognition Process (5)
Process Time 2 minutes
Initial Operators 8 operators
Processing Rate 3.00 documents/min

Text Quality Analysis Process (6)
Process Time 5 minutes
Initial Operators 15 operators
Processing Rate 3.00 documents/min

Text Storage Process (7)
Process Time 2 minutes
Initial Operators 4 operators
Processing Rate 2.00 documents/min

Physical Data Entry Process (8)
Process Time 6 minutes
Initial Operators 8 operators
Processing Rate 1.33 documents/min

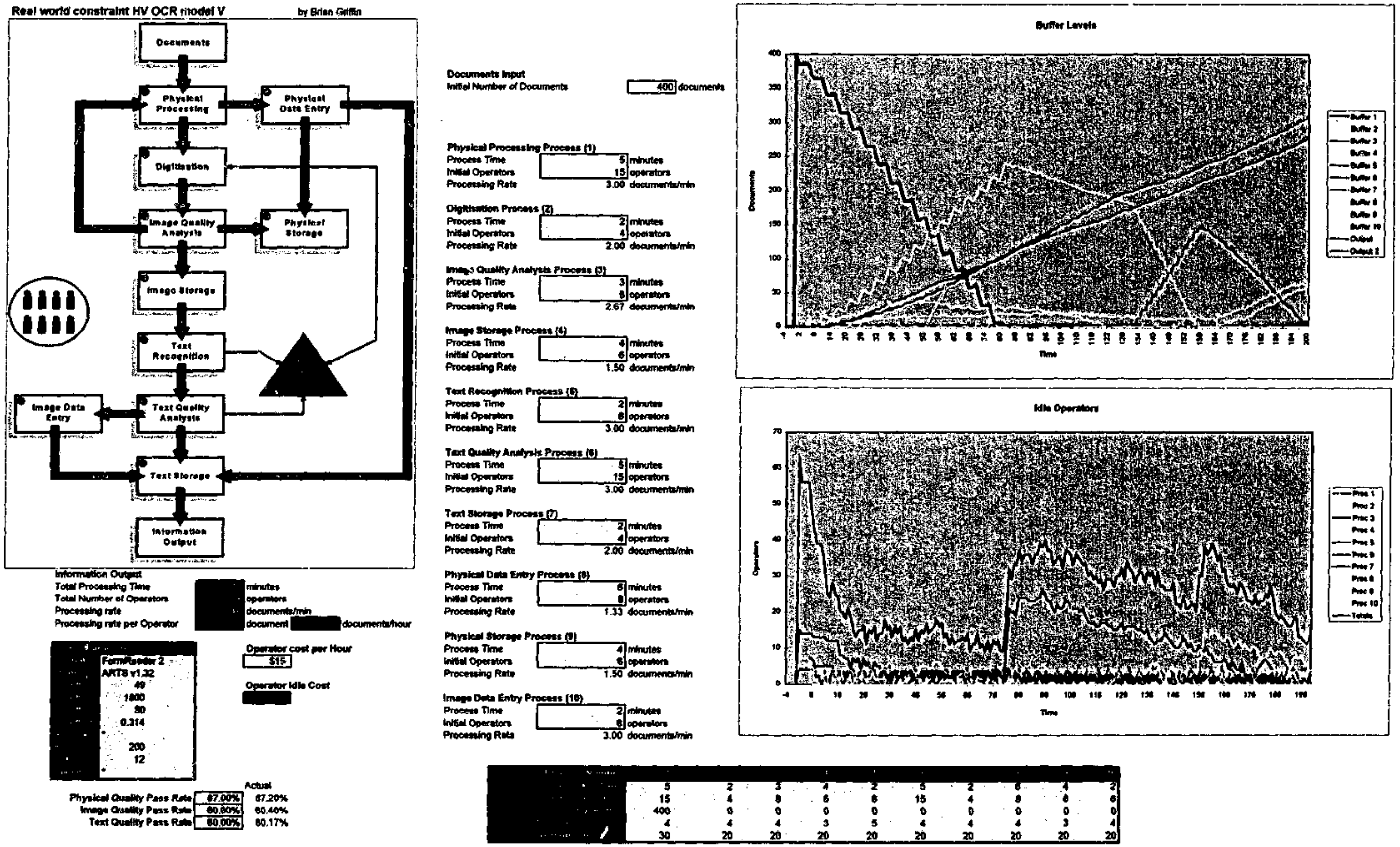
Physical Storage Process (9)
Process Time 4 minutes
Initial Operators 6 operators
Processing Rate 1.50 documents/min

Image Data Entry Process (10)
Process Time 2 minutes
Initial Operators 6 operators
Processing Rate 3.00 documents/min

Buffer Levels

Idle Operators

5	2	3	4	2	5	2	6	4	2
15	4	8	6	6	15	4	8	6	6
800	0	0	0	0	0	0	0	0	0
4	4	4	3	5	4	4	4	3	4
30	20	20	20	20	20	20	20	20	20



Real world constraint HV OCR model V by Brian Griffin

Information Output
Total Processing Time [redacted] minutes
Total Number of Operators [redacted] operators
Processing rate [redacted] documents/min
Processing rate per Operator [redacted] document [redacted] documents/hour

FormReader 2
ARTS v1.32
40
1800
80
0.314
200
12

Operator cost per Hour
$15

Operator Idle Cost
[redacted]

		Actual
Physical Quality Pass Rate	67.00%	67.20%
Image Quality Pass Rate	60.00%	60.40%
Text Quality Pass Rate	60.00%	60.17%

Documents Input
Initial Number of Documents 400 documents

Physical Processing Process (1)
Process Time 5 minutes
Initial Operators 15 operators
Processing Rate 3.00 documents/min

Digitisation Process (2)
Process Time 2 minutes
Initial Operators 4 operators
Processing Rate 2.00 documents/min

Image Quality Analysis Process (3)
Process Time 3 minutes
Initial Operators 8 operators
Processing Rate 2.67 documents/min

Image Storage Process (4)
Process Time 4 minutes
Initial Operators 6 operators
Processing Rate 1.50 documents/min

Text Recognition Process (5)
Process Time 2 minutes
Initial Operators 8 operators
Processing Rate 3.00 documents/min

Text Quality Analysis Process (6)
Process Time 5 minutes
Initial Operators 15 operators
Processing Rate 3.00 documents/min

Text Storage Process (7)
Process Time 2 minutes
Initial Operators 4 operators
Processing Rate 2.00 documents/min

Physical Data Entry Process (8)
Process Time 6 minutes
Initial Operators 8 operators
Processing Rate 1.33 documents/min

Physical Storage Process (9)
Process Time 4 minutes
Initial Operators 6 operators
Processing Rate 1.50 documents/min

Image Data Entry Process (10)
Process Time 2 minutes
Initial Operators 6 operators
Processing Rate 3.00 documents/min

5	2	3	4	2	5	2	6	4	2
15	4	8	6	8	15	4	8	6	6
400	0	0	0	0	0	0	0	0	0
4	4	4	3	5	4	4	4	3	4
30	20	20	20	20	20	20	20	20	20