

A Newspaper Cutting System

Ma Jie

Xi'an Jiaotong University, P.R.China

Nanyang Technology University, Singapore

1995.04

Table of Contents

ABSTRACT.....	I
ACKNOWLEDGMENTS	II
TABLE OF CONTENTS	III
LIST OF TABLES	V
LIST OF FIGURES	VI
1. INTRODUCTION	1
1.1 MOTIVATION	1
1.2 PROJECT OBJECTIVES	2
1.3 SCOPE OF PROJECT	2
1.4 PROJECT DEVELOPMENT CYCLE	3
1.5 ORGANIZATION OF THE REPORT	4
2. FUNCTIONAL SPECIFICATIONS AND SYSTEM REQUIREMENTS	6
2.1 FUNCTIONAL SPECIFICATIONS	6
2.2 SYSTEM REQUIREMENTS	8
3. SYSTEM ARCHITECTURE AND FUNCTIONAL COMPONENTS	10
3.1 SYSTEM ARCHITECTURE	10
3.2 THE CUTTING PROCESS	13
3.2.1 Cutting.....	14
3.2.2 Recognition	15
3.2.3 Image Compression	16
3.2.4 Indexing and Storage	17
3.3 THE RETRIEVAL PROCESS	18
3.3.1 Image Decompression.....	19
3.3.2 Retrieval.....	19
3.3.3 Newspaper Display	21
4. PROGRAM DESIGN AND IMPLEMENTATION	25
4.1 IMAGE CLASS	25
4.1.1 Data Member	26
4.1.2 Member Functions	27
4.2 NEWS CLASS	28
4.3 KEYWORD CLASS	29
4.4 OCR CLASS	30
4.4.1 Data Members.....	31
4.4.2 Member Functions	31
4.5 SET CLASS.....	33
4.5.1 Data Members.....	34
4.5.2 Member functions.....	34

4.6	DATABASE ACCESS	35
5.	PERFORMANCE ANALYSIS.....	38
5.1	COMPRESSION EFFICIENCY	38
5.2	STORAGE CAPACITY.....	38
6.	RECOMMENDATION AND FUTURE EXPANSION AREAS.....	39
6.1	NEWSPAPER CUTTING	39
6.2	AUTOMATIC IDENTIFICATION OF NEWSPAPER ARTICLES	39
6.3	INDEXING	39
6.4	IMAGE COMPRESSION.....	40
6.5	NEWSPAPER ARTICLE RETRIEVAL.....	40
7.	CONCLUSION	41
	REFERENCE	42
	APPENDIX A GRAPHICS FILE FORMAT	A1
	APPENDIX B CALERA SMART HOST LIBRARIES.....	A5
	APPENDIX C HEADER FILES	A9

List of Tables

TABLE 4.1 STRUCTURE OF KEYWORD.DBF	36
TABLE 4.2 STRUCTURE OF NEWS.DBF	37
TABLE 5.1 COMPRESSION RATIOS	38

List of Figures

FIGURE 1.1 SCHEDULE FOR THE DEVELOPMENT OF THE NEWSPAPER CUTTING SYSTEM	4
FIGURE 3.1 SYSTEM ARCHITECTURE.....	10
FIGURE 3.2 THE FUNCTIONAL COMPONENTS OF THE NEWSPAPER CUTTING SYSTEM	12
FIGURE 3.3 NEWSPAPER CUTTING SYSTEM MENU	13
FIGURE 3.4 CUTTING: AN ENGLISH ARTICLE.....	14
FIGURE 3.5 CUTTING: A CHINESE ARTICLE	15
FIGURE 3.6 RECOGNITION COMPLETE WINDOW.....	15
FIGURE 3.7 SHOW TEXT WINDOW.....	16
FIGURE 3.8 INDEXING DIALOG	17
FIGURE 3.9 NEWSPAPER RETRIEVAL SYSTEM MENU	18
FIGURE 3.10 KEYWORD RETRIEVAL DIALOG	21
FIGURE 3.11 DATE RETRIEVAL DIALOG.....	22
FIGURE 3.12 AUTHOR RETRIEVAL DIALOG	22
FIGURE 3.13 RETRIEVAL DIALOG.....	23
FIGURE 3.14 IMAGE WINDOW	24
FIGURE 4.1 RELATIONSHIPS AMONG FUNCTIONAL COMPONENTS AND CLASSES	26
FIGURE 4.2 TIFF FILE FORMAT	27
FIGURE 4.3 DIB FILE FORMAT.....	28
FIGURE 4.4 PROCESS OF THE FUNCTION RECOGNITION()	33
FIGURE 4.5 COMPONENTS OF ODBC	36

CHAPTER 1

1. Introduction

1.1 Motivation

Newspapers are very widely used as a media for transmitting news and information in today's society. News are printed on paper for reading and distribution. Different types of news including home and international news, sports, financial news and even advertisements are printed on newspaper everyday. Some of these news are very important and are required to be kept for future reference. Although newspaper publishing companies have their own libraries to store newspapers both in electronic and paper forms, they usually store them in its original form without any facilities for selective retrieval. To store the original newspapers is deemed to be impractical as it takes up a lot of storage even with compression. As only some articles are of interest to the readers, there should be more a more effective method to keep these articles. In addition, each newspaper publishing company will keep its own newspaper only. Related articles on the same news topic from different newspapers are hard to obtain.

Libraries (university or public) are required to be responsible for keeping and maintaining newspaper articles. With this function provided by libraries, readers can simply retrieve newspaper articles. At the university library of Nanyang Technological University, newspaper articles are read, selected, cut, indexed and filed by a senior librarian manually everyday. The newspapers considered for cutting include local newspapers (Straits Times and LianHe) and some foreign newspapers. These newspapers could be printed in English, Chinese or other languages. Currently, a paper cataloging and filing system is used to store all these newspaper articles. Newspaper articles are indexed in English. The indexing process is tedious and requires human expertise. The retrieval process is also very troublesome. The reader needs to go through a manual card indexing mechanism in order to retrieve a newspaper article. Related articles based on the same topic or keywords from different newspapers are able to be retrieved. However, as time goes by, more newspaper articles are cut and filed and the size of the filing system grows, it makes the manual system more difficult to maintain.

With the advance in computer technology, the concept of the “paperless library” evolves and information can be processed via computers. Therefore, it is necessary to automate the newspaper cutting process and handle newspaper article cuttings in an electronic way. The motivation of this project was to develop an interactive system to cut, index and file newspaper articles automatically.

1.2 Project Objectives

The objective of this project is to design and develop a Newspaper Cutting System. The system must be able to capture and display newspaper images. User interface with some functions to allow the user to manipulate the displayed newspaper article images must be included. Functions should also be provided for the user to define the areas of the newspaper articles to be cut.

The Newspaper Cutting System must be able to identify the selected newspaper articles intelligently inside the cut area, convert the English newspaper article images into text format, understand newspaper articles by heading and body, index newspaper articles automatically and finally store the newspaper articles in a database.

In addition, the Newspaper Cutting System must also be able to compress and decompress non-English newspaper article images. The purpose of the image compression is to reduce the size of the newspaper article images, and thus to minimize the storage capacity. Decompression is needed to restore the compressed newspaper article image so as to display it on the computer screen.

Finally, the Newspaper Cutting System must be able to provide a retrieval interface, through which users can formulate queries to retrieve the stored newspaper articles.

1.3 Scope of Project

This project is a collaboration between the School of Applied Science and the Nanyang Technological University Library. The university library provides the problem scenario. The system specifications for this project are defined by the author. The developed system will be used by the university library for newspaper article cuttings.

This is a 100% software project. There is no hardware design or implementation for this project. The Newspaper Cutting System is designed and developed in

such a way that it is targeted to be used as part of a library information system. This project concentrates on the newspaper cutting, recognizing, indexing and retrieval. No capturing of the newspaper article images is included in this project. Any design of device driver for scanner to capture newspaper article images is beyond the scope of this project. It is developed under Windows 3.1 environment using Visual C++ programming language [1]. The Foxpro database management system [2] is used to store newspaper articles.

1.4 Project Development Cycle

This project is scheduled for completion within a period of four months. Figure 1.1 shows the schedule for the development of this project.

	March				April					May				June			
Requirements	1	2	3	4	1	2	3	4	5	1	2	3	4	1	2	3	Results
Installation and familiarisation																	
Visual C++ 1.5	√	√	√														Familiar with all
Foxpro 2.6			√														the software that
Calera Smart Host Libraries			√	√	√												to be used for the
																	development of
																	the project.
Requirements and specification analysis																	
Overall Newspaper Cutting System			√	√	√	√											Specifications and
Cutting component			√	√													requirements are
Retrieval component					√	√											well defined.
Planning and design																	
Overall Newspaper Cutting System						√	√	√	√	√							System design
Function components																	
1.Cutting Module							√										Module design
2.Recognition Module							√	√									Module design
3.Indexing Module								√									Module design
4.Image Compression								√									Module design
5.Decompression								√									Module design
6.Retrieval Module									√	√							Module design
7.Newspaper Display										√							Module design

	March				April					May				June				
Requirements	1	2	3	4	1	2	3	4	5	1	2	3	4	1	2	3	Results	
Implementation																		
Object																		
1.Image Object								√									Object completed	
2.Database Object									√								Object completed	
3.OCR Object									√								Object completed	
4.Set Object										√	√						Object completed	
Functional Module																		
1.Cutting Module									√								Module completed	
2.Recognition Module										√							Module completed	
3.Indexing Module										√							Module completed	
4.Image Compression										√							Module completed	
5.Decompression										√							Module completed	
6.Retrieval Module											√	√					Module completed	
7.Newspaper Display											√	√					Module completed	
Documentation								√	√	√	√	√	√	√	√	√	Documentation	

Figure 1.1 Schedule for the development of the Newspaper Cutting System

1.5 Organization of the Report

The introductory chapter, Chapter One, of this report gives the motivation for the development of the Newspaper Cutting System. This chapter also gives the objectives and scope of the project.

Chapter Two describes the functional specifications and the system requirements of the Newspaper Cutting System.

Chapter Three gives a view of the system architecture and the functional components of the Newspaper Cutting System.

Chapter Four provides the design and implementation issues for the Newspaper Cutting System. The approaches used to display, cut, identify, recognize, index and retrieve the newspaper articles are explained in this chapter.

Chapter Five contains results of the analysis of the performance of the system. This chapter measures the compression efficiency for non-English newspaper articles. It also discusses the user feedback of the implemented system.

Chapter Six provides some recommendations to the system. Some areas of the Newspaper Cutting System are highlighted for future enhancement.

The concluding chapter, Chapter Seven, gives a review of the Newspaper Cutting System.

CHAPTER 2

2. Functional Specifications And System Requirements

2.1 Functional Specifications

The purpose of the Newspaper Cutting System is to support users to manage newspaper article cuttings. The system supports librarians to cut, index and store articles from newspapers and readers to retrieve newspaper articles from the system. To achieve this, the Newspaper Cutting System must consist of a number of functional components. There must be a component to display, manipulate and cut newspaper article images. A component for automatic identification of newspaper articles within the cut area should be supported. Indexing, storage and retrieval components are also needed. Finally, a text recognition component is required to convert the article image into characters. This component not only can recognize the articles to minimize the storage capacity, but also make the automatic indexing possible.

The capabilities of the various functional components are explained below:

- Capture, Display and Cutting

When a newspaper is scanned in, its image is captured and displayed onto the screen. The captured newspaper image must be pre-processed before further processing. Therefore, there must be some facilities to perform noise removal, skew correction and some manipulation operations such as shrinking and zooming. In addition, users should be able to define cut areas of any desired newspaper articles interactively from the displayed newspaper image. Users can use a rectangular box or a polygon to enclose the cut area. In addition, several articles are allowed to be enclosed within one cut area.

- Automatic Article Identification

FUNCTIONAL SPECIFICATIONS AND SYSTEM REQUIREMENTS

Once a cut area is defined from the scanned newspaper, the system needs to identify newspaper articles within the cut area. In general, as some articles are printed in different shapes (e.g. T shape, L shape, etc.) on the newspaper, it is difficult for users to cut the area using a rectangular box which just encloses one article. Sometimes, a cut area encloses more than one article. Therefore, automatic article identification within the cut area should be supported.

- Text Recognition

Newspaper articles, in general, consist of text description and picture. Newspaper article images are large in size and so takes up a large amount of storage space. Moreover, it is difficult to process automatic indexing. The main function of this component is to segment and classify the newspaper article image into text and picture blocks. The text blocks are converted into text data using Optical Character Recognition (OCR). The picture blocks are saved as image data. The original newspaper article can be reconstructed using the text and image data. After this step, the size of the newspaper articles are much smaller than the original captured images. In addition, automatic indexing is possible.

- Indexing

This component provides automatic or manual indexing methods. In automatic indexing, keywords are identified and indexed from the articles during the cutting process. Users can also index the newspaper article manually. For non-English newspaper articles, only manual indexing is supported.

- Storage

There will be a large number of newspaper articles stored in the system. English newspapers are recognized and stored, while non-English newspapers are stored in its original image forms. The non-English newspaper articles must be compressed such that the storage space can be minimized. A suitable compression process must be available.

- Retrieval

Newspaper articles can be retrieved by readers or librarians. This component makes it easier for users to access newspaper articles. By using the retrieval

FUNCTIONAL SPECIFICATIONS AND SYSTEM REQUIREMENTS

component provided, users can formulate queries to retrieve the stored articles in an efficient manner.

2.2 System Requirements

The Newspaper Cutting System must be able to satisfy the following requirements:

- **Multimedia Objects Support**

The system should be able to support various information objects within a newspaper article such as text, image and graphics.

- **Networking Support**

The system must be able to transmit and receive the newspaper articles over a campus wide network. A client-server architecture should be adopted.

- **Friendly User Interface**

The user interface component in any system determines the quality of the level of interaction between the human and the computer. This involves enabling the non-specialists to feel at ease with the system and equipping the users with a set of different but simple media tools. A friendly and easy-to-use user interface must be available. The librarian should be able to cut, index and store the newspaper articles. The reader should be able to retrieve the newspaper articles through the interface.

- **Efficient Multimedia Storage Support**

The system should be able to satisfy the enormous storage requirements for newspaper articles. Recognized and compressed data should be efficiently stored. The database for newspaper articles is often located at the server of the network. When a reader formulates a query, the selected articles in compressed forms are transmitted through the network to the terminal for display. The compressed algorithm that is used must be reliable and efficient.

In view of the above system requirements, an object-oriented approach is deemed more suitable in providing the storage, modeling and retrieval support a newspaper cutting system required. An object-oriented approach emphasizes

FUNCTIONAL SPECIFICATIONS AND SYSTEM REQUIREMENTS

objects as the unit of access and manipulation. Methods that are applied to objects are well-defined. An object-oriented interface can be constructed to allow the user to point to an object instance to find out the methods applicable to the object. Thus, it provides mechanisms to uniformly define, create and relate objects and object interactions.

CHAPTER 3

3. System Architecture And Functional Components

3.1 System Architecture

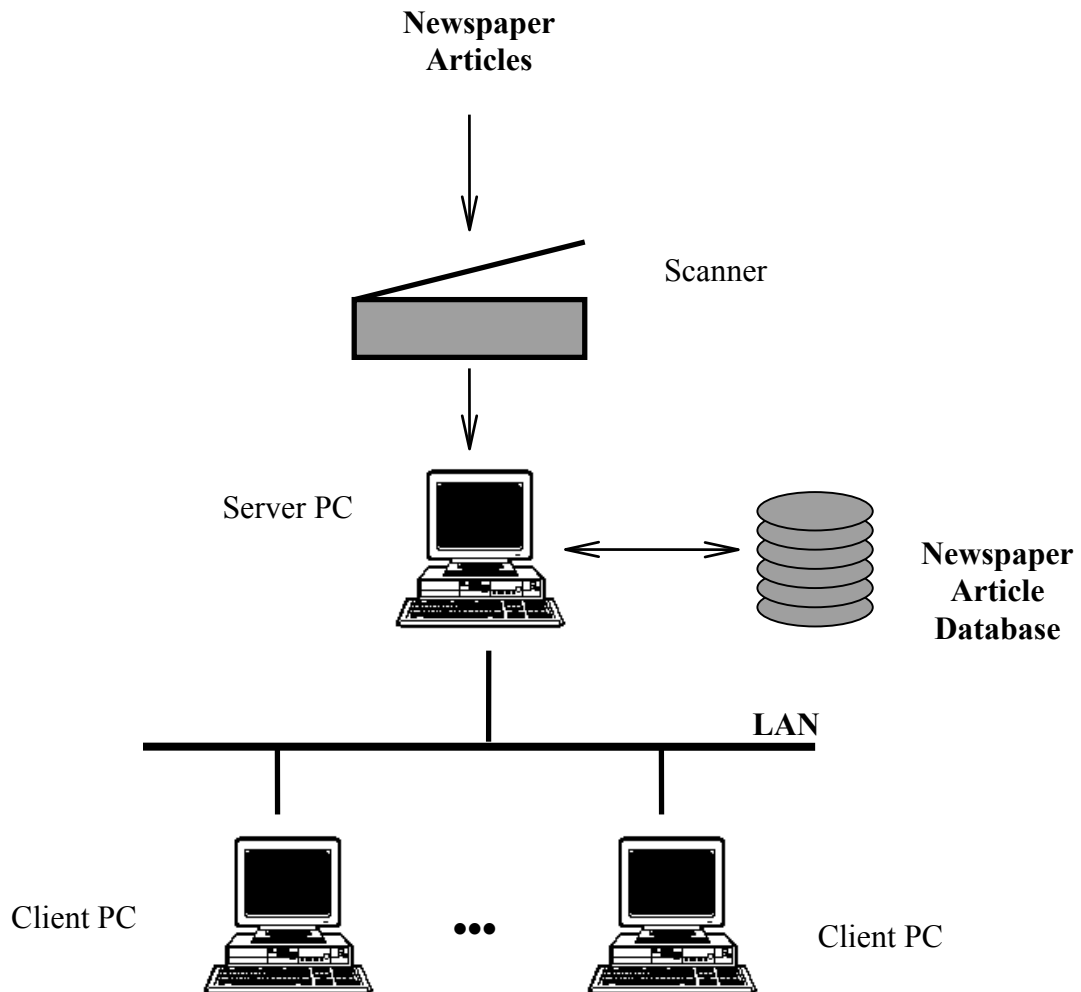


Figure 3.1 System Architecture

SYSTEM ARCHITECTURE AND FUNCTIONAL COMPONENTS

Figure 3.1 shows the client-server system architecture of the Newspaper Cutting System. It consists of a server PC that is used to cut, index and store the newspaper articles into a database. Each client PC is used to retrieve the newspaper articles. Both English and Chinese newspapers can be scanned in, cut and stored in the system. English newspaper articles will be recognized into text format, indexed automatically and finally stored in the database. Chinese newspaper articles will be stored as images. In this case, image compression is used to reduce the size of the newspaper article images. Photographs in English newspaper articles are also compressed before they are stored into the database. All the articles, both English and Chinese, are indexed by English keywords. In the retrieval user interface, readers can retrieve both English and Chinese newspaper articles. The retrieval queries can contain the Boolean operations such as NOT, AND and OR.

Figure 3.2 shows the various functional components of the Newspaper Cutting System. The system is divided into two parts - the Server System and the Client System. The Server System provides the cutting, indexing and storage functions. The Client System provides the retrieval function. These two parts consist of several functional components: Cutting, Recognition, Indexing, Image Compression, Image Decompression, Retrieval and Display.

To cut a newspaper article, the first step is to define a cutting region of the newspaper article. During the cutting process, a newspaper image is displayed on the Server PC 抐 screen. Librarians can define a cutting region by using several rectangular boxes. Each of the rectangular box identifies one component of the article, such as the title, columns and photographs. If Chinese newspaper articles are cut, it will only need to define one rectangular box to enclose the whole article. After this cutting process, English newspaper articles will go through a recognition process to convert the article image into text data. Then, the next step is indexing. The librarians can input the author, date and keywords of the newspaper article. The title of the article is automatically identified during the recognition step. All the indexes, text content and photographs (or article images) are stored into the database. Photographs and article images are compressed before they are stored into the database.

To retrieve a newspaper article, readers need to enter a retrieval query in the retrieval dialog boxes. Then the system will list the newspaper article titles corresponding to the queries. Readers can select one of them to view the text and its photographs.

SYSTEM ARCHITECTURE AND FUNCTIONAL COMPONENTS

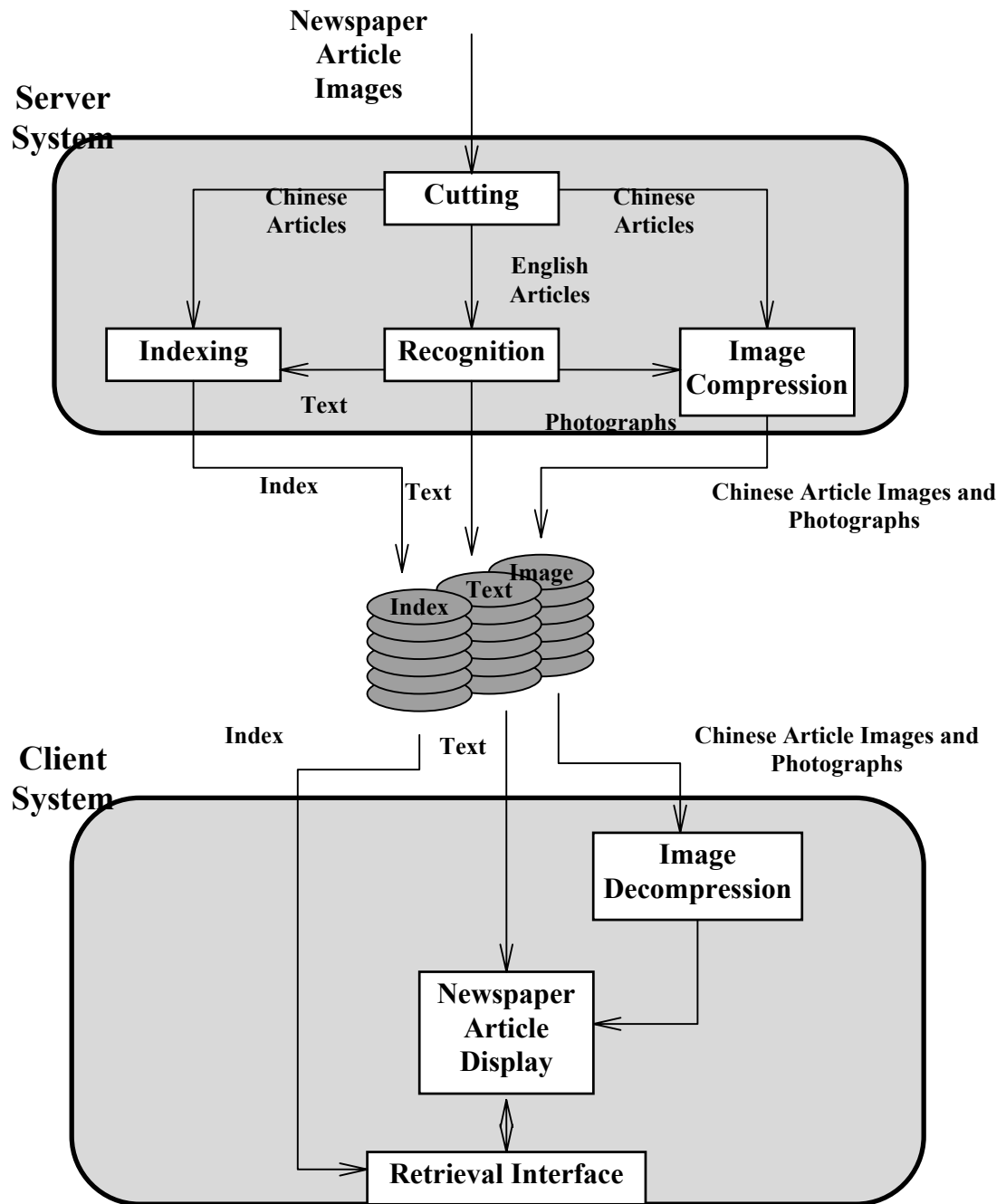


Figure 3.2 The Functional Components of the Newspaper Cutting System

3.2 The Cutting Process

The Newspaper Cutting System provides a user-friendly user interface. This system is an application running under the Windows environment. Like other Windows applications, the system provides menus and dialog boxes for the users to call the required facility.

Figure 3.3 shows the newspaper cutting menu. By using this menu, the librarians can cut and index the newspaper articles. The first step of the cutting process is to open a newspaper image by choosing the 摺pen” menu item. Then the newspaper image will be shown inside a window. The librarian can reduce the size of the image by selecting the percentages listed in the 摺view” menu. After defining the cutting zones by using a mouse, the librarian can recognize the image by using the 摺CR” menu. Then the librarian can enter the title, author, date and keywords in the cutting dialog box shown in section 3.2.4. In addition, the librarian can define the stoplist by selecting the 摺toplist” menu item.

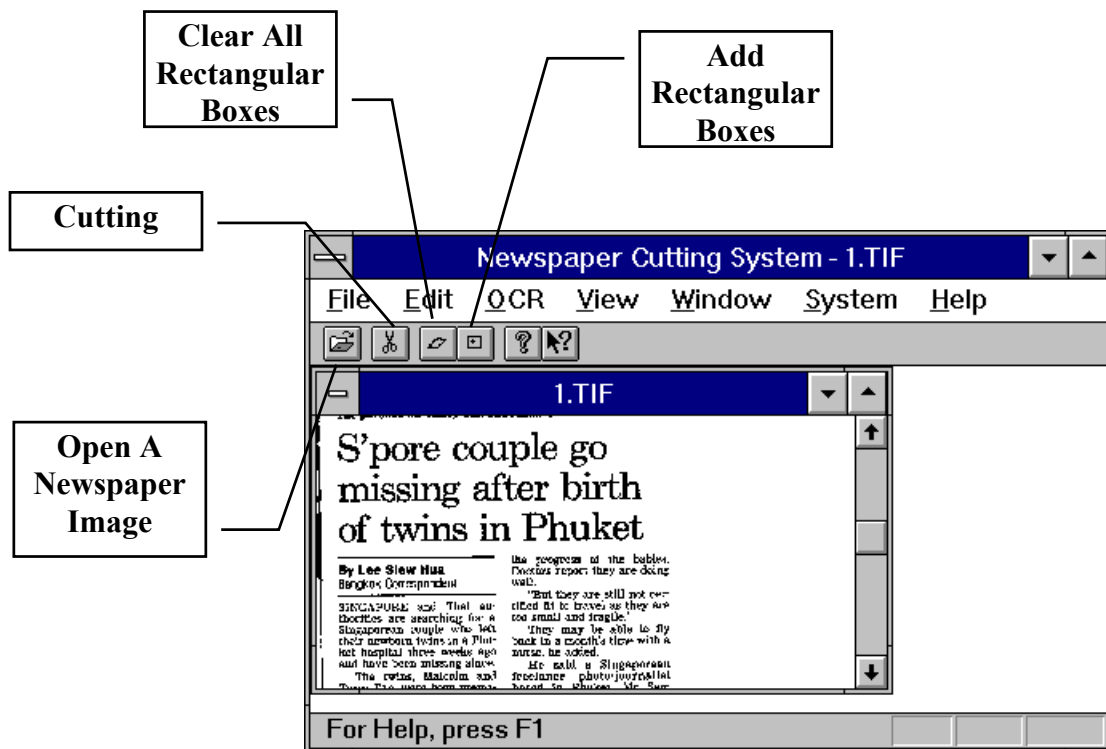


Figure 3.3 Newspaper Cutting System Menu

3.2.1 Cutting

This function includes two components. One is the image display component, and the other is the cutting component. The image display component provides the image I/O, display, zoom and print functions. The cutting component allows the librarian to identify the cutting area. A newspaper page is structurally composed of several rectangular blocks corresponding to title, text columns and photographs. The newspaper article area is defined as a set of rectangles. The librarian can define one or more rectangular boxes to identify the cutting area, so that the T Shape, L Shape or other shapes of the printed articles can be enclosed into one cutting area. These series of rectangular boxes are defined as a list. To define a rectangular box, the librarian can press the left button of the mouse at the left-top point of the rectangular box, move the mouse to the right-bottom point of the rectangular box and release the left button. Then this rectangle can be added to the list. In this way, any shape of articles can be cut.



Figure 3.4 Cutting: An English Article

Figure 3.4 shows an example of a cutting. There are three rectangular boxes defined in this example. First one indicates the title of the newspaper article; the other two rectangular boxes contain the two columns of this article. In addition,

the photograph is included in another rectangular box. Figure 3.5 shows another example of the cutting process. A Chinese article is cut in this example. The whole article is defined in one rectangular box.

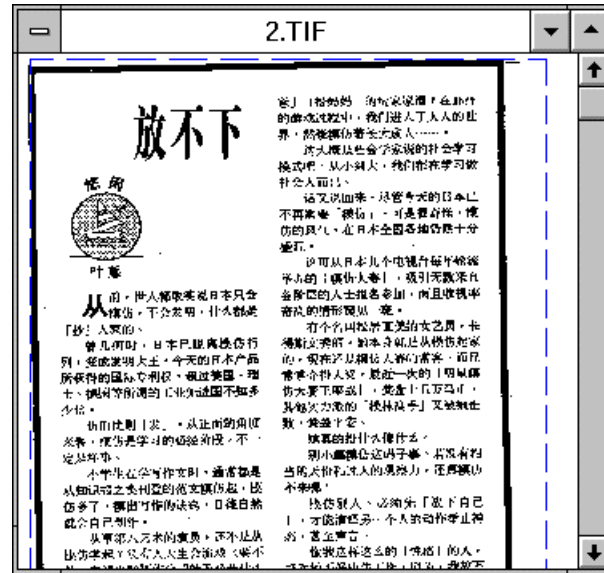


Figure 3.5 Cutting: A Chinese Article

3.2.2 Recognition

This function is used to recognize the newspaper article images into text format. In the system, the Calera Smart Host Libraries [3] is used to recognize English newspaper articles. Non-English newspaper articles will be stored as images. Calera is a set of libraries. It provides many functions for the recognition of English image into text format.



Figure 3.6 Recognition Complete Window

SYSTEM ARCHITECTURE AND FUNCTIONAL COMPONENTS

In the system, when the OCR menu item is selected, text recognition is executed automatically. After the recognition is completed, the message window (Figure 3.6) will be shown on the screen. The librarian can then view the text (see Figure 3.7) by selecting the 摺how Text” menu item under the 搨CR” menu.

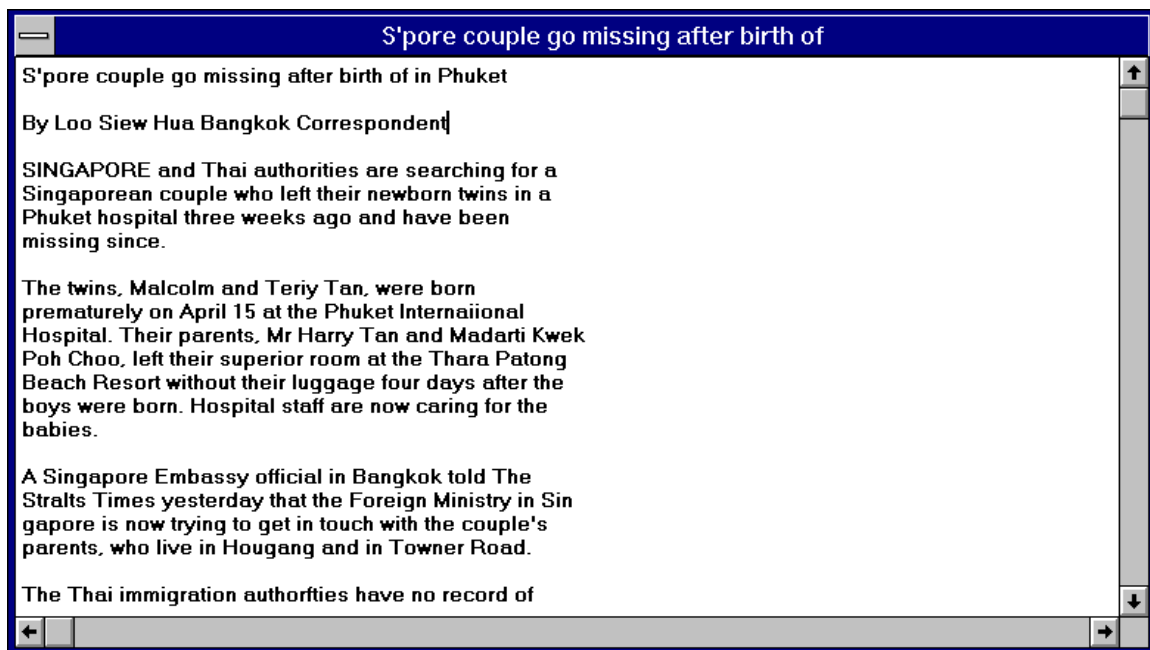


Figure 3.7 Show Text Window

3.2.3 Image Compression

This function is to compress the newspaper article images to reduce the size of the images. This function compresses the newspaper article image using the Huffman algorithm. After the compression of the newspaper article image, the compressed image is then stored into the database.

As shown in Figure 3.2, non-English newspaper articles are stored as images. The pictures and photos in English newspaper articles are also stored as images. In order to reduce the size of the images, this image compression function is used when the image is cut.

A newspaper image captured by using a scanner is a two-level (1 bit per pixel) image. Like most two-level images, the newspaper image does not have equiprobable levels. It has a higher probability for a white pixel (background) than a black pixel (text or graphics). In this system, the choice of the Huffman

coding is due to its simplicity. Many other image data compression algorithms can be applied.

3.2.4 Indexing and Storage

The indexing process provides an interface to index newspaper articles. The main task of this process is to get the article title, author, date and keywords. All this information will be stored in the database. The indexing interface is shown in Figure 3.8.

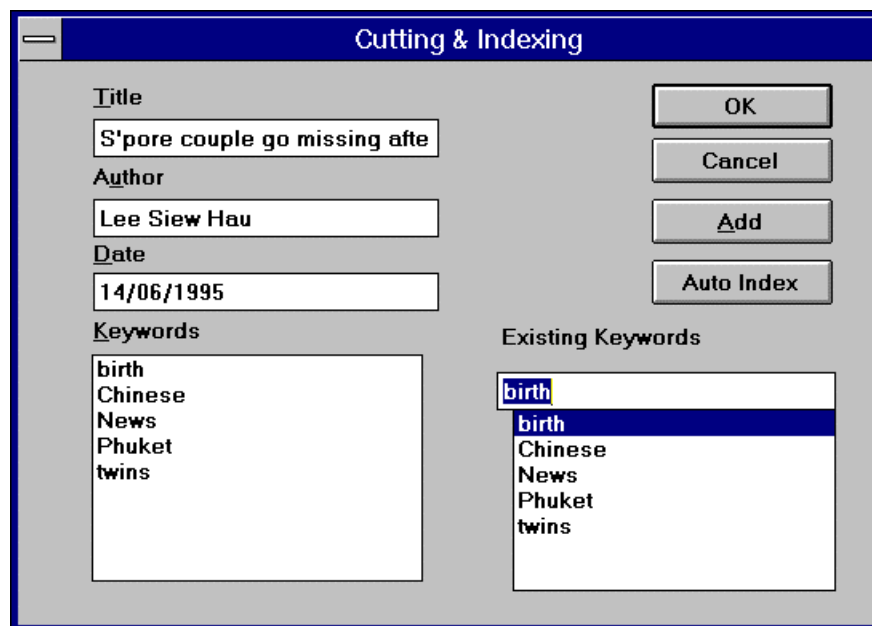


Figure 3.8 Indexing Dialog

For English newspaper articles, automatic indexing is supported. To implement automatic indexing, a lexical analyzer and a stoplist are used. Lexical analysis is the process of converting an input stream of characters into a stream of words or tokens. Tokens are groups of characters with collective significance. Lexical analysis is the first stage of automatic indexing. The lexical analysis phase produces candidate index terms that may be further processed, and eventually added to indexes. In this system, candidate index terms are checked to see whether they are in the stoplist. Stoplist words are known to make poor index terms, and they are immediately removed from further consideration as index terms when they are identified. For Chinese and non-English newspaper articles, indexing can only be done manually. The librarians should input the keywords by themselves.

In the system, a simple lexical analyzer is implemented. The lexical analyzer only breaks the input stream of characters into stream of words. A predefined stoplist is used to filter the words and generate the candidate index words. Users can choose the words listed in the “keyword” list as the index words and add to the indexes.

3.3 The Retrieval Process

Figure 3.9 shows the newspaper retrieval menu. In the retrieval submenu, three retrieval methods are supported. The retrieval dialog boxes are shown in Figure 3.10 - 3.12.

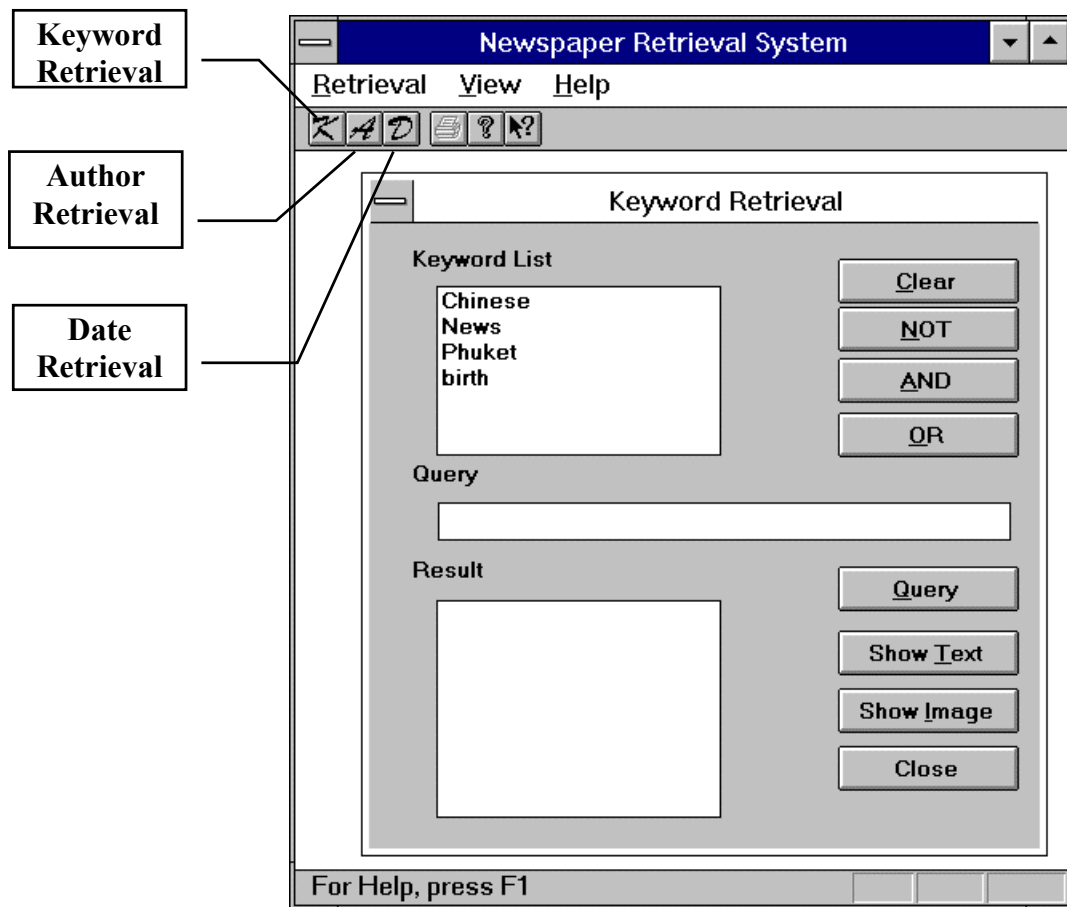


Figure 3.9 Newspaper Retrieval System Menu

3.3.1 Image Decompression

This function is the reverse of the Image Compression function. It accepts a compressed newspaper article image from the database and decompresses it. The decompressed newspaper article image is then output to the screen.

3.3.2 Retrieval

The main function is to provide an interface to retrieve the stored articles. Newspaper articles can be retrieved in many ways, such as keyword retrieval, date retrieval and author retrieval. Open Database Connectivity (ODBC) [4] is used in this module to access the data source independently.

User requests are typically phrased in terms of Boolean operations. Boolean expressions are formed from user queries. The reader enters them directly. They represent a request to determine what documents contain (or do not contain) a given set of keywords. For example

Find all documents containing 搠nformation”

is a query that, when evaluated, should yield a (possibly empty) set of documents each of which contains the word 搠nformation” somewhere within its body. However, such a sentence contains redundant words (words which do not contribute to the key meaning of the query). By definition a query searches a set of documents to determine their content. The above is therefore usually represented as the Boolean expression:

information

which means, 搠 set whose elements are the names of all documents containing the pattern 搠nformation’.”

Boolean expressions may be formed from other Boolean expressions to yield rather complex structures. Consider the following query:

Find all documents containing 搠nformation”, 搠etrieval” or not containing both 搠etrieval” and 搠cience”.

This translates into the following Boolean expression:

SYSTEM ARCHITECTURE AND FUNCTIONAL COMPONENTS

(information and retrieval) or not (retrieval and science)

Parentheses are often helpful to avoid ambiguity.

Each portion of a Boolean expression yields a set of documents. These portions are evaluated separately. That is, all documents containing information yields a set of documents D_1 , and all documents containing retrieval yields some set D_2 . The system will combine these two sets to yield the set D_3 that contains only both information and retrieval.

Combining the terms of Boolean expressions is conceptually quite simple. It involves sequences of familiar set operations. Let U represent the names of all documents stored. Let D_1 and D_2 represent the names of those documents that contain patterns P_1 and P_2 respectively. The following list defines how to evaluate Boolean expression operators in terms of the sets.

- $U - D_1$ is the set of all documents not containing P_1 (not)
- $D_1 \cap D_2$ is the set of all documents containing both P_1 and P_2 (and)
- $D_1 \cup D_2$ is the set of all documents containing either P_1 or P_2 (or)
- $D_1 \cup D_2 - D_1 \cap D_2$ is the set of all documents containing either P_1 or P_2 , but not both (xor)

The Set Class (in Chapter 4) provides the typical set operations, such as union, intersection and different. Other basic set operations, such as insert or delete an element, are also implemented in this class. More details will be discussed in the next chapter.

In keyword retrieval, users can select keywords from the keyword list, and connect them with the Boolean operator NOT, AND, OR. By pressing the Query button, the system shows all the article titles which are selected, and users can display its content. Figure 3.10 shows the keyword retrieval dialog.

SYSTEM ARCHITECTURE AND FUNCTIONAL COMPONENTS

In date retrieval, users can key in two dates, and the results are the article titles between the two dates. If the first date or the last date is ignored, the results are the article titles after or before the date. Figure 3.11 shows the date retrieval dialog.

In author retrieval, users can select an author from the author list, and the results are shown in the title list. Figure 3.12 shows the author retrieval dialog.

Another retrieval method is shown in Figure 3.13, which combines the date retrieval, the author retrieval and the keyword retrieval.

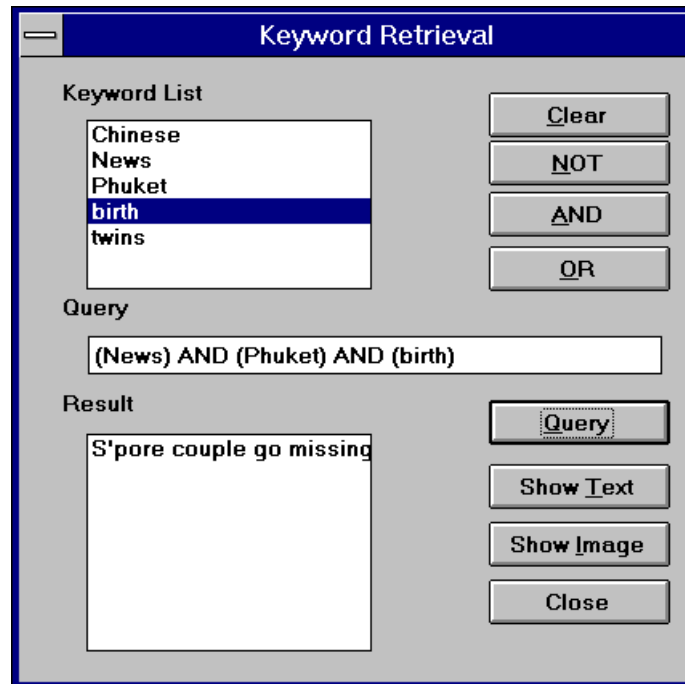


Figure 3.10 Keyword Retrieval Dialog

3.3.3 Newspaper Display

The main function is to display the newspaper articles, both the article images and texts, onto the screen of the Client PC.

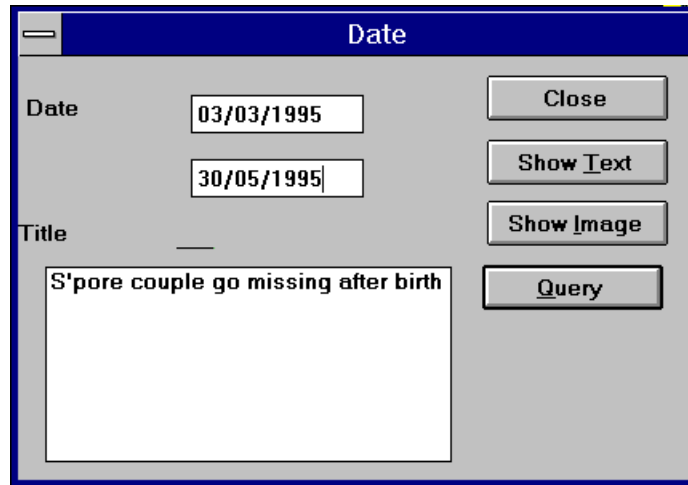


Figure 3.11 Date Retrieval Dialog

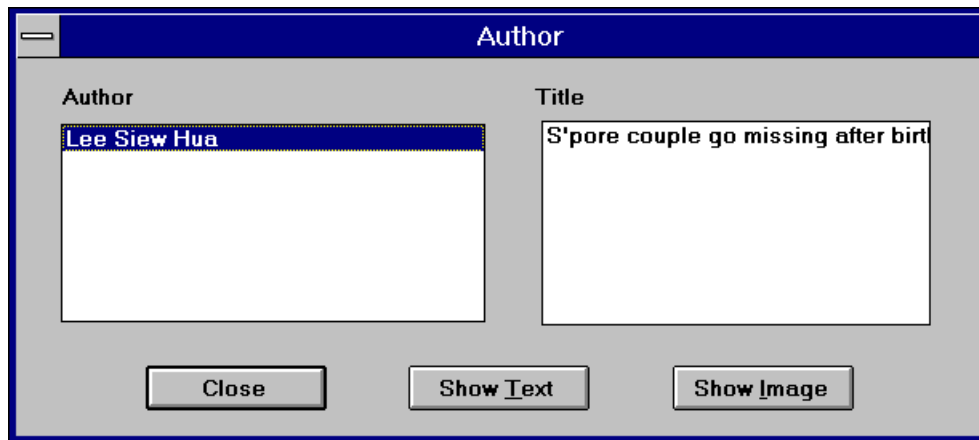
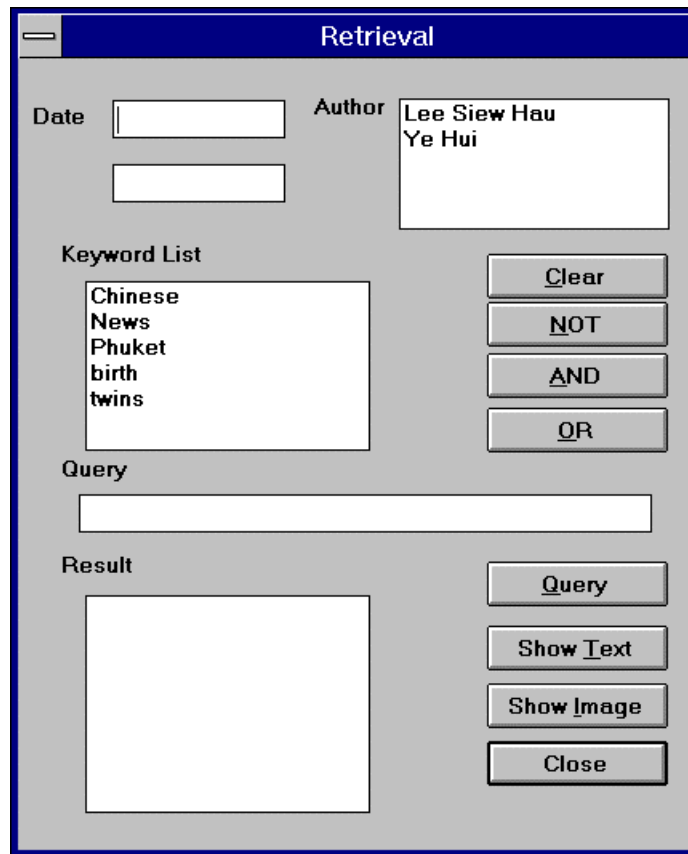


Figure 3.12 Author Retrieval Dialog

Text display is easier to handle than image display. The system only needs to read in the ASCII codes and displays them into a window. Figure 3.7 shows the interface of the text window.

To display an article image, one needs to know the graphics file format. TIFF file format and bitmap are used in this system. The Image Class (in Chapter 4) provides the image I/O and display functions. Figure 3.13 shows the image window.



The image shows a 'Retrieval' dialog box with a blue title bar. It contains several input fields and buttons. The 'Date' field has two empty text boxes. The 'Author' field contains the text 'Lee Siew Hau' and 'Ye Hui'. The 'Keyword List' field contains a list of keywords: 'Chinese', 'News', 'Phuket', 'birth', and 'twins'. To the right of the keyword list are four buttons: 'Clear', 'NOT', 'AND', and 'OR'. Below the keyword list is a 'Query' field with an empty text box. At the bottom left is a 'Result' field with an empty text box. To the right of the result field are four buttons: 'Query', 'Show Text', 'Show Image', and 'Close'.

Retrieval	
Date	<input type="text"/> <input type="text"/>
Author	Lee Siew Hau Ye Hui
Keyword List	<div>Chinese News Phuket birth twins</div> <div>Clear NOT AND OR</div>
Query	<input type="text"/>
Result	<div><input type="text"/></div> <div>Query Show Text Show Image Close</div>

Figure 3.13 Retrieval Dialog

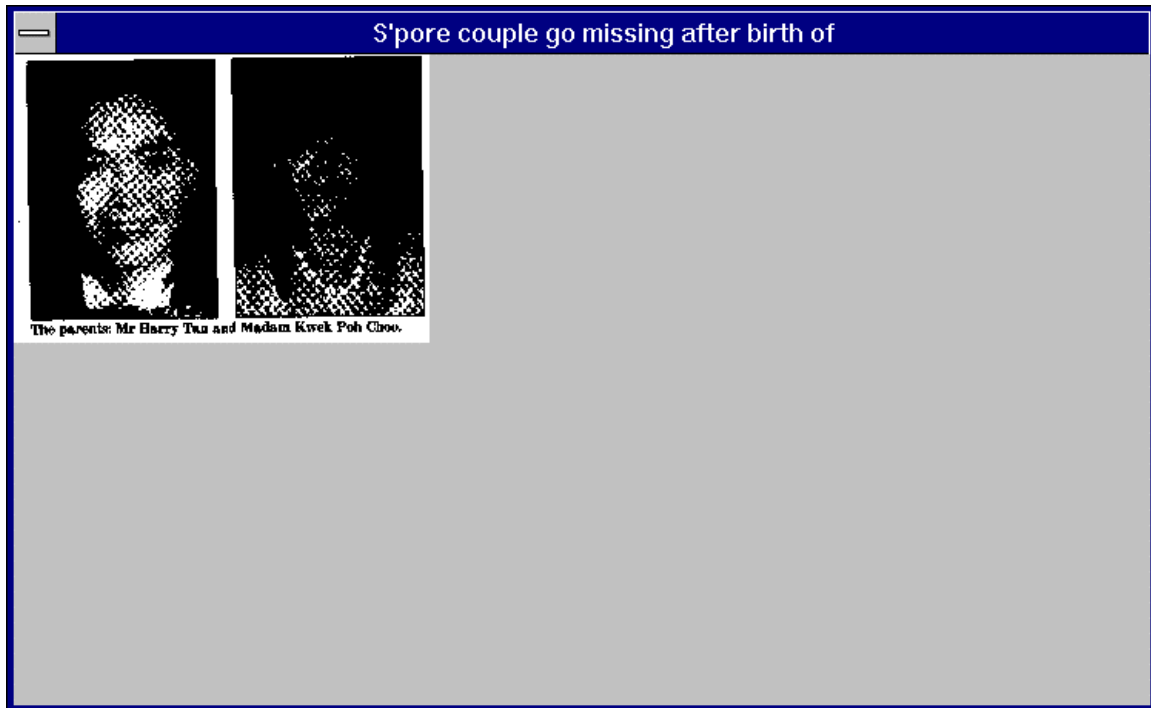


Figure 3.14 Image Window

4. Program Design and Implementation

Image Class provides image manipulation functions which include the graphics file I/O, image display and zoom functions.

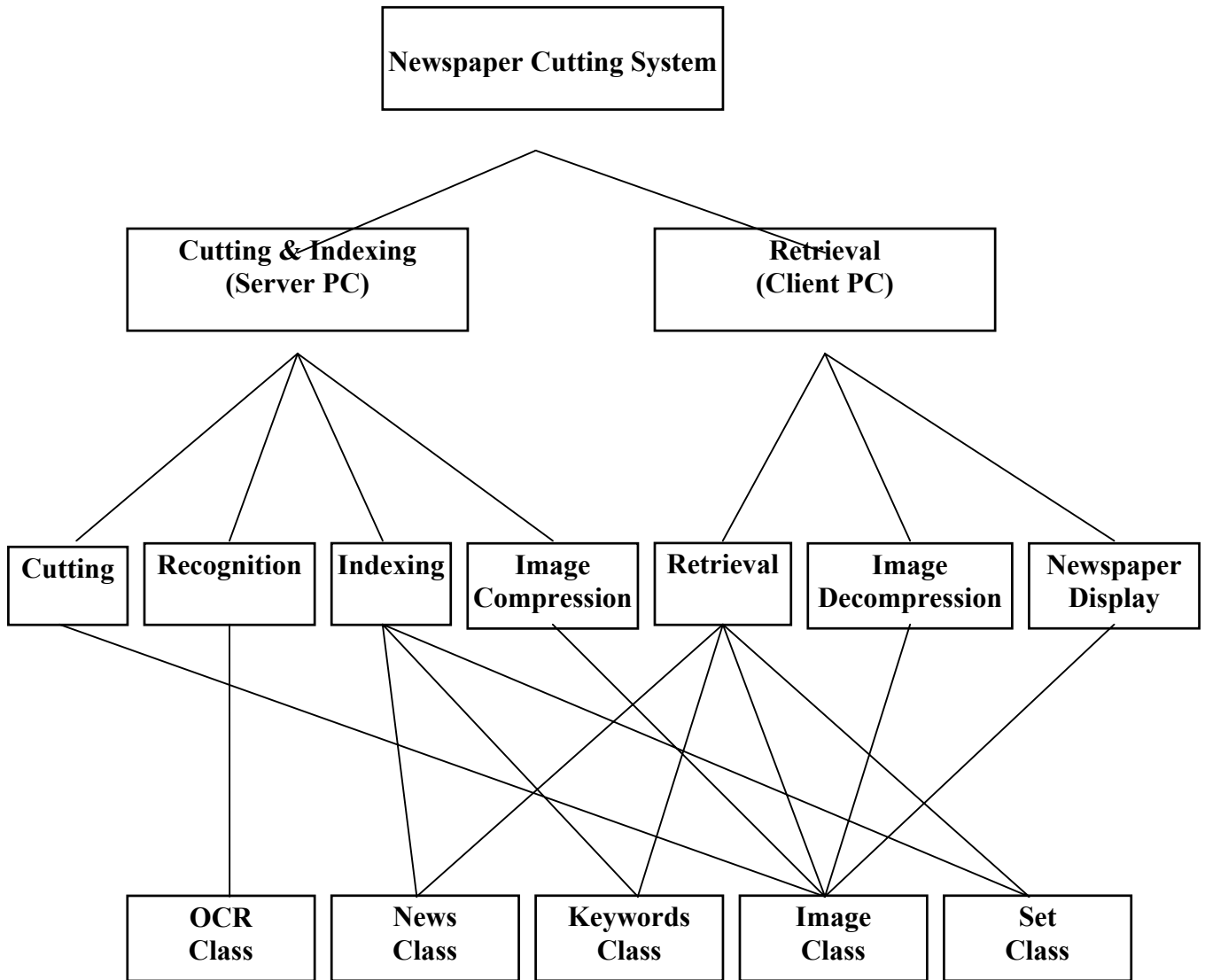


Figure 4.1 Relationships Among Functional Components and Classes

4.1.1 Data Member

This value is the handle of the in-memory bitmap. When an image object is constructed, this value is initialized to NULL. After reading graphics file from disk, m_hDIB contains the handle of the in-memory bitmap. Users can access this value by the member function GetHDIB(). When destroying an image object, the system will release the memory assigned to the object.

4.1.2 Member Functions

. . .

This member function returns the handle of the in-memory bitmap. The handle will be NULL if there is no graphics file read from the disk.

.

This member function provides the in-memory bitmap display. The first parameter is the HDC to display the image. The HDC can be either screen or printer, so that this function can also print the in-memory bitmap on a printer. The second and third parameters are the region of the output device and the in-memory bitmap. The two rectangles can be equal in size or not. If they are not equal, the function will zoom the image to fit the size of the output device.

. . .
. . .
. . .
. . .

These four functions get the attributes of the in-memory bitmap. The first function FindDIBBits() returns the address of the in-memory bitmap bits. Functions DIBWidth() and DIBHeight() get the bitmap width and height. And the last one, DIBNumColors() calculates the number of colors in the in-memory bitmap color table.

.
.
.

These three functions provide the graphics file I/O service. In this system, the scanned-in image is stored in Tag Image File Format (TIFF). The selected article image is stored in Device Independent Bitmap (DIB) file format. So two types of file I/O are presented in this class.

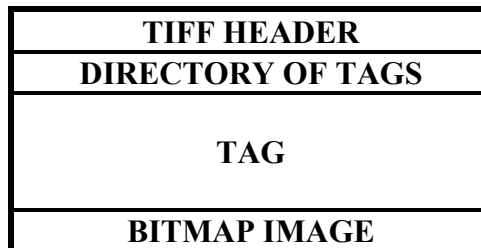


Figure 4.2 TIFF file format

There are four separate sections of a TIFF file as shown in the Figure 4.2. They include the header, the directory, tags and their associated data within the directory, and finally the bitmap image data.

Figure 4.3 shows the format of the DIB file. The DIB format consists of four sections. The first is the BITMAPFILEHEADER, which contains some useful information of the DIB file. Next is the BITMAPINFOHEADER, which contains the information of the bitmap. After the BITMAPINFOHEADER structure, a DIB will contain the color table. This is a set of RGBQUAD data structures holding the RGB color for each of the colors used in the bitmap. There will be as many RGBQUAD entries as there are color choices in the bitmap. The last section is the bitmap image data.

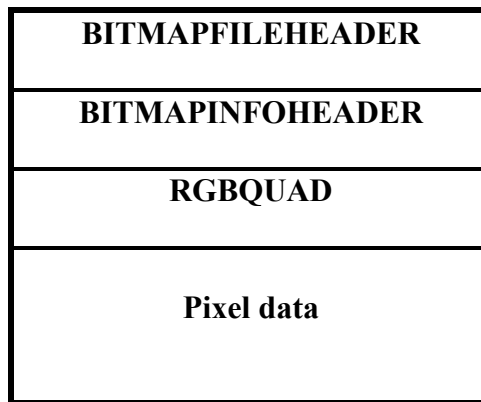


Figure 4.3 DIB file format

More details can be found in the Appendix A.

.
.

These two functions provide some cutting operations on the in-memory bitmap. The first one will return a handle of a bitmap which is a part of the source bitmap. The region is identified by the parameter rect. The Zoom() function also returns a handle of an in-memory bitmap. This bitmap size is smaller than the source bitmap. The parameter percentage gives the reduction ratio.

4.2 News Class

News Class provides a call-level interface that allows users to access the newspaper articles index. The following type definition shows the components of the News Class.

.

Member values `m_ID`, `m_TITLE`, `m_AUTHOR` and `m_DATE` provide the document ID, article title, author and date of each newspaper article. The member function `GetTitle()` returns the article title associated with the document ID. The member function `AddNews()` adds a record of newspaper article. Function `GetAllAuthor()` returns the author list of the system. The last two functions `GetDataByAuthor()` and `GetDataByDate()` get the article title list associated with the author or dates given.

Keyword Class provides a call-level interface that allows users to access the newspaper articles' keyword index. The following type definition shows the components of the Keyword Class.

29

```

~ ~      ~
.      ~ ~      .
.      .      ..      .

~ ~

.      .      . . .      .
.      .      . . .      ..      . .
.....
```

The member values `m_ID` and `m_KEYWORD` provide the document ID and its keywords. Member function `GetAllKeyword()` gets all the keywords stored in the database. The member function `AddKeyword()` adds some records of the document ID and its keywords.

4.4 OCR Class

OCR Class provides a call-level interface that allows applications to translate an image into text format. Calera Smart Host Libraries are used in this object. This set of libraries include the CRD Interface Library (CIL), Generic I/O Service Interface (IOS) Library and the Calera TIFF library [3].

CRD Interface Library is a set of routines that provide a high level interface to a CRD. The routines translate the configuration, control, zone, and processing functions into sets of CRD commands (sequences of characters), send them to the CRD using the host-independent I/O services, read the results, and convert them into useful data structures. The CIL also converts the data returned by the CRD, providing routines for moving documents to and from the CRD host memory and files.

IOS library is a set of routines, unique to each operating system, that translates generic I/O requests made by the CRD Interface Library into host operating system specific system call. The division between the CRD interface level and the I/O server level is to separate implementation of host-independent conversion functions from that of host-dependent I/O functions; this enables the CRD Interface Library to be reused from host to host. The IOS also includes communication protocol support not provided by lower levels.

Calera TIFF libraries, `F_JTIFF.SDR` and `F_JTIFF.WDR`, are input file drivers. They are parts of the Scanner Interface Library (SIL). SIL is a set of routines for

generically controlling scanners and devices that look like scanners. This routines can be used directly to scan images from scanners conforming to the SIL specification.

The following type definition shows the implementation of the OCR Class.

```
.....
```

4.4.1 Data Members

```
.
```

This value is the title of the recognized text. Each article has a title. Generally, the size of the title characters is bigger than the size of the other characters of the article. So in this system, we assume that the line with the largest font size is the title of an article. After recognition, the title is stored in the member value m_TITLE.

4.4.2 Member Functions

```
.
```

These two member functions provide the initialization and closing of the CRD recognition engine. The first one initializes the CRD recognition engine, assigns a TIFF file as the source image file. The last one shut down the CRD recognition engine.

Member function InitOCR() uses the function c_initialize() of CIL. Since it is running under Windows environment, and using the TIFF file as the source image file, the driver name is

```
rsp:winocrvi,1,0+F_JTIFF.image.tif
```

Member function CloseOCR() uses the function c_done() of CIL to shut down the recognition engine. More details can be found in the Appendix B.

.

This member function translates the image file into text format. As discussed in the cutting process, a series of rectangular boxes are used to define a newspaper article. In this function, these rectangular boxes are considered as a list of zones, added to the recognition zone bank by using the CIL function c_a_pzone(). After all the rectangular boxes are added into the zone bank, the recognition engine will start by calling the CIL function c_readnext(). Figure 4.4 shows the process of the member function Recognition().

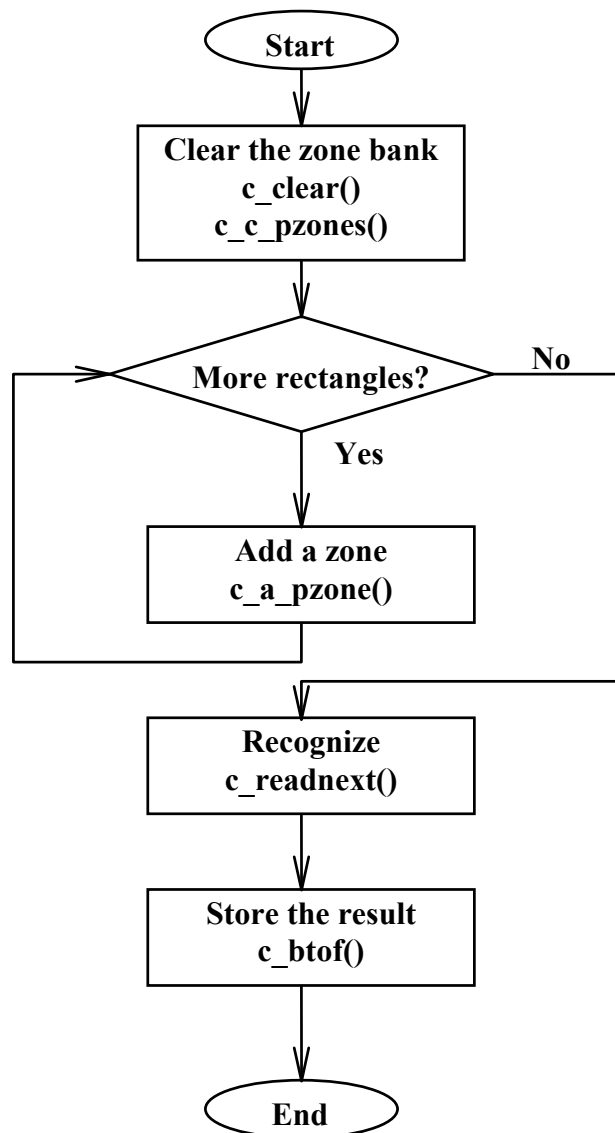


Figure 4.4 Process of the Function Recognition()

.

The recognition engine recognizes image into text, and uses the `c_btof()` command to take these results from the box and store them in a PDA document file. PDA file is a ASCII file. Unlike other text files, PDA file contain some attributes, which are encoded in the form of escape sequences. These sequences start with the character 0x9b or 0x1b and are followed by numbers separated by semicolons, finally ending with a single character end-code.

This member function will read in a .PDA file and convert it to a .TXT file. While converting the file, the title is detected and stored in the member value `m_TITLE`.

4.5 Set Class

A set is a homogeneous, unordered collection of elements. In programming language terms, a set has an associated “ element data type ”, and all elements of the set must be of this type. Each element data type has a key. For a given set, no two elements ever simultaneously possesses the same key. An element data type may also specify other data that will be included along with the element. These data values need not be unique among all elements of a set.

In this system, the Set Class is derived from the Class CObList. The CObList Class supports ordered lists of nonunique CObject pointers accessible sequentially or by pointer value. CObList lists behave like doubly-linked lists. A variable of type POSITION is a key for the list. Since a set is an unordered collection of elements, a check must be done before each insertion to determine whether the element is already in the set.

The following type definition shows the implementation of the Set Class.

```
. . . . .  
  
.  
.  
.....
```

● ● ● ● ● ●

. . . .
. . . .

The member function CopyTo() makes a duplicate of the source set. The member function IsMember() tests whether the element is in the set.

4.6 Database Access

The two database files provide a call-level interface that allows applications to access data stored in the Foxpro database [2]. They are implemented by using the ODBC driver. ODBC is the database portion of the Microsoft Windows Open Services Architecture (WOSA), an interface which allows Windows-based desktop applications to connect to multiple computing environments without re-writing the application for each platform.

An application achieves independence from DBMSs by working through an ODBC driver written specifically for a DBMS rather than working directly with the DBMS. The driver translates the calls into commands its DBMS can use, simplifying the developer抯 work, and making it available for a wide range of data sources. The database classes support any data source for which you have an ODBC driver.

The ODBC architecture has four components:

- Application: Performs processing and calls ODBC functions to submit SQL statements and retrieve results.
- Driver Manager: Loads drivers on behalf of an application.
- Driver: Processes ODBC function calls, submits SQL requests to a specific data source, and returns results to the application. If necessary, the driver modifies an application's request so that the request conforms to syntax supported by the associated DBMS.
- Data source: Consists of the data the user wants to access and its associated operating system, DBMS, and network platform (if any) used to access the DBMS.

The Driver Manager and driver appear to an application as one unit that processes ODBC function calls. Figure 4.5 shows the components of ODBC.

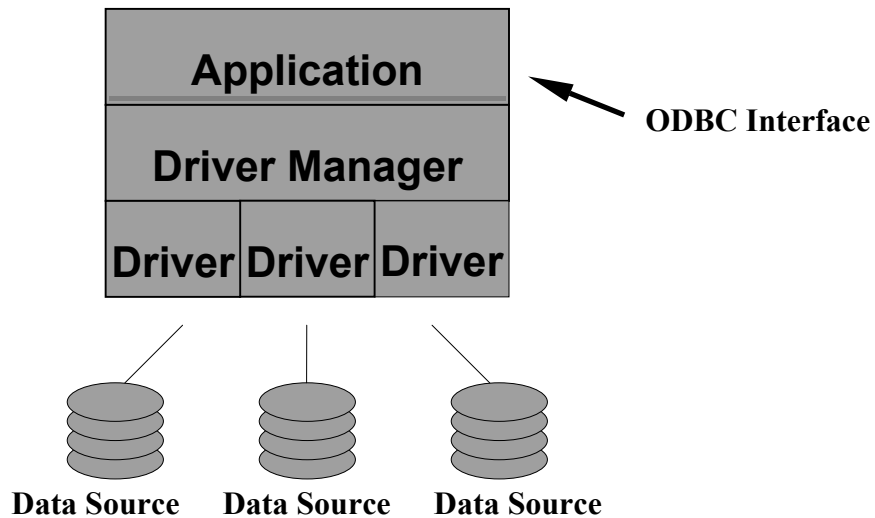


Figure 4.5 Components of ODBC

The two database files created in the Foxpro DBMS are the NEWS.DBF and KEYWORD.DBF. Tables 4.1 and 4.2 show the structures of the database files.

Structure
Keyword
Document ID

Table 4.1 Structure of KEYWORD.DBF

Structure
Document ID
Author
Date
Title

Table 4.2 Structure of NEWS.DBF

CHAPTER 5

5. Performance Analysis

5.1 Compression Efficiency

Table 5.1 shows the compression ratios of three newspaper article images. Each newspaper article image is identified by a document ID.

Newspaper Article Image	Original Size (Bytes)	Compressed Size (Bytes)	Compression Ratio
1	266298	51108	80.8%
2	36862	10910	70.4%
3	22862	5756	74.8%

Table 5.1 Compression Ratios

These bilevel images are compressed under the Huffman algorithm. The compression technique produces a compression ratio of about 75%.

5.2 Storage Capacity

About 75% of storage can be saved after compression. Assume that each newspaper article image size is about 100 Kbytes. If the library needs to store 100000 newspaper article images, it only needs a storage capacity of 200 Mbytes to hold the compressed newspaper article images. Keeping the uncompressed images will need the library to purchase more than 1000 Mbytes of store space. Therefore, by compressing the images, it not only reduces the storage capacity required, but also cuts the cost in purchasing the storage media.

CHAPTER 6

6. Recommendation And Future Expansion Areas

6.1 Newspaper Cutting

The Newspaper cutting interface can be further improved to define other shapes (e.g. T shape, L shape, etc.) directly. Librarians can use a mouse to define any shape of area that indicates a newspaper article to be cut. In addition, the system should be able to analyze the layout of the newspaper article automatically. In the current system, only one photograph or picture image can be defined for a newspaper article. The system can be extended by allowing one article to have more than one images. The feature is very useful in some financial news articles, in which there are many tables and charts which will be captured as images.

6.2 Automatic Identification of Newspaper Articles

Currently, recognition can only be processed on the English newspapers. The system can be extended to recognize non-English newspapers. In addition, by improving the recognition engine, more attributes of the newspaper articles (e.g. font, size, line spacing, etc.) can be handled. Therefore, the system can analyze the newspaper layout automatically — finding text, titles and photos, and recognize the whole page of the newspaper automatically.

Another area to be investigated in the future is to verify the recognized text and correct the misspelled words automatically.

6.3 Indexing

First thing to improve on indexing is to design another data structure to store the index file. Currently, it uses the Foxpro database file to store the index. Other data

structures (e.g. B-tree, B⁺-tree, etc.) can be investigated to reduce the storage capacity and improve the performance of the retrieval.

Another area which needs to be improved is automatic indexing. Currently, the system only gets a single word as keyword. It can be further improved by using two words term or three words term as the keyword while processing automatic indexing. This method is also useful in indexing non-English articles.

Non-English (Chinese) articles are indexed manually in the current system. It uses English words or terms as keywords. The system may give the relationship between English terms and Chinese terms so that automatic indexing can also be possible with Chinese articles.

6.4 Image Compression

The Huffman algorithm is used in this system to compress newspaper article images. Other compression algorithms can be investigated to produce a better compression ratio. Huffman algorithm is a lossless compression technique. Other lossy compression techniques can be studied for the system.

6.5 Newspaper Article Retrieval

The current system only retrieves the articles and shows the text and images separately. The system can be extended to allow text and images to be reconstructed, and display onto the screen in the same layout as they are on the newspaper.

A short form table can be defined to improve the performance of the retrieval. For example, if we retrieve articles which contain the keyword 摺ingapore”, articles which have the keyword 摺捺ore” should also be retrieved.

CHAPTER 7

7. Conclusion

The Newspaper Cutting System has been designed and developed so that it can be used in libraries to keep and cut newspapers in an electronic way. By using this system, newspaper articles are selected, cut, indexed and filed automatically. The concept of the “paperless library” developed and information can be processed via computers.

Object Oriented Programming technology is used. Five classes are defined. They are Image Class, News Class, Keyword Class, OCR Class and Set Class. These Classes not only can be used in this system, but also can be used in other applications.

Recognition engine is used in this system. English characters can be recognized from image format to text format. Chinese articles and other non-English articles are stored as compressed images. An user friendly indexing and retrieval interface is provided in this system.

The Newspaper Cutting System is a very useful application in libraries. Automation of newspaper cutting process will definitely be found in most libraries in the near future.

References

1. 拑icrosoft Visual C++ Reference Book”, Microsoft Corporation, 1993.
2. 拑icrosoft Foxpro Reference Book”, Microsoft Corporation, 1993.
3. 拑alera Smart Host Libraries Reference”.
4. 拑DBC API Programmer拑 Reference”, Microsoft Corporation, 1993.
5. 拑isual C++ object-oriented programming”, Andrews, Mark, Carmel, Ind.: SAMS Pub., 1993.
6. 拑bject-oriented programming: analysis, design and implementation method”, John Caspers, Charleston, S.C.: Computer Technology Research Corp., 1994.