A KNOWLEDGE-BASED APPROACH TO DERIVING LOGICAL STRUCTURE FROM DOCUMENT IMAGES

by

Debashish Niyogi

A dissertation
submitted to the Faculty of the Graduate School
of State University of New York at Buffalo
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

August 1994

©Copyright by Debashish Niyogi, 1994.
All rights reserved.

To my parents,
for the values they have instilled in me,
their faith in my abilities,
and their constant encouragement and support
in my educational endeavors.

Dissertation Committee

Chairman:

Dr. Sargur N. Srihari
Dept. of Computer Science, SUNY at Buffalo

Members:

Dr. William J. Rapaport
Dept. of Computer Science, SUNY at Buffalo

Dr. Raj S. Acharya
Dept. of Electrical & Computer Engineering, SUNY at Buffalo

External Reader:

Dr. Richard G. Casey
IBM Almaden Research Center

This dissertation was defended on August 3, 1994.

Acknowledgements

I would first and foremost like to acknowledge a tremendous debt of gratitude to my advisor Dr. Sargur N. Srihari for providing me with the opportunity to conduct this research, for his constant encouragement and guidance in my academic efforts, and for his enormous patience and belief in my abilities throughout my years as a graduate student.

I am very grateful to my committee member Dr. William J. Rapaport who scrutinized my dissertation in great detail and gave me invaluable feedback that enabled me to improve its content and presentation, and who, as the Director of Graduate Studies as well as my French language examiner, helped me in many ways to finish my graduate program.

I will forever be grateful to Mrs. Eloise Benzel of the Computer Science Department, whose never-ending help and advice made my life as a graduate student so much more enjoyable, and whose cheerfulness and words of encouragement were instrumental in helping me to maintain my dedication and morale over the years.

I would like to thank Dr. Dayanand S. Rajan, whom I have been proud to call a true friend for many years, for his advice and help, both personal as well as professional, through the years. Without his constant encouragement, and his example as an excellent researcher, this dissertation may never have been completed.

I would like to offer my thanks to Gloria Koontz and Sally Elder of the Computer Science Department, who have always been eager to help in solving any problems that arose during my graduate career, and who helped to keep my spirits up during the difficult stages.

It was an honor to have Dr. Richard Casey of IBM Almaden Research Center as my external reader, and I thank him for reading my dissertation in great detail and giving me valuable suggestions to help me improve the clarity of its presentation.

I am grateful to Dr. Shoshana Hardt for being a mentor during the early stages of my graduate career, and for giving me some very profound advice that convinced me to pursue a doctoral degree in the first place.

I would like to thank Dr. David Sher, who once served on my committee, for providing me with insightful feedback on my research as well as advice on sound research techniques.

Thanks to my committee member Dr. Raj Acharya for taking on the task at such short notice and for his flexibility in adjusting to my graduation schedule.

Finally, I am grateful to the many CEDAR staff members and students who made the last couple of years more productive by always being willing to help in solving technical problems, and especially to Heike Berlind, Natalie Weaver and Eugenia Smith of CEDAR who were always helpful in dealing with various administrative matters.

Abstract

An important application of artificial intelligence is knowledge-based document image understanding, specifically the analysis of document images to identify and logically classify objects in an image. The object classification problem is particularly interesting in the domain of document images, where the problem becomes that of segmenting the different regions, or blocks, of printed matter using standard image processing techniques, and then using spatial domain knowledge to first classify these blocks (e.g., text paragraphs, photographs, etc.), then group these blocks into logical units (e.g., newspaper stories, magazine articles, etc.), and finally determine the reading order of the text blocks within each logical unit. This essentially translates to the problem of converting the physical structure of the document into its logical structure with the use of domain knowledge about document layout. The objective of this work is to develop a computational model for the derivation of the logical structure of document images using certain formalisms designed for this task. In this approach, a simple yet powerful rule-based control strategy utilizes the data obtained from the invocation of different types of operations on a digitized document image, and makes inferences about the image using a knowledge base of document layout rules. In this model, the domain knowledge about document structure is represented in a unified multi-level hierarchical form, and inferences are made by reasoning processes which use that knowledge. The main issues investigated in this research are: the kind of domain and control knowledge that is required, how to represent this knowledge in a globally integrated form, and how to devise a control strategy that efficiently utilizes this integrated knowledge base. A knowledge-based document logical structure derivation system (DeLoS) has been developed based on this approach. The system consists of a hierarchical rule-based reasoning system that controls the block classification, block grouping and read-ordering operations; a global data structure that stores the physical and logical structures; and layout analysis routines that facilitate the transfer of information to this system from a set of image processing modules that are utilized to provide input data about the image. The document images that this system deals with are primarily digitized newspaper pages. Applications of this approach include use in comprehensive document understanding systems that can read document textual information and allow interactive querying of the semantic information in a document, thus leading to a conceptual understanding of the contents of a document.

Contents

1 Introduction **1**

1.1 Statement of Work 4

1.2 Motivation 14

1.3 Research Objectives 16

1.4 Organization of the Dissertation 19

2 Background **20**

2.1 Document Image Analysis 20

2.2 Knowledge-Based Image Analysis 23

2.3 Document Layout Analysis 26

2.4 Document Image Understanding 28

3 A Model for Document Logical Structure Derivation **32**

3.1 Analysis of Document Structure 32

3.1.1 Newspaper formats 36

3.2 Development of a Computational Model 39

3.2.1 Framework Used 39

3.2.2 Design Methodology 41

3.3 Control Issues in Logical Structure Derivation 45

3.3.1 Block Classification 47

3.3.2 Block Grouping 48

3.3.3 Determination of Read-Order for Text Blocks 50

4 DeLoS: A System For Deriving Logical Structure 52

4.1 Development of a Logical Structure Derivation System 52

4.2 System Organization 53

4.3 Rule-Based Control Structure 55

4.3.1 The Knowledge Base 56

4.3.2 The Inference Engine 57

4.4 The Global Data Structure 59

4.4.1 Representation of Domain Data 61

5 Development of the Knowledge Base in DeLoS 64

5.1 Domain Knowledge 65

5.2 Document World Knowledge 66

6 Implementation of the DeLoS System **71**

6.1 System Implementation 71

6.2 System Inputs 74

6.3 System Outputs 75

6.4 Image Processing 78

7 Experimental Results and Performance Analysis for DeLoS **82**

7.1 Experimental Results 82

7.2 Analysis of System Performance 84

8 Summary and Future Directions **97**

8.1 Summary 97

8.2 Future Work 100

A DeLoS Domain Knowledge **103**

List of Figures

1.1 Examples of different types of documents. 3

1.2 Logical structure of a document. 5

1.3 Example of document layout analysis. 8

1.4 Logical groupings of blocks in the document image. 9

1.5 Logical description derived from document image. 10

1.6 Logical structure tree for example image. 12

1.7 Reconstruction of physical structure of example image. 13

1.8 Different document states during document understanding. 18

3.1 Syntax of physical and logical structures of a typical newspaper. 34

3.2 Examples of newspaper layout rules. 38

3.3 Computational model for logical structure derivation. 40

3.4 Control strategy for block classification. 47

3.5 Control strategy for logical block grouping. 49

3.6 Control strategy for read-order determination. 50

3.7 Example of read-order of text blocks within a unit. 51

4.1 The DeLoS logical structure derivation system. 54

4.2 Examples of knowledge rules in the rule-based system. 57

4.3 Examples of control rules in the rule-based system. 59

4.4 Examples of strategy rules in the rule-based system. 60

4.5 Slots in block and unit frames. 62

5.1 Rules using domain knowledge. 67

5.2 Rules using document world knowledge. 69

6.1 Acceptable block types and block labels for the DeLoS system. 74

6.2 Sample input for DeLoS. 76

6.3 Sample output from DeLoS. 77

6.4 Image processing steps for documents. 79

7.1 Results of block segmentation on a page from *The Buffalo News*. 85

7.2 Output of DeLoS for the page from *The Buffalo News*. 86

7.3 Reconstructed physical structure for the page from *The Buffalo News*. . . . 87

7.4 Results of block segmentation on a *USA Today* page. 88

7.5 Output of DeLoS for the *USA Today* page. 89

7.6 Reconstructed physical structure for the *USA Today* page. 90

List of Tables

7.1 Results of DeLoS on pages of *USA Today*. 91

7.2 Results of DeLoS on pages of *The Buffalo News*. 92

7.3 Performance of DeLoS on pages of *USA Today*. 95

7.4 Performance of DeLoS on pages of *The Buffalo News*. 96

Chapter 1

Introduction

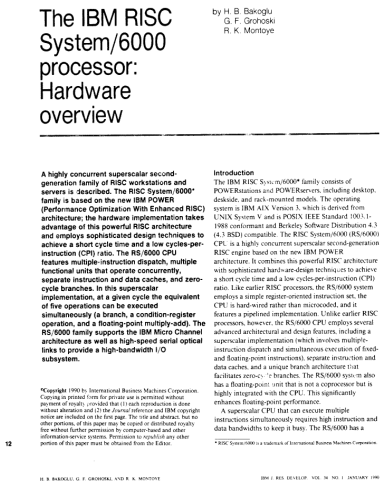
A rapidly growing field of research in artificial intelligence is the application area of document image understanding, where current AI techniques are applied to the analysis of document images. A *document image* is a visual representation of a printed page such as a journal article page, a magazine cover, a newspaper page, etc. Typically a document consists of blocks of text, i.e., letters, words, and sentences, that are interspersed with halftone pictures, line drawings, and symbolic icons. A document image is therefore a digital two-dimensional array representation of a document obtained by optically scanning and raster digitizing a hard copy document. Document image *analysis* is the task of recognizing objects in an image by using techniques that extract homogeneous regions within the image. Document image *understanding* is the goal-oriented task of deriving a symbolic representation of the contents of a document image, which involves detecting and interpreting different blocks (like photographs, text, line drawings, etc.), accounting for the interactions of the different components, and coordinating the interpretations to achieve an end result.

Document images present a more structured domain than general images. First, a document image is a 2-D representation of a 2-D scene, rather than a 2-D representation of a 3-D scene. Consequently, factors such as perceptions of depth, occluded objects, and

shadows are absent in a document image (except for photographs contained within the document), and do not have to be accounted for in the analysis of the image. Second, documents generally conform to a certain geometric structure that dictates that the document be composed of a set of interconnecting rectangular printed regions, or blocks. The layout of the different rectangular blocks with respect to each other, i.e., the spatial relationships between these blocks, varies among different types of documents and is dependent on the layout conventions used for specific types of documents. For example, different newspapers, or even different pages of the same newspaper, can have significantly different layouts of printed blocks. Figure 1.1 shows some examples of different types of documents.

However, there is an underlying structure for all printed documents that is governed by certain basic constraints: First, the physical blocks into which all printed documents can be spatially divided represent meaningful physical divisions of the document. Second, each of the physical blocks of printed matter can be classified according to certain basic categories like "text", "photograph", "line-drawing", etc. Third, these physical blocks can be logically grouped to make up units that represent meaningful logical entities in a document, e.g., a newspaper story, magazine article, etc. Fourth, there exists a specific order in which the text blocks within each unit must be read in order for the information in the block to make sense syntactically and semantically. Thus, spatial knowledge is a very important factor in the task of classifying, grouping, and ordering object blocks in a document image, which is one of the primary objectives of document image understanding.

The first two of the above constraints can be met by performing appropriate tasks, i.e., the segmentation and identification of the different blocks in an image. This refers to the extraction of the physical structure of the document from the image, and is known as *document layout analysis*. All documents have a physical structure that can be derived and



The IBM RISC System/6000 processor: Hardware overview

by H. B. Bakoglu
G. F. Grohoski
R. K. Montoye

A highly concurrent superscalar second-generation family of RISC workstations and servers is described. The RISC System/6000* family is based on the new IBM POWER (Performance Optimization With Enhanced RISC) architecture; the hardware implementation takes advantage of this powerful RISC architecture and employs sophisticated design techniques to achieve a short cycle time and a low cycles-per-instruction (CPI) ratio. The RS/6000 CPU features multiple-instruction dispatch, multiple functional units that operate concurrently, separate instruction and data caches, and zero-cycle branches. In this superscalar implementation, at a given cycle the equivalent of five operations can be executed simultaneously (a branch, a condition-register operation, and a floating-point multiply-add). The RS/6000 family supports the IBM Micro Channel architecture as well as high-speed serial optical links to provide a high-bandwidth I/O subsystem.

Introduction

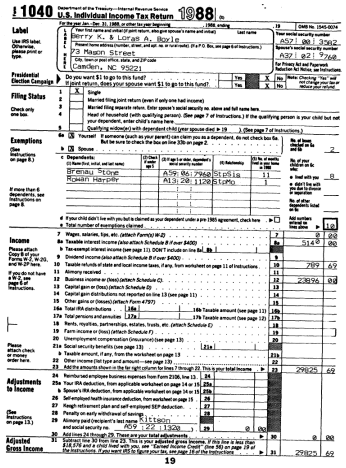
The IBM RISC System/6000* family consists of POWERstations and POWERservers, including desktop, deskside, and rack-mounted models. The operating system is IBM AIX Version 3, which is derived from UNIX System V and is POSIX IEEE Standard 1003.1-1988 conformant and Berkeley Software Distribution 4.3 (4.3 BSD) compatible. The RISC System/6000 (RS/6000) CPU is a highly concurrent superscalar second-generation RISC engine based on the new IBM POWER architecture. It combines this powerful RISC architecture with sophisticated hardware-design techniques to achieve a short cycle time and a low cycles-per-instruction (CPI) ratio. Like earlier RISC processors, the RS/6000 system employs a simple register-oriented instruction set, the CPU is hard-wired rather than microcoded, and it features a pipelined implementation. Unlike earlier RISC processors, however, the RS/6000 CPU employs several advanced architectural and design features, including a superscalar implementation (which involves multiple-instruction dispatch and simultaneous execution of fixed- and floating-point instructions), separate instruction and data caches, and a unique branch architecture that facilitates zero-cycle branches. The RS/6000 system also has a floating-point unit that is not a coprocessor but is highly integrated with the CPU. This significantly enhances floating-point performance.

A superscalar CPU that can execute multiple instructions simultaneously requires high instruction and data bandwidths to keep it busy. The RS/6000 has a

*Copyright 1990 by International Business Machines Corporation. Copying in printed form for private use is permitted without payment of royalty, provided that (1) each reproduction is done without alteration and (2) the *Journal* reference and IBM copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free without further permission by computer-based and other information-service systems. Permission to republish any other portion of this paper must be obtained from the Editor.

* RISC System/6000 is a trademark of International Business Machines Corporation.

(a) A journal page



1040 U.S. Individual Income Tax Return 1988

(b) A form

TIME

MAY 11, 1992 THE WEEKLY NEWSMAGAZINE VOLUME 139, NO. 19

THE WEEK 10

NATION: Throw the bankers out

WORLD: Unexpected resignation and widespread strikes rattle Germany

BUSINESS: Welcome to the IMF family

HEALTH & SCIENCE: A hot spot in the sea could mean global warming is finally here

COVER STORIES: A Shocked Nation Wonders What Went Wrong and How to Mend It

18 THE OVERVIEW: The Fire This Time
Why race still divides America and its people

26 VIOLENCE: Los Angeles in a Fury
Waiting for a match to ignite its frustration

30 THE TRIAL: The Tale of the Tape
How the jury managed to acquit the accused officers

37 INTERVIEW: New Top Cop Takes Command
Willie Williams hopes to rebuild bridges in Los Angeles

38 RACE: The Limits of Black Power
Why electing African Americans has not calmed ghetto rage

41 THE POLITICAL INTEREST: What Bush Can Do
A chance to make up for Willie Horton

42 MEMOIRS: Mikhail Gorbachev's Story
The fallen leader reflects on how he lost his job

48 THE BALKANS: Beyond Comprehension
Why Yugoslavia's ethnic bloodletting has no quick fix

51 ECONOMY: GATT Matters (Really)
If trade talks fail, Americans will notice

53 ENVIRONMENT: Odd Serenity in a Nuclear Town
Don't tell the people of Oak Ridge they might get cancer

56 LIVING: A Good $5 Cigar
Smoking may be out, but premium stogies are in style

57 THEATER: The Quirky Voice of Falsettos
Composer William Finn brings rich emotions to Broadway

18 Shotgun at the ready, an L.A. policeman enforces city curfew

REVIEWS

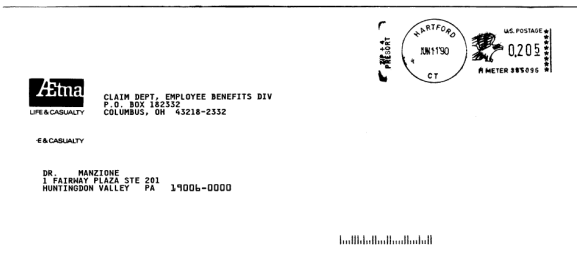
TELEVISION Why all the fuss over *Murphy Brown*? 59
THEATER A British satire loses its zip on Broadway 60
CINEMA *Midnight Cheer* is a compelling wartime anecdote 60
ART An overblown show of a popular American eye-fooler 61
BOOKS William Greider on the ailing political system 63

Cover: Photograph for Time by John T. Barr—Gamma Liaison

DEPARTMENTS

MANAGING EDITOR'S LETTER 4
LETTERS 6
GRAPEVINE 9
MILESTONES 17
SHORT TAKES 65
PEOPLE 67
ESSAY 68

(c) A magazine page



Aetna
LIFE & CASUALTY

CLAIM DEPT, EMPLOYEE BENEFITS DIV
P.O. BOX 182332
COLUMBUS, OH 43218-2332

E & CASUALTY

DR. MANZIONE
1 FAIRWAY PLAZA STE 201
HUNTINGDON VALLEY PA 19006-0000

HARTFORD CT

U.S. POSTAGE 0.205

(d) A mailpiece

Figure 1.1: Examples of different types of documents.

represented in a hierarchical form. Document layout analysis is the first step towards the larger goal of document image understanding.

The third and fourth constraints mentioned above can be met by performing appropriate tasks, i.e., classification and logical grouping of blocks. This refers to the extraction of the logical structure of the document, which we will call *logical structure derivation*. This is the next major step towards document image understanding.

Document layout analysis and logical structure derivation enable us to determine the relationship between the *physical* layout of a document page (consisting of the geometric structure and spatial relationships of the different blocks of printed matter) and its *logical* layout (consisting of the logical groupings of related blocks into composite units). This can then be used to actually "translate" a physical document into its logical symbolic representation, so that the document can be "read" and "understood" by a computer in a manner similar to human reading and comprehension.

The logical structure of a document is independent of the physical layout of the printed document, and is defined in terms of the logical entities into which a document is divided. Figure 1.2 shows what the graphical representation of the logical structure for a typical document might look like.

1.1 Statement of Work

The transformation of the physical structure of a document into the logical structure is a critical component of document image understanding, as mentioned above. Currently this transformation process is not adequately defined, especially for complex multi-articled documents (such as newspapers). Therefore, there exists a need for the development of

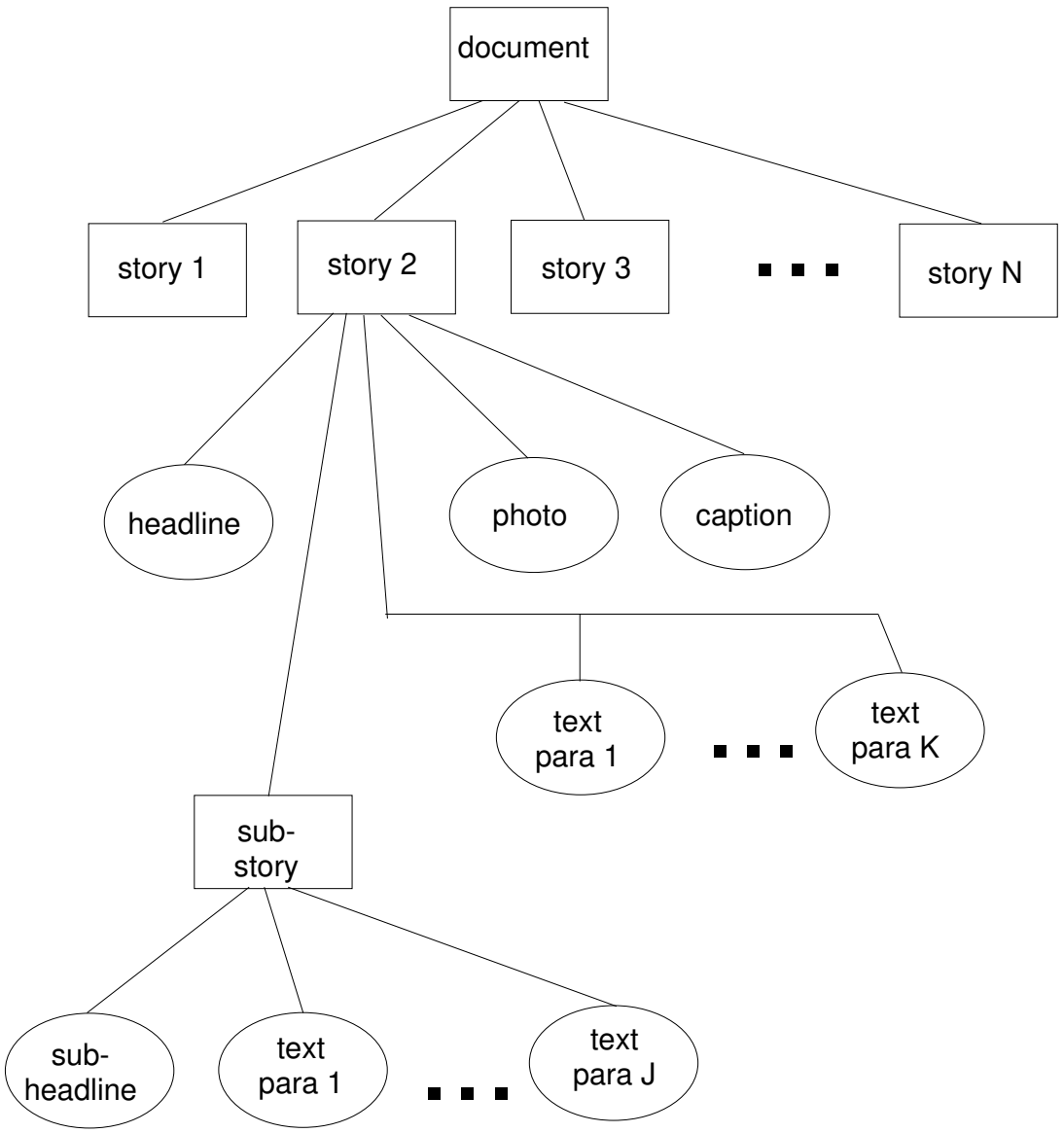


Figure 1.2: Logical structure of a document.

a computational model for the transformation of a *physical* representation of a document (e.g., a digitized document image) into a *logical* representation of the same document (e.g., an editable file containing a complete symbolic description of the document, as explained below). There is also a need for a system design methodology that enables efficient knowledge organization and control of this transformation process, leading to the design of a system that derives the logical structure of a document page from the digitized image of that page.

The steps involved in the transformation of a scanned image of a physical document into its logical description are:

1. Physical segmentation of the image into its constituent printed "blocks".
2. Categorization of each of these blocks into block types or categories.
3. Labeling of these blocks, using domain knowledge, according to their specific identities.
4. Logical grouping of these blocks, using further domain knowledge, into logical "units".
5. Determining the reading order of text blocks within each unit, also using domain knowledge.
6. Translation of the block and unit information into an editable symbolic description of the logical structure of the document.

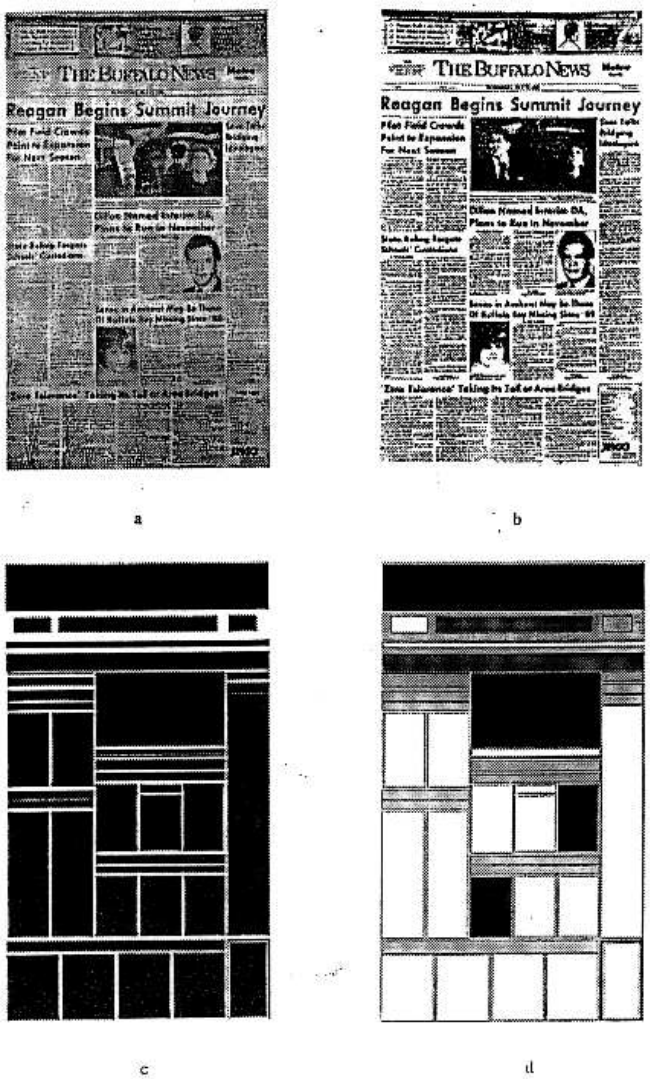
The above steps can be illustrated by showing an example of the derivation of logical structure from the physical structure of a typical document (in this case, the front page of an edition of *The Buffalo News*). Figure 1.3 demonstrates the layout analysis of a document image (i.e., steps 1 and 2 from the above list). Figure 1.3 (a) shows the original,

digitized, gray-level document image. Figure 1.3 (b) shows the binary image obtained after performing a thresholding operation on the original image. Figure 1.3 (c) shows the result after segmenting the image into its various printed blocks (seen in this image as black rectangles). Figure 1.3 (d) shows the result after inferring the basic categories of the blocks. The different categories are shown in different shades (white — "small-text", black — "half-tone", light gray — "medium-text", dark gray — "large-text").

In the logical structure derivation process (items 3–6 in the above list), the blocks shown in Figure 1.3 (d) are then classified into various domain-defined block labels, grouped into several logical units by following rules that are based on current layout conventions for that type of document, and read-ordered for text reading. Figure 1.4 shows the grouping of the blocks into logical units. The thick lines define the logical units, and each unit contains one or more of the labeled blocks. (The logical units represent contextually related blocks, such as those that make up a newspaper story. In this example all the blocks that belong in a unit are physically adjacent, but in the general case that is not necessarily true.)

Figure 1.5 is a symbolic description of the document showing the logical layout of its different units and blocks. Each line in this description contains detailed information on the location and characteristics of the block or unit it represents. The details given about each block are: the block number, the dimensions of the block, the coordinates of the upper-left corner of the block, the block type, the label of the block, the previous and next text block in logical reading order, and the number of the unit to which the block belongs. (A list of possible labels for a block is given in Chapter 6). The details given about each unit are: the unit number, the number of blocks in the unit,, and the block number of each block that is contained in the unit.

Figure 1.6 shows the hierarchical form of the logical structure of the example document



(a) Original digitized gray-level document image
(b) Binarized image after thresholding
(c) Segmented image with constituent printed blocks
(d) Image with blocks categorized into different types

Figure 1.3: Example of document layout analysis.

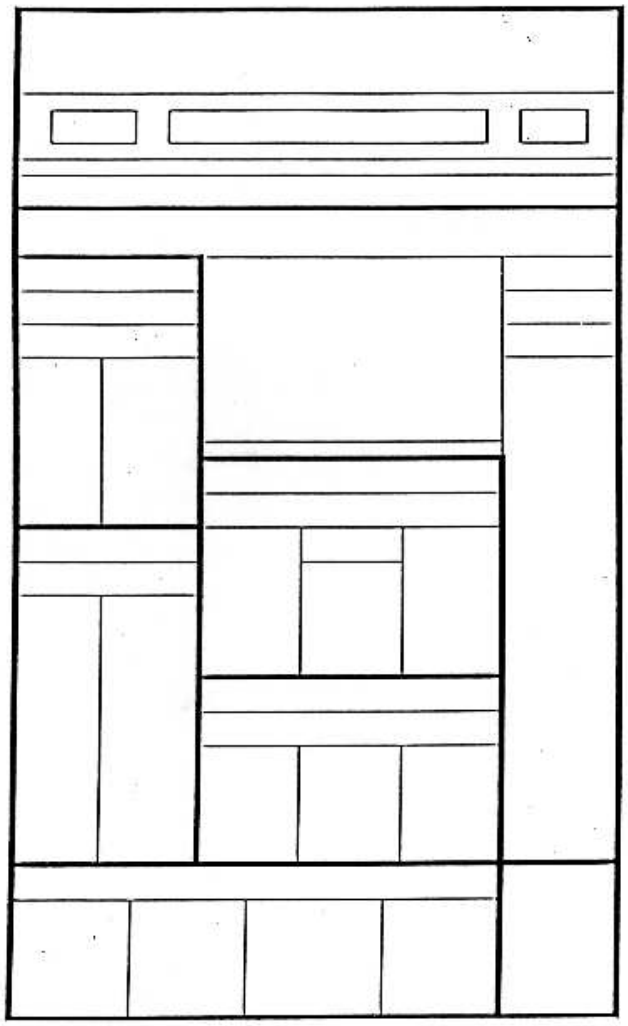


Figure 1.4: Logical groupings of blocks in the document image.

```
block(0,[dim(183,85),pos(42,273),type(1),blk_lbl(11)]).
block(0,[prevtxt(99),nexttxt(30),unit_lbl(7)]).
block(1,[dim(787,85),pos(264,273),type(3),blk_lbl(31)]).
block(1,[prevtxt(-1),nexttxt(-1),unit_lbl(7)]).
block(2,[dim(138,85),pos(1111,273),type(2),blk_lbl(31)]).
block(2,[prevtxt(-1),nexttxt(-1),unit_lbl(7)]).
block(3,[dim(1316,11),pos(4,2281),type(4),blk_lbl(43)]).
block(3,[prevtxt(-1),nexttxt(-1),unit_lbl(5)]).
block(4,[dim(1316,235),pos(4,16),type(4),blk_lbl(42)]).
block(4,[prevtxt(-1),nexttxt(-1),unit_lbl(7)]).
block(5,[dim(206,10),pos(1101,1884),type(4),blk_lbl(43)]).
block(5,[prevtxt(-1),nexttxt(-1),unit_lbl(8)]).
block(6,[dim(3,374),pos(1101,1895),type(6),blk_lbl(71)]).
block(6,[prevtxt(-1),nexttxt(-1),unit_lbl(8)]).
block(7,[dim(4,374),pos(1303,1895),type(6),blk_lbl(71)]).
block(7,[prevtxt(-1),nexttxt(-1),unit_lbl(8)]).
block(8,[dim(206,10),pos(1101,2270),type(4),blk_lbl(43)]).
block(8,[prevtxt(-1),nexttxt(-1),unit_lbl(8)]).
block(9,[dim(183,357),pos(1112,1903),type(1),blk_lbl(11)]).
block(9,[prevtxt(99),nexttxt(99),unit_lbl(8)]).
block(10,[dim(536,4),pos(4,2276),type(2),blk_lbl(31)]).
block(10,[prevtxt(-1),nexttxt(-1),unit_lbl(5)]).
block(11,[dim(259,324),pos(4,1945),type(1),blk_lbl(11)]).
block(11,[prevtxt(99),nexttxt(12),unit_lbl(5)]).
block(12,[dim(260,324),pos(277,1945),type(1),blk_lbl(11)]).
block(12,[prevtxt(11),nexttxt(13),unit_lbl(5)]).
block(13,[dim(258,325),pos(554,1948),type(1),blk_lbl(11)]).
block(13,[prevtxt(12),nexttxt(14),unit_lbl(5)]).
block(14,[dim(259,325),pos(827,1948),type(1),blk_lbl(11)]).
block(14,[prevtxt(13),nexttxt(99),unit_lbl(5)]).
block(15,[dim(1316,91),pos(4,458),type(3),blk_lbl(31)]).
block(15,[prevtxt(-1),nexttxt(-1),unit_lbl(6)]).
block(16,[dim(208,2),pos(1100,551),type(6),blk_lbl(81)]).
block(16,[prevtxt(-1),nexttxt(-1),unit_lbl(6)]).
block(17,[dim(208,1144),pos(1100,725),type(1),blk_lbl(11)]).
block(17,[prevtxt(99),nexttxt(99),unit_lbl(6)]).
block(18,[dim(427,5),pos(7,551),type(4),blk_lbl(43)]).
block(18,[prevtxt(-1),nexttxt(-1),unit_lbl(1)]).
block(19,[dim(205,372),pos(9,751),type(1),blk_lbl(11)]).
block(19,[prevtxt(99),nexttxt(20),unit_lbl(1)]).
block(20,[dim(207,372),pos(227,751),type(1),blk_lbl(11)]).
block(20,[prevtxt(19),nexttxt(99),unit_lbl(1)]).
block(21,[dim(205,619),pos(7,1243),type(1),blk_lbl(11)]).
block(21,[prevtxt(99),nexttxt(22),unit_lbl(2)]).
block(22,[dim(206,619),pos(225,1243),type(1),blk_lbl(11)]).
block(22,[prevtxt(21),nexttxt(99),unit_lbl(2)]).
block(23,[dim(205,330),pos(444,1117),type(1),blk_lbl(11)]).
block(23,[prevtxt(99),nexttxt(45),unit_lbl(3)]).
block(24,[dim(206,275),pos(663,1170),type(1),blk_lbl(11)]).
block(24,[prevtxt(45),nexttxt(99),unit_lbl(3)]).
block(25,[dim(202,330),pos(880,1117),type(4),blk_lbl(41)]).
block(25,[prevtxt(-1),nexttxt(-1),unit_lbl(3)]).
block(26,[dim(204,293),pos(445,1570),type(4),blk_lbl(41)]).
block(26,[prevtxt(-1),nexttxt(-1),unit_lbl(4)]).
block(27,[dim(205,293),pos(663,1570),type(1),blk_lbl(11)]).
block(27,[prevtxt(99),nexttxt(28),unit_lbl(4)]).
block(28,[dim(205,293),pos(884,1570),type(1),blk_lbl(11)]).
block(28,[prevtxt(27),nexttxt(99),unit_lbl(4)]).
block(29,[dim(647,377),pos(444,556),type(4),blk_lbl(41)]).
block(29,[prevtxt(-1),nexttxt(-1),unit_lbl(6)]).
block(30,[dim(1316,51),pos(4,385),type(1),blk_lbl(11)]).
block(30,[prevtxt(0),nexttxt(99),unit_lbl(7)]).
block(31,[dim(1082,57),pos(4,1884),type(2),blk_lbl(31)]).
block(31,[prevtxt(-1),nexttxt(-1),unit_lbl(5)]).
block(32,[dim(208,37),pos(1100,564),type(2),blk_lbl(31)]).
block(32,[prevtxt(-1),nexttxt(-1),unit_lbl(6)]).
block(33,[dim(208,51),pos(1100,617),type(2),blk_lbl(31)]).
block(33,[prevtxt(-1),nexttxt(-1),unit_lbl(6)]).
block(34,[dim(208,53),pos(1100,670),type(2),blk_lbl(31)]).
block(34,[prevtxt(-1),nexttxt(-1),unit_lbl(6)]).
block(35,[dim(427,46),pos(7,569),type(2),blk_lbl(31)]).
block(35,[prevtxt(-1),nexttxt(-1),unit_lbl(1)]).
block(36,[dim(427,54),pos(7,635),type(2),blk_lbl(31)]).
block(36,[prevtxt(-1),nexttxt(-1),unit_lbl(1)]).
block(37,[dim(427,41),pos(7,697),type(2),blk_lbl(31)]).
block(37,[prevtxt(-1),nexttxt(-1),unit_lbl(1)]).
block(38,[dim(647,56),pos(444,994),type(2),blk_lbl(31)]).
block(38,[prevtxt(-1),nexttxt(-1),unit_lbl(3)]).
block(39,[dim(647,48),pos(444,1055),type(2),blk_lbl(31)]).
block(39,[prevtxt(-1),nexttxt(-1),unit_lbl(3)]).
block(40,[dim(427,45),pos(7,1140),type(2),blk_lbl(31)]).
block(40,[prevtxt(-1),nexttxt(-1),unit_lbl(2)]).
block(41,[dim(427,38),pos(7,1190),type(2),blk_lbl(31)]).
block(41,[prevtxt(-1),nexttxt(-1),unit_lbl(2)]).
block(42,[dim(647,39),pos(444,942),type(1),blk_lbl(12)]).
block(42,[prevtxt(-1),nexttxt(-1),unit_lbl(6)]).
block(43,[dim(647,37),pos(444,1467),type(2),blk_lbl(31)]).
block(43,[prevtxt(-1),nexttxt(-1),unit_lbl(4)]).
block(44,[dim(647,41),pos(444,1521),type(2),blk_lbl(31)]).
block(44,[prevtxt(-1),nexttxt(-1),unit_lbl(4)]).
block(45,[dim(206,41),pos(663,1118),type(1),blk_lbl(11)]).
block(45,[prevtxt(23),nexttxt(24),unit_lbl(3)]).
unit(1,blocks(5),blockmum([18,19,20,35,36,37])).
unit(2,blocks(4),blockmum([21,22,40,41])).
unit(3,blocks(6),blockmum([23,24,25,38,39,45])).
unit(4,blocks(5),blockmum([26,27,28,43,44])).
unit(5,blocks(5),blockmum([3,10,11,12,13,14,31])).
unit(6,blocks(4),blockmum([15,16,17,29,32,33,34,42])).
unit(7,blocks(5),blockmum([0,1,2,4,30])).
unit(8,blocks(5),blockmum([5,6,7,8,9])).
```

Figure 1.5: Logical description derived from document image.

in Figure 1.3. As can be seen from this figure, the root node for each tree is the entire document, and the leaf nodes represent the individual blocks in the document.

The logical description in Figure 1.5 is contained in a text file that can be edited using a standard text editor. By modifying this description, we effectively modify the logical structure and hence the physical layout of the original document. Therefore, if the reverse derivation process, i.e., the transformation from the logical structure to the physical structure of the document, were to be applied, we could then create a document that was different in structure and/or content by appropriately modifying the logical description. This would be an invaluable tool in designing the formats of different kinds of office documents and periodicals. Of course, any modification of the logical structure would have to be done by following very strict rules, so that the modified structure conforms to the constraints that enable it to be translated back into a physical structure that still obeys the geometric constraints imposed on the original document.

One way to reconstruct a physical structure from its logical symbolic description is to translate the logical description into a set of formatting commands, such as those contained in the LaTeX document formatting system. Since the logical structure also contains the physical coordinates of each of the original blocks in the image, the *picture* environment of LaTeX [32] can be utilized to good effect. Geometric information for each block can be translated into the appropriate formatting command, and when the resulting file of formatting commands is processed by LaTeX, it produces a line-drawing showing the original physical structure of the document. Figure 1.7 shows the results of performing such an operation on the document in Figure 1.3. The figure shows each of the blocks in the original image.

In order to make documents accessible through on-line document libraries, it is also pos-

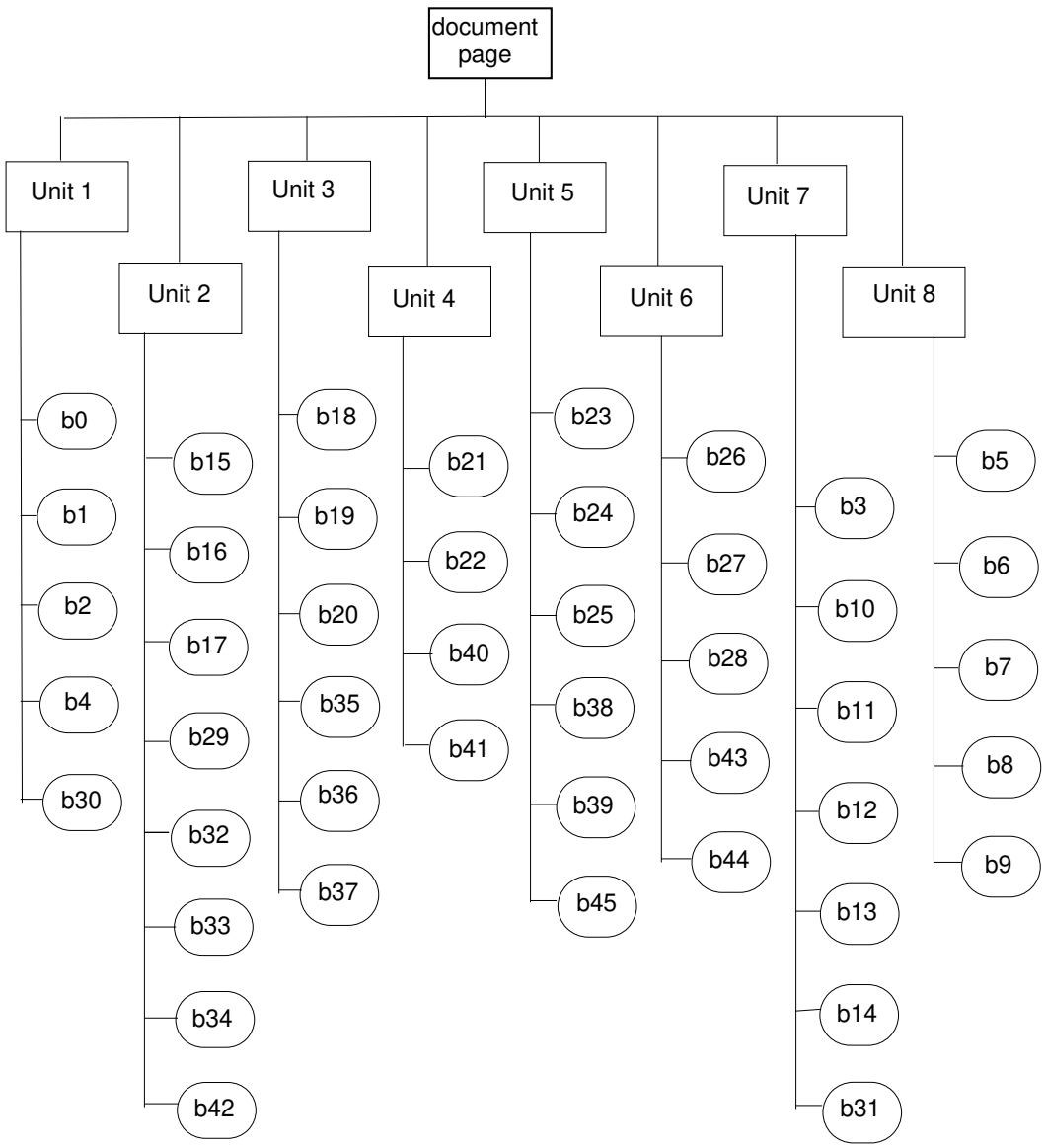


Figure 1.6: Logical structure tree for example image.

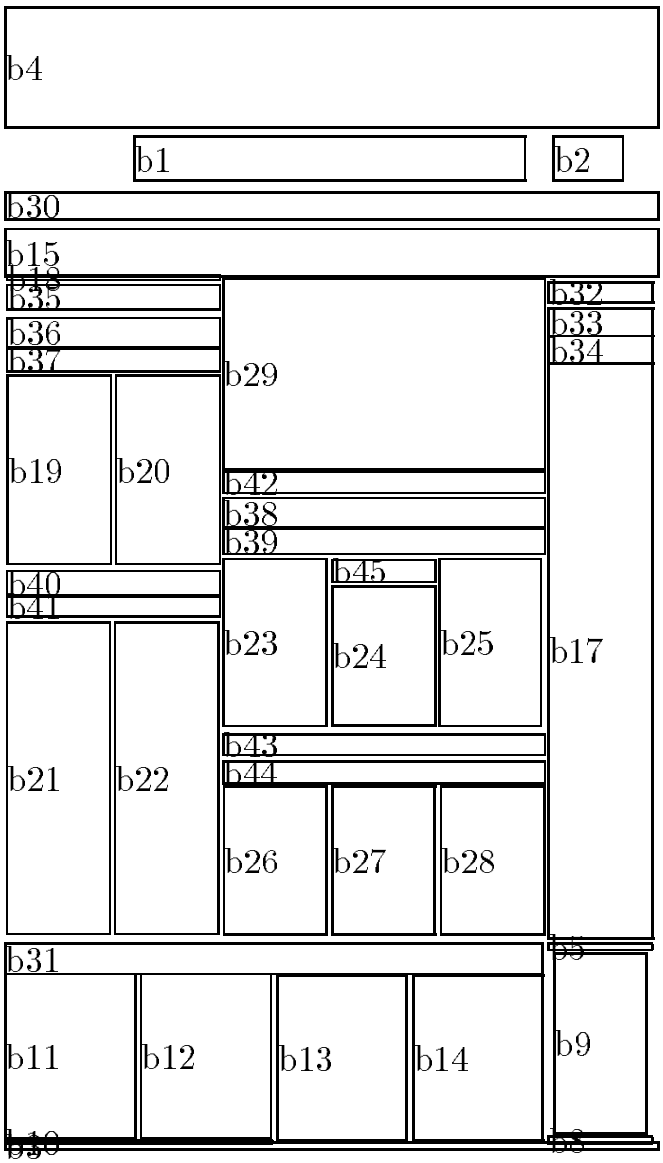


Figure 1.7: Reconstruction of physical structure of example image.

sible to convert the above-mentioned LaTeX form into *hypertext* [51] using a *LaTeX2HTML* conversion program that translates a document formatted for LaTeX into an "HTML" form (HTML stands for HyperText Markup Language, which is an implementation of the Standard Generalized Markup Language (SGML) [28] and is commonly used to represent hypertext documents). This would allow access to the original document contents in a form that is unconstrained by the original physical structure.

1.2 Motivation

In order to read and understand a document, human beings use a lot of world knowledge about documents. For example, we can read a specific story on a newspaper page without straying into other adjacent stories. We also can peruse headlines and other titles without reading the rest of the story, or simply look at pictures on a page without reading any text (in fact, children start out by looking at pictures on a page before they learn to read text).

Human beings are able to determine the logical structure of a document by using a variety of visual cues. These cues include proximity and geometrical relations of various printed components on a document page, as well as other visual highlights such as different typefaces and font styles, bounding boxes around stories or articles, horizontal and vertical lines separating rows or columns of information, etc. Recognizing these visual cues is a learned process among humans. Often, we have to be *taught* how to read and understand a document. Therefore, there must be a set of rules that humans use to perform this type of document understanding.

There are of course limitations to this human ability to read documents. Sometimes we err in reading a document, as for example when we cannot easily determine whether

a photograph "belongs" to a text block. These situations happen primarily when our knowledge about the physical and/or logical structure of that document is incomplete (or simply incorrect), or when we apply our learned rules improperly, or when the physical document layout is not completely consistent.

Any document, even complex ones like newspapers, forms, annotated schematic diagrams, etc., can be understood by a human who has learned the rules for understanding that particular class of documents. Thus, each of these types of documents must have rules for physical layout and logical structure that can be formalized so that the understanding of these documents can be performed by a computer system. The motivation for this work, therefore, is to formalize these rules for document understanding by constructing a model that enables rule-based derivation of the logical structure of document images.

Deriving the logical structure of a document is a critical step in document image understanding. In fact, it can be considered to be document *structure understanding*, which can then lead to *semantic understanding*, which involves text reading. Once the logical structure is derived, the appropriate textual parts can be read by an optical character recognition machine, and "understood" by a natural language understanding system.

This work also provides other benefits. First, extracting the logical structure of a document allows us to analyze the document semantically without being constrained by its geometrical layout. Second, such analysis gives us insights into document layout conventions and their effectiveness. Third, the derivation of logical document structure from physical structure should enable us to develop formalisms that can effect physical to logical transformation of other types of images.

Document logical structure derivation is important also from the viewpoint of *office*

automation. Office documents frequently need to be transmitted electronically to other locations. If an office document can be digitally scanned, "translated" to derive its logical structure, stored in a compact representation, transmitted to another location, and then reconstructed from the compact representation into its original format on paper, it would contribute greatly to efficiency and convenience of handling and storage methodologies for documents.

Last but certainly not least, document logical structure derivation is a crucial step in the development of *digital document libraries.* Currently, extraction of logical sections of a document in response to user queries is extremely difficult and cumbersome, and is thus not feasible in real-time information retrieval systems. Extraction of the logical structure of a document will allow individual logical components to be stored separately in digital libraries, thus making their indexing and access much faster and easier.

1.3 Research Objectives

The main objectives of this research are to develop a computational model for deriving the logical (layout-independent) representation of a document from its physical (layout-dependent) representation, and to utilize the formalisms thus developed to design a working system that analyzes digitized images of newspaper pages and extracts the logical structure of a newspaper page.

Figure 1.8 shows how the different tasks undertaken in this research relate to each other within the context of the overall document understanding problem by showing the different states of a document as it is processed by a document understanding system. As shown in the figure, complete document understanding also includes high-level text and graphics

understanding. The broken-line box in the figure marks the scope of my dissertation.

Three fields of exploration, namely, artificial intelligence, image analysis, and office automation, have made significant contributions to the field of document image understanding, and this work attempts to incorporate each of these three fields of research. The field of image analysis gives us techniques to isolate and categorize printed blocks in an image, while the field of office automation systems gives us techniques that can be used to obtain a symbolic description of the logical structure of a document page by formally defining the syntactic structure of a page. The field of artificial intelligence gives us techniques to design a knowledge-based methodology that can be used to effect the physical-to-logical structure transformation.

The domain used for this work is primarily that of newspaper images, although other document classes such as journals and magazines are also discussed. Newspapers provide a wide variety of layouts that are determined according to editorial conventions and are therefore extremely interesting for our analysis purposes. The formatting of newspaper pages is more complicated than, for instance, office documents, and provides an opportunity for developing a wider variety of interesting techniques.

The significant accomplishments of this research, which will be described in more detail in later chapters, are as follows:

- The computational model is implemented in a coherent system that makes use of past techniques in a novel organization.

- The model describes a general process for determining reading order as well as block classification and grouping.

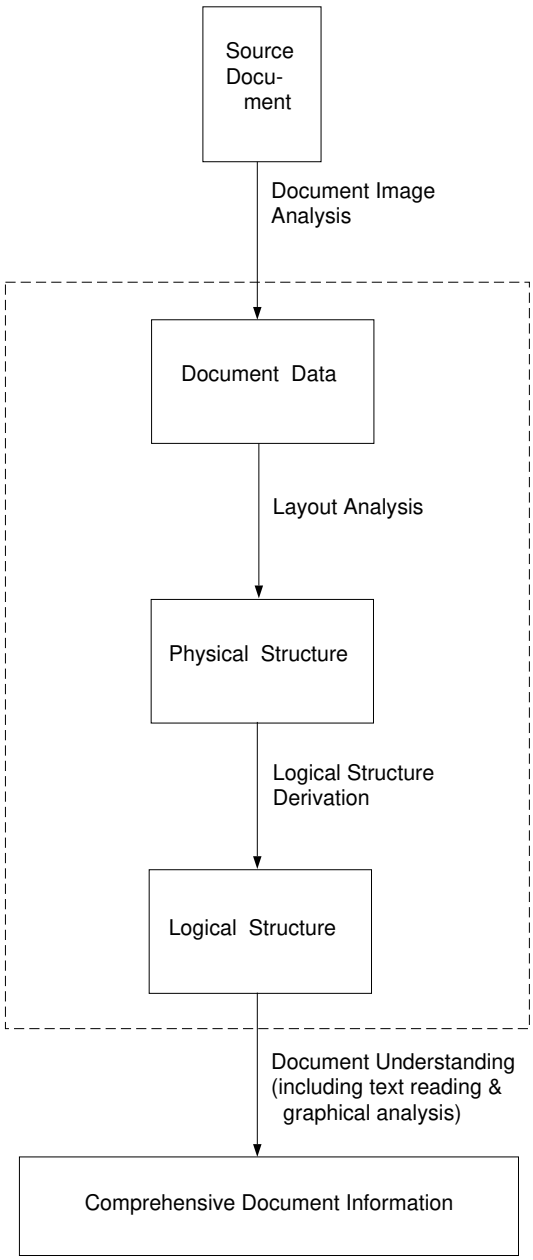


Figure 1.8: Different document states during document understanding.

- The approach deals with multi-unit documents such as newspapers.
- Evaluation is done by recreating the physical structure of analyzed documents.
- Interaction between block classification and block grouping provides flexibility to the model.
- The model separates the overall control from the document analysis operations.

1.4 Organization of the Dissertation

Chapter 2 gives a brief review of the literature in document image analysis, knowledge-based image understanding, and document layout analysis. It also discusses the problem of document image understanding as it relates to the above areas. Chapter 3 describes the problem of deriving the logical structure of a document, which is part of the general document image understanding problem. It also formulates an approach towards the solution of this problem. Chapter 4 describes the techniques used in the application of this approach towards the development of a system that derives the logical structure of newspaper pages. Chapter 5 further explores the types of domain knowledge and document world knowledge required by this system, and Chapter 6 describes the specific details about the implementation of this system, including the image processing operations required for the system to completely analyze a document image. Chapter 7 describes the results that have been obtained using our model, and analyzes the performance of the implemented system. Chapter 8 summarizes the major points made in this dissertation and discusses their significance.

Chapter 2

Background

As a domain for serious research, document image understanding has rapidly gained in importance over the years. This chapter discusses the development of this field and how it has evolved and incorporated techniques from various related research areas. The areas discussed are: document image analysis, knowledge-based reasoning, and document layout analysis. This chapter also discusses the current status of research in document image understanding, and how it relates to the document logical structure derivation problem described in this dissertation.

2.1 Document Image Analysis

In the document image understanding problem, the first task that must be performed is the segmentation and classification of all the blocks in the document image. This constitutes document image *analysis*, and is the most commonly studied aspect of document image understanding.

It is interesting to note that the research in document image analysis stemmed from the field of image analysis and primarily concentrates on techniques like segmentation and

filtering that can extract objects of interest from an image. Seminal work in the area of document image analysis includes the use of a run-length smoothing algorithm (RLSA) for the segmentation and classification of digitized printed documents into regions of text and images (Wong et al. [83]), and the use of recursive horizontal and vertical cuts (known as "X-Y cuts") to divide a document image into its constituent printed blocks (Nagy et al. [48]).

In the RLSA method, the document image is "smeared" by converting strings of white pixels of run-length smaller than a given threshold into black pixels. This smearing operation is done separately in the horizontal as well as vertical directions, and the two resulting smeared images are then combined using a logical AND operation to give a final smeared image. Segmentation of this smeared image is then done using projection profiles, yielding the printed blocks in the original image. Varying the smearing threshold allows the segmentation of different types of images. This method even works for images that contain some "noise" (i.e., spurious black pixels introduced in the printing, reproduction or scanning process), although large amounts of noise will result in inaccurate segmentation. Also, this method does not require a known, *a priori* document syntax. However, skew correction needs to be done first, and also block spacing thresholds need to be known in advance.

In the X-Y cut method, the document is recursively divided by using the projection profile technique. In this method, projection profiles are calculated (i.e., all the black pixels are counted along each row/column of the document image) in the horizontal as well as vertical directions. A zero pixel count indicates the presence of a "white space", i.e., a space along the entire length/width of the document image that is devoid of print. A horizontal or vertical "cut" is then made along that white space, yielding two document sub-images. For each of these sub-images, the same operation is done in the other direction

(i.e., if the first cut was vertical, then this one would be horizontal). Thus, this process is recursively applied to each sub-image of the document until no further cuts can be made. The resulting blocks of black pixels are then considered to represent the indivisible blocks of printed matter in the document. For images containing noise, the zero pixel threshold for detecting white space can be changed to a low non-zero value, but the results are not as accurate.

The RLSA and X-Y cuts are bottom-up segmentation techniques that give fairly accurate segmentation results for document images. Combining the two techniques can provide even more accurate segmentation results. (A method using using feature spaces and texture analysis for segmenting and classifying blocks in newspaper pages is described by Wang & Srihari [78]. This method uses a combination of the RLSA and X-Y cuts method for block segmentation.) These two techniques work best when the image is not skewed significantly from the horizontal. One effective way to remove document skew is by using the Hough transform method, as demonstrated by Srihari & Govindaraju [70].

Another segmentation method that provides good results even in the presence of noise is the "docstrum" (document spectrum analysis) approach (O'Gorman [60]), which also performs skew correction. In this method, all the connected components in the image are first detected, and then blocks are generated in a bottom-up fashion by finding the neighbors of each component and clustering the different components. This segmentation method yields very good results for various types of document images, particularly journal and newspaper pages, and has been used in my research work as the primary segmentation method. This method is described in more detail in Section 6.4.

Document image analysis techniques have been applied to various types of document images. For example, techniques used in analyzing pieces of letter-mail to locate the des-

tination address are discussed by Wang & Srihari [77] and by Srihari et al. [71]. A survey of document image analysis by Srihari [69] gives a comprehensive overview of techniques in all aspects of document image analysis.

2.2 Knowledge-Based Image Analysis

One of the primary interests in document image understanding is that of using knowledge about the basic structure of different kinds of documents to analyze and identify the different components of a document. This is known as knowledge-based image analysis, or increasingly, *image understanding*.

Knowledge-based image understanding systems have been used in the past in various image domains, e.g., aerial photographs, natural outdoor scenes, medical images, etc. The importance of using knowledge-based techniques in image analysis was mentioned early on by Shapiro [65].

Many image analysis systems have used scene or object models to represent object characteristics and spatial relationships among components. Brooks [4] & [5], Kim et al. [29], Levine & Shaheen [37], Sloan & Bajcsy [66], and Wesley & Hanson [80] have described the use of models to perform object recognition in specific image domains. Ballard et al. [2] and Binford [3] have given comprehensive overviews of model-based image analysis systems. Systems like VISIONS by Hanson & Riseman [22], ACRONYM by Brooks et al. [6], and SIGMA by Matsuyama & Hwang [41] are quite well-known in the literature. Model-based text recognition has been discussed by Srihari [68]. Model-based systems can achieve high degrees of accuracy within a particular image domain, since the restricted domain allows for a very specific definition of a model based on the object characteristics and relationships

in that domain. However, some model-based systems may lack flexibility due to their dependence on rigid domain models.

In the development of knowledge-based image analysis systems, various other control strategies for the analysis of images and representations for image data have been designed. Among them is the use of semantic networks to represent knowledge (Bunke & Sagerer [7], Tanimoto [73]) and frame-based knowledge representation schemes (Numao & Ishizuka [59]). Many early approaches used various knowledge-based techniques such as evidence accumulation (Matsuyama et al. [42]), image feature maps (McKeown & Denlinger [43]), constraint satisfaction (Russell [64]), and quadtree refinement (Tucker [76]). Also, various knowledge representation schemes were developed for image analysis (Latombe & Lux [34], Matsuyama [40], Reynolds et al. [63], and Wesley & Hanson [80]). As the above shows, a wide variety of intelligent techniques have been applied to the analysis of images, with varying degrees of success, and it would be difficult to list all the different approaches that have been used over the years.

Rule-based inferencing strategies have been very popular for image analysis, particularly in domains where image characteristics and object relationships can be effectively represented in terms of rules. For example, rule-based strategies for image interpretation have been proposed by Weymouth et al. [81], and a knowledge-based computer vision system using rules has been described by Levine [36]. Other uses of rule-based systems for image analysis include an expert system for low-level image segmentation of visual scenes by Nazif & Levine [49] and a rule-based system for aerial imagery by McKeown et al. [44]. The application of knowledge-based techniques to document image understanding have been discussed by Kubota et al. [30], who describe the application of a production system concept to an experimental document understanding system. Fisher et al. [19] describe a rule-based

system for segmenting a document image into text and non-text blocks. The rule-based systems used in the document image domain, however, have not fully exploited the depth and breadth of knowledge that is available about specific document domains (as will be discussed in later chapters).

Rule-based systems have the advantage that knowledge can be represented in a declarative manner that does not dictate the search strategy to be followed by the inferencing system. They, however, have the disadvantage that new knowledge about the domain has to be explicitly added to the rule base, and may lead to inconsistencies with existing rules. This disadvantage is usually resolved by performing checks to ensure rule base integrity when adding new rules.

The representation of domain knowledge in such systems has also been extensively studied. One of the architectures proposed for maintaining domain knowledge in a system is the blackboard control architecture, which was used very effectively in the HEARSAY II speech understanding system (Erman et al. [15]). The blackboard control architecture has been described in detail by Hayes-Roth [23]. An excellent overview of the blackboard model of problem solving has been written by Nii [52] & [53]. The use of a blackboard control architecture in document image understanding is also described in my research work (Niyogi [57]) leading to the development of this dissertation. Blackboard control systems provide a flexible control environment that facilitates the interaction of distributed problem-solving processes to provide an integrated solution.

Many of the principles introduced in knowledge-based image analysis, such as the use of rules to formalize knowledge about object features and relationships, and also the use of global data structures such as blackboards to represent domain knowledge, are suitable for use in the domain of document images, since documents have well-defined structures

and object relationships. Since rules can be used to declaratively store knowledge about document characteristics, I have used rules in my approach to represent document domain knowledge. Also, since multi-level inference strategies require interaction with a globally accessible data representation, I have used a global data structure to store data and inferences in my approach to knowledge-based document image understanding.

2.3 Document Layout Analysis

Document layout analysis involves the extraction of the physical structure of a document. Some of the early work in document image analysis, such as Nagy's X-Y cuts [48] and Wong et al.'s RLSA algorithm [83], attempted to extract the complete physical structure of the document using bottom-up techniques. However, since no knowledge of image features was used, the structure obtained was only an approximation and the systems could be defeated by elaborate document structures that were not amenable to the usual decomposition methods.

More recently, researchers have worked on the interpretation of office documents through the classification of the blocks in a document image. Anastasil, developed by Dengel & Barth [11], is a knowledge-based system that uses X-Y cuts (i.e., alternating horizontal and vertical cuts along regions of white space to progressively divide the document into physical rectangular regions) on images of standard office documents and forms a geometric tree whose nodes represent possible labelings of the document. A best-first search of the tree yields the correct labeling. This system, however, deals only with block classification and not with block grouping. Ciardiello et al. [8] describe an office document processing system that uses statistical techniques to extract features of blocks, and then to merge adjacent textual blocks based on similarity of features prior to text interpretation. However, this system's

categorization is limited to "text" and "non-text", and no logical grouping of blocks is attempted. Esposito et al. [17] have described a system that can classify a given office document into one of several classes (e.g., letter, form, magazine index page, etc.) based on standard geometric relations between regions (a "page layout signature"). However, no semantic labeling or logical grouping of individual blocks is attempted.

Layout analysis methods have been suggested for other restricted document types such as table-form documents and library cataloging cards (Watanabe et al. [79]). Some approaches to defining document structures, such as the use of attribute grammars (Lemone [35]) have concentrated solely on the logical structure of a document and not on its physical structure. Mizuno et al. [46] have used character segmentation and feature vector computations to extract the layout of Japanese newspapers (no qualitative knowledge about layout is used). Derrien-Peden [12] has used a frame-base technique to perform layout analysis for scientific papers. Esposito et al. [16] have used automatically detected geometrical features of documents to perform layout analysis and logical labeling of document components.

Layout analysis has long been an area of interest to magazine and newspaper editors, who have to decide what particular layout of paragraphs or stories is the most efficient and also makes the best impression on its readers. Even before the time of electronic typesetting, layout analysis has been an important aspect of publishing. Several books, notably the ones by Arnold [1], Crowell [10], Gibson [21], Hurlburt [24], Hutt [25], Nelson [50], and White [82], give us important clues about the structural make-up of different kinds of published documents. As these books have mentioned, each type of document that can be electronically typeset has very specific typesetting heuristics that are faithfully followed within the publishing industry (of course allowing for creative variations, particularly in newer magazines). These books discuss document formatting techniques, document layout

conventions, electronic typesetting, and heuristics for document markup and layout. Many of the layout rules used in my research have been taken from the above books on document layout.

The field of office automation has also made substantial contributions to existing knowledge about document layout analysis. Doster [14] describes how a document description database can be used to represent knowledge about the characteristics of a document image. Peels et al. [61] describe the opposite process, i.e., document formatting, where the logical description of a document (using a document description language) is transformed into a physical formatted representation. The most commonly used standard representations for describing document structure are the Standard Generalized Markup Language (SGML) [28] and the Office Document Architecture (ODA) [27]. The problem of logical document structure derivation is important for office automation, since a logically analyzed document can be electronically transmitted to other locations and re-constructed from its logical description, thus bringing closer the utopian vision of a "paperless" office. In my work, I have attempted to perform document image analysis in a manner suitable to document reconstruction, such that the logical structure obtained from a document image can be used to reconstruct the original document physical structure.

2.4 Document Image Understanding

Document image understanding involves the syntactic and semantic interpretation of the various components of a document image, and requires domain knowledge about document features and characteristics. Previous work in this area includes an elucidation by Nagy [47] on the types of knowledge required for document image understanding. Survey papers in the area of document image understanding include those by Tang et al. [72] and Srihari [69].

Also, various approaches to the solution of this problem have been proposed in the past few years. For example, Taylor et al. [74] have attempted to use multiple knowledge sources and a blackboard control architecture to derive document structure and then use linguistic knowledge to label various document components. Luo et al. [39] have used a rule-based approach to interpreting the physical and logical structure of Japanese newspaper pages. Fujihara & Mukerjee [20] have defined a qualitative language to specify spatial constraints among blocks in journal pages. Multi-domain document layout understanding has been discussed by Lam & Srihari [31].

Document logical structure derivation is a crucial component of the overall document image understanding process. Thus, recent interest in this area has been high. Tsujimoto & Asada [75] have suggested a rule-based method for deriving the physical and logical structures of a multi-articled document, Yamashita et al. [84] have used relaxation techniques to derive the logical labeling of blocks in Japanese patent application documents. Fisher [18] has proposed a method for deriving logical structures that works on a limited set of single-unit documents (thus, no block grouping or read-order determination). Ingold & Armangil [26] have used formal document descriptions to derive the layout and logical structures of simple legal texts using artificial data. Porter & Rainero [62] have described a document "reconstruction" method, by which a document is analyzed to derive its physical structure, and the logical document structure is "reconstructed" from the document layout description in terms of document "views" of the original structure. Each of these have suggested methods to derive the physical and logical structure of a document, with varying degrees of success.

A new approach

As is shown in detail in later chapters, the approach described here for document logical structure derivation uses some tools that are similar to the previous approaches mentioned. For example, a knowledge base of rules that describes the document physical and logical layout constraints is used, similar to Tsujimoto & Asada [75]. The actual form of the multi-level rule base is similar to Nazif & Levine's [49], although in their work the rule base was used for low-level image segmentation.

Like the previous work in document image understanding mentioned above, my approach analyzes a document image to infer the classes and labels of the blocks in the image, and like some of the above-mentioned work (most notably the work by Tsujimoto & Asada [75]), my approach also combines these blocks to make up logical units such that all the blocks in an unit are logically related to one another. However, it goes further in its scope. In addition to the above, my approach determines the logical reading order of the text blocks in each unit, thus enabling a document reader to read the different blocks comprising a story or article in the appropriate order. No previous work has tackled the determination of reading order of text blocks as part of the document image understanding problem; all the approaches that have used block grouping into units have used pre-determined document templates for pattern matching and have implicitly assumed a standard top-to-bottom, left-to-right reading order for text blocks. My work is unique in that it uses spatial constraints and document domain knowledge to not only group the blocks into units but also derive the reading order of documents.

Also, most previous work deals with office documents, or with pages from technical journals (although Japanese newspapers have also been used by some, as mentioned above,

but these approaches have not gone beyond logical labeling). Office documents and even technical journal pages are typically single-articled documents, i.e., they typically contain a single logical unit. My approach deals with newspaper images, which are structurally more complicated than office documents, since a newspaper page typically contains several units which are structurally related but which may or may not be logically related.

Thus, the merits of my approach include the fact that it tackles a class of documents (i.e., newspaper images) that other researchers in document image understanding have not yet fully explored, and it also deals extensively with the concept of the determination of the reading order of text blocks within a logical unit (based on the spatial inter-relations between blocks) *in conjunction with* logical block grouping, whereas previous work dealt mainly with layout analysis and structure understanding through block classification, and in some cases limited grouping, without regard to textual reading order. Also, the modular nature of the knowledge base in this model improves upon previous work and allows for the potential to process a variety of document classes by substitution of the appropriate set of knowledge heuristics describing the appropriate document class.

Chapter 3

A Model for Document Logical Structure Derivation

3.1 Analysis of Document Structure

Various types of documents exist in the business world, and each has a unique structure associated with it. Office documents, newspapers, journals, forms, etc., each have specific characteristics associated with them that enable a human to distinguish between them with only a cursory glance. One of the things that distinguishes these different classes of documents from one another is the *layout* of the document, i.e., the properties and physical arrangement of the different regions of printed matter in the document. The study of these distinct layout patterns and how to formalize them is commonly known as *layout analysis*.

With the advent of electronic typesetting, layout analysis has become an integral part of the publishing world. Books, newspapers, magazines, journals, and most other mass-produced documents are published with the help of electronic layout software. For example, *The Buffalo News* publishes four different regular editions (sunrise, afternoon, metro, and evening) of the same newspaper every weekday, plus regional editions. As news develops during the day, the editions get changed to reflect these breaking news stories. As a result,

the front page, for example, may undergo several changes in format during the course of the day, as stories rise or fall in importance relative to other stories (or in many cases get included in or dropped from that edition). Also, the regional editions may contain the same stories as the metro edition but in a different layout format. To accommodate these changes and variations, the layout is modified electronically by the layout editor and his staff, who are constrained by the layout conventions of that particular publication. These layout conventions constitute the *structure* of that newspaper.

The *physical* structure of a document page reflects the layout of different geometrical units within the document in a specific presentation format. The basic element of the physical structure of a document is a "block", which is defined as a homogeneous geometric region of printed matter in the document (i.e., each connected component within a block is of similar type and size). Blocks are separated from one another by regions of white space.

Newspapers are examples of complex documents. Each page of a newspaper typically contains many blocks of textual or graphical information, arranged so that each rectangular block of information is interconnected with its adjacent blocks so as to make a grid-like pattern of blocks on the page (as described by Hurlburt [24]). The syntax of the physical structure of a typical newspaper page is given (in Extended BNF format) in Figure 3.1 (a). The OR relation in the figure represents the fact that the categories of blocks are mutually exclusive.

The right-hand side of the last syntax line in Figure 3.1 (a) enumerates the basic block types into which newspaper blocks can be categorized. A "large-text" block contains large letters (typically 24-pt. and above). A "medium-text" block contains letters between 14-pt. and 20 pt. A "small-text" block contains 8-pt. to 12-pt. size letters, which is typically the range for regular newspaper text. A "line-drawing" block contains one or more large

```
<document>  ::=  { <page> }
<page>      ::=  { <block> }
<block>     ::=  <large-text> | <medium-text> | <small-text> |
                 <line-drawing> | <half-tone> |
                 <boundary>
<boundary>  ::=  <horizontal-line> | <vertical-line> |
                 <line-rectangle>


where { X } means one or more occurrences of X,
      and ::= means ``consists of'',
      and | means the logical OR relation.
```

(a) Physical structure

```
<document info>  ::=  { <unit> }
<unit>           ::=  <title> | <graphical area> |
                                    <story> | <piction>
<piction>        ::=  [ <title> ] <photo> <caption>
<graphical area> ::=  <page banner> | <horizontal band> |
                                    <other graphics>
<story>          ::=  [ <sub-story> ] | <title> [ <sub-title> ]
                                     { <text-para> }
                                     [ <piction> ]
                           [ [ <title> ] <chart> <caption> ]
                           [ [ <title> ] <table> <caption> ]
<sub-story>      ::=  <story>


where [ X ] means zero or more occurrences of X,
      and the symbols {} and | have the same meaning as above.
```

(b) Logical structure

Figure 3.1: Syntax of physical and logical structures of a typical newspaper.

connected components. Finally, a "half-tone" block contains one large connected component with a large percentage of black pixels (typically a photograph or other graphical data).

The *logical* structure of a document reflects the hierarchy of logical units that comprise the document. Each logical unit is itself a composite of text or graphics elements. Logical structure is layout-independent; i.e., the contents of a newspaper story and of the various components in the story (e.g., text, photograph, caption, etc.), as well as the semantic relations between these components, are not affected by the manner in which the story is geometrically arranged on the newspaper page. The syntax of the logical structure of a typical newspaper page is given in Figure 3.1 (b).

In the syntax lines for the logical structure in Figure 3.1 (b), "piction" refers to a unit which contains a photograph and a caption, and perhaps a title, but does not contain any text paragraphs. (The term "piction" was first used by R. Srihari [67] in a system that used captions to understand photographs in newspaper images.) Also, a "graphical area" is typically a newspaper page banner or other mixed mode area on the page. A newspaper "story" usually has a title (major headline), perhaps a sub-title (minor headline), some text paragraphs, and some combination of photographs, charts, tables, etc. A story can also have several sub-stories, each of which has its individual logical components. As shown above, sub-stories have the same structure as stories; thus, the logical structure description of a document can be recursive in nature.

Each publisher of a document uses a set of rules to determine the physical and logical structures of the document. These *layout rules* not only determine the "look" of that particular publication (i.e., its physical structure) but also the semantic makeup of the document (i.e., the logical structure).

The problem of extracting logical structure from physical structure involves identifying the physical structure of a type or class of document, designing a transformation that maps the physical structure to the logical structure, identifying a set of layout rules that describe the physical and logical composition of the given type of document, and using these layout rules to effect the transformation from the physical to the logical structure.

Layout rules may vary widely among different types of documents, and even among different examples of the same type of document. Thus, spatial relationships between blocks in the title page of a journal article are likely to be different from those in a newspaper page, and even more different from a pre-printed form. Also, the identities of the blocks in these different types of documents are likely to be different. For example, a rule that says, "the author's name block can be found directly below the title block" may apply only to a title page of a paper and not to, say, a newspaper page. Conversely, a rule that says, "multi-line titles are always left-justified" may apply only to a newspaper page and not to other types of documents. Some rules, however, may apply to a majority of documents, e.g., "bigger titles are always placed above smaller titles". Thus, a knowledge base of layout rules for document logical structure derivation will contain some general rules that apply to a majority of documents and some specific rules that apply only to the class of document being analyzed. The construction of such rules is discussed in detail in Chapter 5.

3.1.1 Newspaper formats

An important part of the logical structure derivation process is identifying a set of layout rules that describe the physical and logical composition of the given type of document. Layout rules vary widely among different types of documents. For example, spatial relationships between blocks, as well as the identities of the blocks themselves, in the title page

of a journal article are likely to be different from those in a newspaper page. However, some features are common to most documents. In the case of newspaper images, a knowledge base of layout rules will contain spatial rules that can be inferred from newspaper layouts. Examples are:

1. Headlines occupying more than one printed line are always left-justified.

2. Captions are always below photographs, unless two or more photographs have a common caption.

3. A thin line rectangle around a set of blocks always signifies an independent story or unit.

4. Thin vertical lines are column separators.

Translated into more formal rules, the first two of the above rules would look something like Figure 3.2.

As mentioned before, different documents have different layouts. Individual newspapers have their own layout "styles" which reflect editorial policies and layout conventions. Some of the most common layout styles for newspapers are:

- The "modular" layout style (i.e., each story is put into a rectangle, and rectangles are fitted together on a page until it is filled up). *The Buffalo News* is an example of this layout style.

- The "vertical" layout style (i.e., all headlines are one-column wide, and a story starts directly under its headline). *The Wall Street Journal* is an example of this layout style.

```
1.      IF a block B1 is a headline, and a block B2 is also a headline,
            AND IF block B1 is *above* block B2,
            AND IF the left edges of the two blocks B1 and B2
                   are on the same straight line,
        THEN the two blocks B1 and B2 belong to the same unit.


2.      IF a block B3 is a photograph, and a block B4 is text,
            AND IF block B4 is *below* block B3,
            AND IF the following condition is NOT true:
                   {there exists a block B5 which is text
                      and is *next to* block B3,
                    and a block B6 which is a photograph
                      and is *next to* block B5}
            AND IF the left edges of the blocks B3 and B4
                   are on the same straight line,
        THEN block B4 is a caption, and the blocks B3 and B4
               belong to the same unit.
```

Figure 3.2: Examples of newspaper layout rules.

- The "horizontal" layout style (i.e., the page is divided up into several horizontal bands into which stories are fitted). *The Des Moines Register* is an example of this layout style.

Several newspapers adopt a hybrid layout style which combines the vertical and the horizontal styles, for example *The New York Times*, which follows different styles for its front page than it does for its other pages. Some layout conventions are common to each of these styles, but there is also plenty of room for a newspaper to display its individuality. Thus, as explained by Gibson [21], some of the rules given above (e.g., Rules 2 and 4) apply to all broadsheet (i.e., standard-size, 24 inches high and 14 inches wide) newspapers, whereas some others (e.g., Rules 1 and 3) apply only to some current newspapers. Therefore, a knowledge base of rules has to account for each of these different layout styles.

3.2 Development of a Computational Model

I have developed a computational model for the derivation of logical structure of a document. The model involves strategies for extracting the physical structure of a document (i.e., layout analysis), translating into the logical structure (i.e., logical structure derivation), and representing the logical structure in an appropriate symbolic form suitable for use by other processes that perform deeper levels of document understanding (e.g., text reading). This model is discussed in detail through the rest of this chapter. Figure 3.3 shows a schematic diagram of the computational model.

3.2.1 Framework Used

The computational model used to derive logical structure uses the following components:

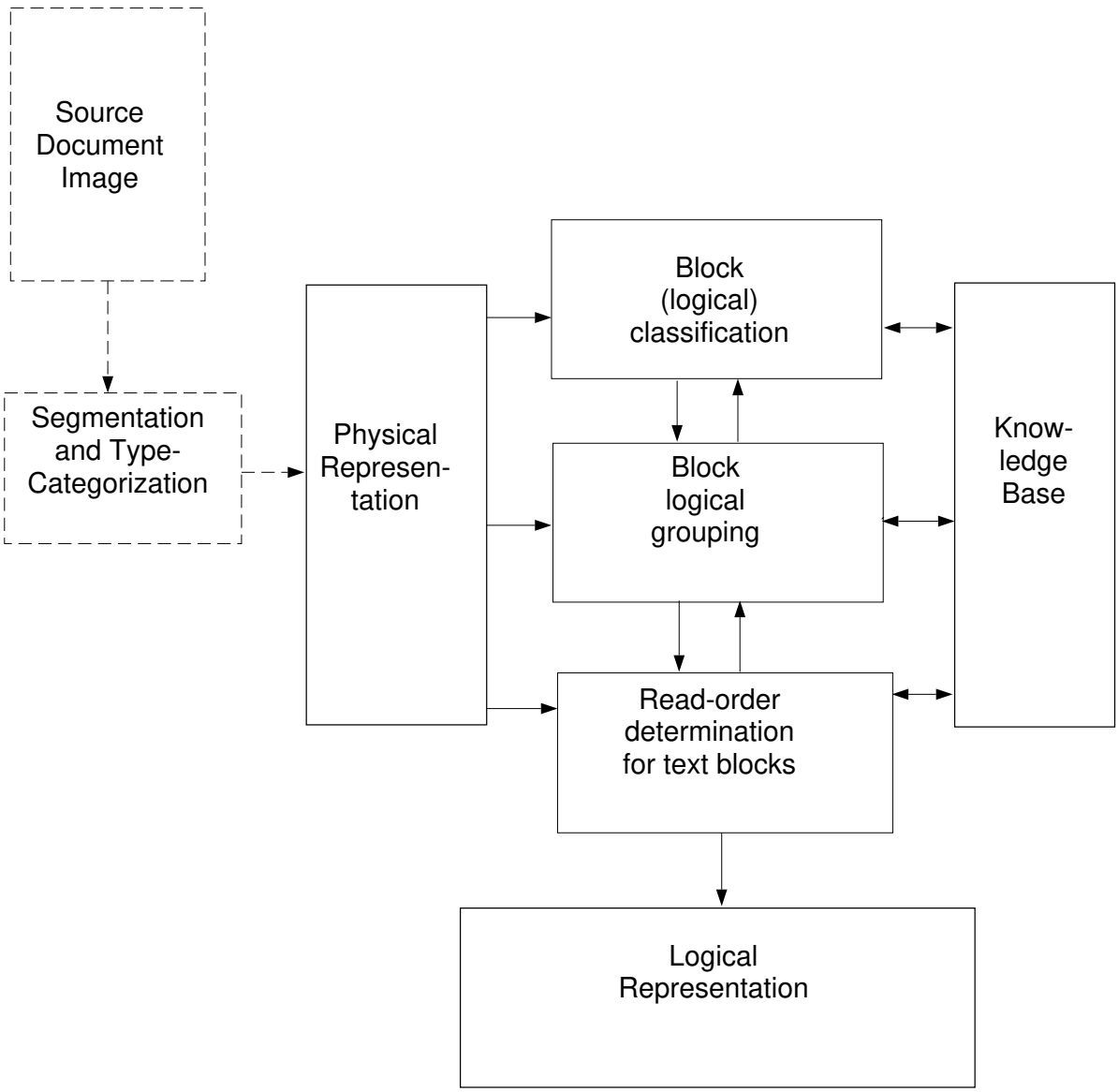


Figure 3.3: Computational model for logical structure derivation.

FRAMEWORK

1. A set of rules that identify and label all the distinct blocks in an image

2. A set of rules that group these blocks into logical units

3. A set of rules that determine the read-order of the text blocks within each logical unit

4. An inference engine that applies these rules to the document image

5. A control structure that monitors the classification, grouping and ordering processes and creates the logical representation (i.e., the symbolic description) of the document

6. A global data structure that maintains the domain and control information.

Each of these components will be discussed in more detail later in the dissertation.

3.2.2 Design Methodology

In order to create a computational model to tackle the given problem, we have developed a set of formalisms that enable us to structure the solution space and to help us design the appropriate framework for solving the problem. The formalisms that have been developed are:

1. *A mechanism for the creation and verification of the knowledge base.*

   Building a knowledge base of rules that accurately describe the physical and logical structures of a document involves close scrutiny of various properties of not only the class of documents in question, but also general properties of all documents. A representative knowledge base has to contain rules that apply to a wide variety of documents, as well as other rules that describe the specific class of documents under

consideration. We have taken a representative sample of newspaper pages and "hand-segmented" (i.e., manually measured) them in order to analyze the various blocks, their geometric and logical relationships, and the composition of the various news stories on a page. The data obtained from this data-gathering effort has been used in the development of the knowledge rules that pertain to geometric features and relationships. Thus, a knowledge base has been built that describes the structure and spatial relationships of the blocks and logical units. Also, we have used knowledge from standard textbooks on newspaper and magazine layout (Arnold [1], Crowell [10], Gibson [21], Hutt [25], Nelson [50], White [82]), which describe common newspaper and magazine layout conventions, in order to formulate the rules that comprise the knowledge base. The knowledge base of rules has then been modified based on tests of the model conducted with various newspaper images. The consistency of the knowledge base is verified by running a series of test programs every time a new rule is added, so as to detect and eliminate any cycle of rule dependencies. The rule-based system is described in Chapter 4, and the domain knowledge about document structures that is used in building up the rule base is described in Chapter 5.

2. *A formalism for describing the control behavior of the model.*

The control behavior of the overall logical structure derivation process encompasses a combination of top-down and bottom-up strategies. Layout analysis is essentially a bottom-up approach which uses techniques such as connected component analysis, run-length smoothing, and X-Y cuts (each of which has been described in Section 2.1) to derive the physical structure of a document. Logical structure derivation is a top-down process which uses heuristics about document layout conventions to drive a hypothesize-and-test strategy and extract the logical structure of the docu-

ment. Even within the logical structure derivation process, the different components are not sequentially invoked. There are rules for block classification, other rules for block grouping, and still other rules for block ordering. Since there are classification rules that depend on grouping information from neighboring blocks, and vice versa, we cannot perform the classification of *all* the blocks *before* starting on the grouping process. The control behavior thus consists of concurrent classification and grouping, with classification rules taking precedence over grouping rules at certain stages of the process. The control structure is described in more detail in Section 3.3.

3. *A mechanism to deal with incomplete data or knowledge.*

Very often, a single set of rules cannot provide a complete, unambiguous analysis of a document. For example, different sets of rules may act on a particular set of blocks and make different decisions about the identities of these blocks. When such conflicts occur, a higher-level conflict-resolution strategy is required. Often, this conflict-resolution strategy requires more information about the set of blocks in question. When the knowledge in the knowledge base is incomplete for the purpose of comprehensive analysis, we need a method of storing any incomplete inferences made, so that later evidence can be used to complete the inference. To facilitate this, we use a global data structure to store all intermediate states of computation. While the block classification, grouping, and ordering operations are being performed, all the data about inferred classes, groups, and orders are stored in this global data structure, which can be perused and/or updated by the inferencing mechanism in order to help it make further inferences about the document. The global data structure is discussed in more detail in Chapter 4.

4. *An efficient knowledge representation formalism.*

   The knowledge representation chosen to store the data and inferences about a document greatly influences the ease and efficiency with which the inferencing mechanism manipulates the data. As we have seen, documents have a hierarchical format, both in their physical layouts as well as their logical structures. Thus, a tree is a logical choice to represent the different components of a document. Now, each component of a document, i.e., a block or a unit, has various characteristics, or properties, that fully describe it (e.g., block coordinates, type, etc.). Thus, an ideal representation for storing knowledge about each component would be a "frame", which is a knowledge representation structure (first proposed by Minsky [45]) that contains several attribute-value "slots" to store information about the associated properties of a concept. The knowledge representation chosen here is a generalized "frame-tree", i.e., a tree of frames wherein each block in the document is represented as a frame structure (with slots for storing the values of the various properties associated with the block), and the grouping of the different blocks is represented by a tree with the block frames as individual nodes, and the ordering of text blocks within each unit is represented by pointers to the appropriate associated blocks. The frame tree can actually be constructed incrementally as the classification, grouping, and ordering processes proceed, by means of pointers that define the tree branches. This knowledge representation scheme is described in more detail in Chapter 4.

5. *A methodology for testing the effectiveness of the overall process.*

   An accurate measure of the effectiveness and efficiency of a model lies in the results obtained by testing the model on a variety of subjects. Thus, the measure of this computational model lies in how well it performs in accurately deriving the logical

structure of a given set of documents. To obtain this measure, we have assembled a representative "training" sample of digitized newspaper images. We have then run the complete layout analysis and logical derivation process on data from each of these images and determined whether a complete analysis has been performed on each document. The process (in this case primarily the knowledge base and the control strategy) has then been improved where necessary until it can satisfactorily deal with the entire training sample. The overall process has then been thoroughly run on another representative "target" sample of document images. The output from the process has then been run through a "document formatting" program that translates the logical structure information into a LaTeX file that is processed to re-create the original physical document structure. Analysis of the total percentage of correct and incorrect block identifications, unit groupings, and read-order lists (by comparing the input image with the output re-created image) gives us a measure of the effectiveness of the process. This is discussed in detail in Chapter 7.

3.3 Control Issues in Logical Structure Derivation

As mentioned above, the control strategy for the document logical structure derivation process uses a combination of top-down and bottom-up techniques. The initial image processing operations which act on the raw document image to perform segmentation and layout analysis are data-driven, bottom-up processes. These operations are performed sequentially to prepare input data for the logical derivation process. The control structure for the logical derivation process itself uses both top-down and bottom-up strategies at different stages, as explained below.

The overall strategy briefly consists of the concurrent application of classification rules

and grouping rules on the blocks in the image, alternating between the two depending on the data available to the process at any given time and the status of incomplete inferences made by the process. The classification and grouping operations are invoked by the status of certain control parameters, and to that extent act in a manner similar to *knowledge sources* in a *blackboard control system* (Hayes-Roth [23]). In fact, an earlier version of my approach (Niyogi & Srihari [57]) proposed the use of independent opportunistic knowledge sources to perform the classification and grouping operations, with intermediate data being stored in a global blackboard. Subsequently, upon further refining the overall control strategy, I discovered that the control processes governing the classification, grouping and read-ordering operations did not in fact have to be opportunistic but rather could be controlled under an unified strategy (since the processes are primarily goal-driven rather than purely data-driven). Thus, in the current model, the classification and grouping processes are not opportunistic and are also not strictly data-driven. As will be seen later, certain "meta-rules" in the control structure use top-down, hypothesize-and-test strategies to arrive at block identifications and groupings.

Once the block classification and grouping processes are complete and logical units are completely formed, the read-order determination rules are applied to each unit identified in the document. The information obtained from read-order determination is then incorporated into the logical structure of the document.

Segments of pseudo-code for the block classification and logical block grouping processes should help to illustrate the overall control strategy (Figure 3.4 and Figure 3.5). The determination of read-order is performed after the block classification and grouping processes have been completed and the units have been completely identified. The control strategy for the read-order determination is shown in Figure 3.6.

3.3.1 Block Classification

The pseudo-code for the block classification process (Figure 3.4) shows that all blocks which can be classified independently or in relation to an already classified block are selected first, and classification of such blocks is attempted. Thus, the process avoids getting into a mutual-dependency "deadlock" where two blocks which depend on one another for their classification are selected and produce an infinite loop within the process. If classification of a block is not successful for any reason (such as lack of sufficient data), then the partial results that have been obtained are stored in the global data structure for later retrieval. Note that partial results are not stored in the block frames, but rather in a separate part of the global data structure for further processing.

```
BEGIN

   REPEAT
      SELECT a block;
      WHILE <classification of selected block depends on another block>
        AND <that block is not already classified> DO
                SELECT another block;
      APPLY classification rules to the block;
      IF <classification is successful>
        THEN
             STORE successful classification results
        ELSE
             STORE the partial classification results;
   UNTIL <no more unselected blocks remain>;

END.
```

Figure 3.4: Control strategy for block classification.

3.3.2 Block Grouping

The pseudo-code for the block grouping process (Figure 3.5) shows that grouping is anchored around a major or minor headline. The implicit assumption is that every unit must contain a major or minor headline, and by our definition of an unit (Figure 3.1) we know that this is so (except for "special units" that contain only graphical areas or "piction" blocks). Again, if any grouping attempts are unsuccessful due to the lack of sufficient data, the partial grouping results are stored in the global data structure. The second part of the pseudo-code for block grouping shows that all unclassified and/or ungrouped blocks are then subjected to a set of "special unit" rules for classification and grouping, until no unclassified or ungrouped blocks remain. The whole process is then considered complete.

As seen from the above pseudo-code, the classification and grouping processes are not independent of one another. Some classification rules for a block may depend on knowing the identity of a neighboring block. In such cases, if the neighboring block is not yet classified, the partial classification results are stored in the global data structure, so that at a later time the block can be fully classified with the availability of more data. Similarly, the grouping of some blocks cannot proceed until all the blocks in question are completely classified. In such cases, classification rules get priority over grouping rules, and the partial data is retrieved from the global data structure and processed, until all the blocks in question have been classified, and grouping of these blocks can proceed. Thus, the control strategy requires interaction between the classification process and the grouping process, through the exchange of partial data in the global data structure.

```
BEGIN

   SELECT a classified block;
   REPEAT
      WHILE <block is not a major headline> AND
            <block is not a minor headline> DO
                SELECT another classified block;
      FIND all the neighboring blocks;
      APPLY grouping rules to the blocks;
      IF <complete logical unit is formed> THEN
              STORE the successful grouping results
         ELSE
              STORE the partial grouping results;
   UNTIL <no more unselected blocks remain>;

   REPEAT
      RETRIEVE partial classification results, if any;
      APPLY classification rules for special units;
      RETRIEVE partial grouping results, if any;
      APPLY grouping rules for special units;
   UNTIL <no more partially grouped units remain>;

END.
```

Figure 3.5: Control strategy for logical block grouping.

3.3.3 Determination of Read-Order for Text Blocks

The pseudo-code for the strategy used in determining the read-order for text blocks within each unit is illustrated in Figure 3.6. As mentioned above, read-order determination can only be done after the classification and grouping operations are complete and all the blocks within each unit have been identified.

```
BEGIN

   REPEAT
      SELECT a unit;
      WHILE <unit does not contain any text blocks> OR
            <unit is already read-ordered> DO
               SELECT another unit;
      APPLY read-order rules to the unit;
      MARK unit as read-ordered;
   UNTIL <no more unselected units remain>;

END.
```

Figure 3.6: Control strategy for read-order determination.

The most basic rule for determining read-order is that the normal read-order of text blocks within a unit is from top to bottom and from left to right. Thus, for example, for a multi-column unit containing three text blocks in the first column, one text block in the second column, and two text blocks in the third column (in addition to other non-text blocks), the read-order might be: Column 1 Text-block 1, Column 1 Text-block 2, Column 1 Text-block 3, Column 2 Text-block 1, Column 3 Text-block 1, Column 3 Text-block 2. (This is illustrated in Figure 3.7, in which the dotted arrows indicate the read-order). There are

other constraints that have to be obeyed for read-order determination too, such as spatial relations between the text blocks and the non-text blocks in a unit, and the presence of embedded units. These constraints are encoded in the knowledge base of read-order rules.

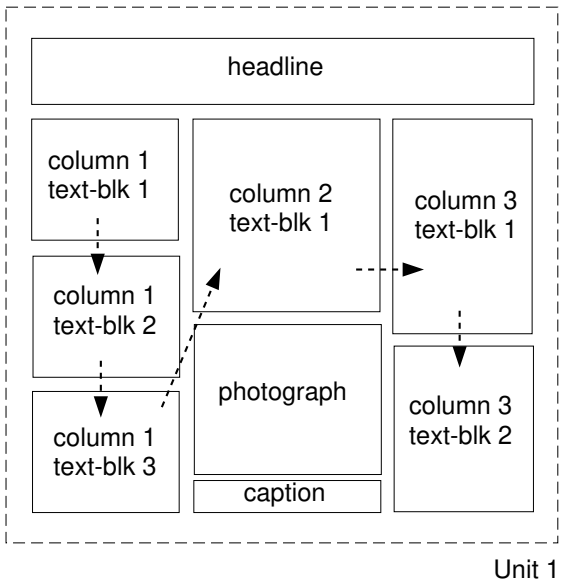


Figure 3.7: Example of read-order of text blocks within a unit.

Chapter 4

DeLoS: A System For Deriving Logical Structure

4.1 Development of a Logical Structure Derivation System

We have developed and implemented a knowledge-based system for the derivation of document logical structure based on the computational model described in Chapter 3. This system, called DeLoS ("DErivation of LOgical Structure") takes as input the digitized image of a newspaper page and produces as output a symbolic description of the logical structure of the page, including a classification of each of the printed blocks on the page, a grouping of these blocks into logical units, and the reading order of text blocks within each of these units. Various image processing operations are performed to extract basic properties of the components of the image, and this data is analyzed under the control of a rule-based system, which operates in conjunction with a global data structure to monitor the entire classification, grouping, and block-ordering process. Preliminary research on the design and development of this system has been described in previously published papers (Niyogi & Srihari [55], [57], [56], [58] & [54]).

4.2 System Organization

Figure 4.1 shows the components of the DeLoS system. The unbroken arrows between the boxes indicate the flow of data between the various components of the system, and the dotted arrows represent the control flow in the system.

As seen in Figure 4.1, the DeLoS system consists of a multi-level, rule-based reasoning system, an image pre-processing sub-system, and a partitioned global data structure. The rule-based system utilizes a top-down, backward-chaining structure. An inference engine within the rule-based system makes deductions about the document using a hierarchical knowledge base that contains rules describing all the identifiable characteristics of document images. The rules are classified into three levels: knowledge rules, control rules, and strategy rules.

The global data structure facilitates the transfer of information between the image processing modules and the rule-based system. A common data area stores all intermediate computation results and other control information. The global data structure utilized in this system is similar to a standard blackboard (Nii [52] & [53]) to the extent that it is a global data area that is accessed by different sub-processes within the system and is used to store and modify incremental data and results. It is, however, different from a standard blackboard to the extent that the sub-processes that access and modify the data are not independent opportunistic "knowledge sources", but are rather sets of rules that are activated within the system's rule-invocation structure.

The image-processing modules directly access the document image to extract various kinds of information about the document. Intrinsic properties of the different printed blocks as well as the spatial relationships between the different blocks constitute the information

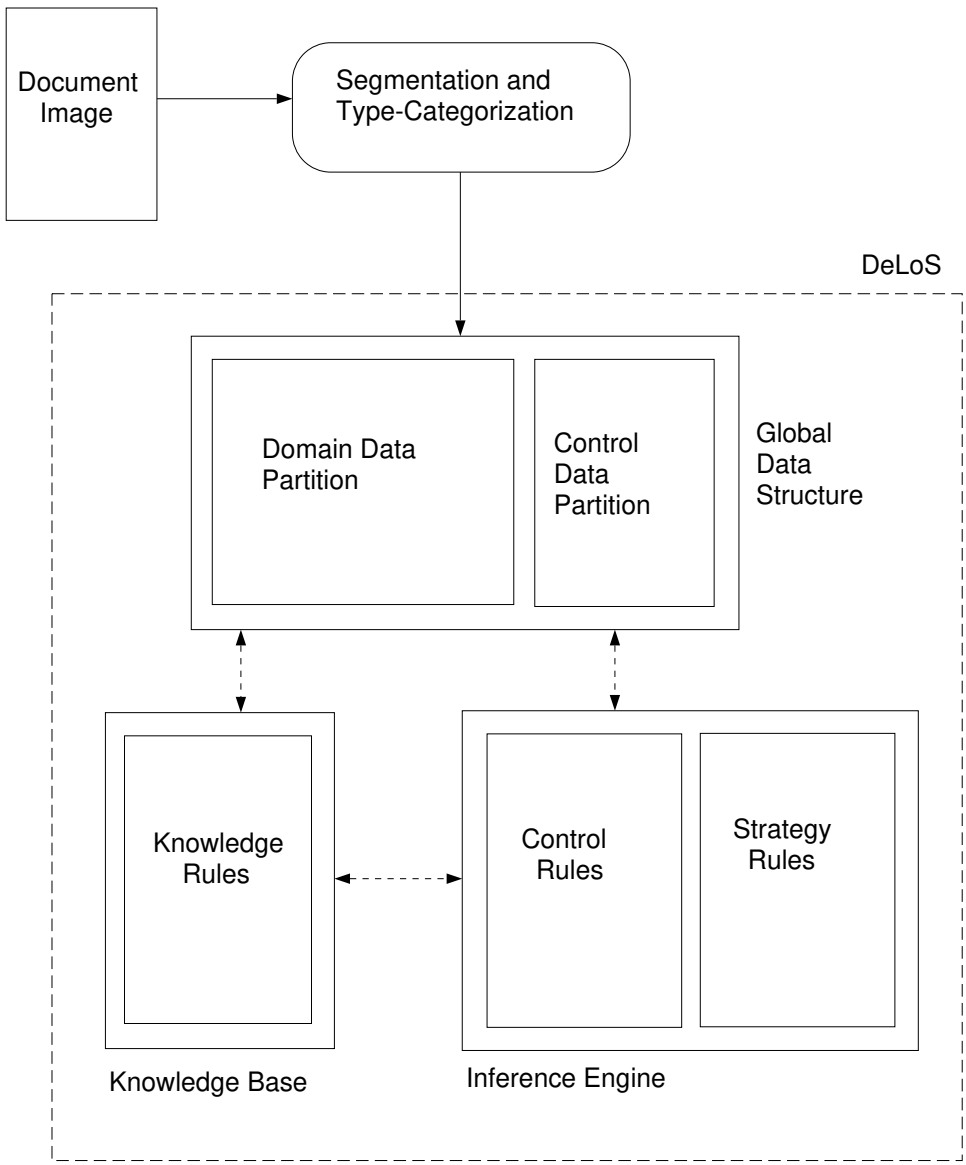


Figure 4.1: The DeLoS logical structure derivation system.

that is passed back to the control structure through the global data structure. Some image-processing modules perform the basic tasks of binarization, connected-component analysis, etc., so as to prepare the image for other modules (e.g., block segmentation) that can extract information from it.

As mentioned above, in the logical-structure derivation system, rule-based control is integrated with a global data structure to achieve maximum flexibility. We will now discuss the actual design details of the system.

4.3 Rule-Based Control Structure

The rule-based system consists of three levels of rules. The three-level rule structure for this system was inspired by Nazif and Levine's work [49] on low-level image segmentation of natural scenes using a rule-based system. It was demonstrated there that using a hierarchical structure of three progressively abstract levels of rules provided a large amount of flexibility in the inference mechanism, and allowed a modular formulation of the solution within the image analysis problem domain. The three levels into which the rules in this system are classified are as follows:

- Knowledge Rules (level 1)
- Control Rules (level 2)
- Strategy Rules (level 3)

The domain knowledge in the system is maintained in the Knowledge rules, and the control decisions in the system are made by the Control and Strategy rules, as described below.

4.3.1 The Knowledge Base

The Knowledge rules comprise the knowledge base that contains all the domain knowledge for the system, expressed in terms of first-order predicates. These rules define the general characteristics expected of the usual components of a document image and the usual relationships between such components in the image. Thus, all common characteristics of different types of document blocks (e.g., text blocks, photographs, etc.), as well as spatial constraints commonly followed in document layout (e.g., the positioning of captions relative to photographs, etc.), are encoded into the knowledge base. These knowledge rules in the knowledge base contain all the properties and spatial relations for different types of document blocks, and can be used for block classification, block grouping, or text block ordering as and when required according to the control strategy.

Knowledge rules can be further categorized into the following:

- *Unary rules:* These rules allow for the classification of a block based solely upon its properties; e.g., a thin graphical block in a newspaper image can be classified as a row separator or a column separator based solely upon its size and aspect ratio.

- *Simple binary rules:* These rules depend on the positions of adjacent blocks for classification; e.g., a half-tone block in a newspaper image can be classified as a page banner based upon not only its size and aspect ratio, but also its position relative to its neighboring blocks.

- *Complex binary rules:* These rules depend on the identities of adjacent blocks for classification; e.g., a small-text block in a newspaper image can be classified as a caption based not only on its own size and aspect ratio, but also the properties and

identities of its neighboring blocks (i.e., whether it is adjacent to a photograph block and meets certain spatial constraints).

The hierarchical structure of the knowledge base mentioned above is thus created by these different categories of rules.

Some examples of knowledge rules are shown in Figure 4.2.

```
IF a block X is of type ‘‘small-text’’,
  AND IF it is below another block Y which is a photograph,
  AND IF the widths of the two blocks are equal,
THEN block X is a photo-caption.


IF a block Z is of type ‘‘large-text’’,
  OR IF it satisfies the following three conditions:
      {it is of type ‘‘medium-text’’,
         AND it is below another block W,
         AND block W is not of type ‘‘large-text’’ or ‘‘medium-text’’},
THEN block Z is a major headline.
```

Figure 4.2: Examples of knowledge rules in the rule-based system.

4.3.2 The Inference Engine

The control structure for the rule-based system contains an inference engine which is also rule-based, and contains two levels of rules: *control rules* and *strategy rules*. These rules regulate the analysis of the document image, and decide when a consistent interpretation of the image has been obtained.

The rules comprising the inference engine are also formulated in terms of first-order

predicates. The control structure determines the order in which these rules are executed in order to test various conditions effectively. Control rules regulate the invocation of the knowledge rules, based on appropriate data configurations or processing states. Control rules can be further categorized as follows:

- *Focus-of-attention Rules:* Focus-of-attention rules are designed to recognize data configurations that indicate the blocks on which to apply knowledge rules next, e.g., a rule that decides that the identity of the block directly above the current block should determined next.

- *Meta-Rules:* Meta-rules, on the other hand, are designed to recognize processing states and data configurations that indicate what category of knowledge rules are to be executed next as the highest priority, e.g., a rule that decides that all the relevant knowledge rules for captions should be executed next so as to determine whether the current block is a caption.

Examples of control rules are shown in Figure 4.3.

Strategy rules guide the search in a more general way, i.e., they determine what control strategy is to be followed at any given time for analyzing the image. This means that the strategy rules regulate the invocation of, and determine the execution order of, the control rules. Strategy rules also decide on the stopping criteria for the system, i.e., whether a consistent interpretation and grouping of the blocks in the document image has been achieved (as determined by the absence of incomplete block or unit data in the global data structure, and by the completeness of the logical structure tree). The system control strategy algorithms shown in Chapter 3 (Figure 3.4, Figure 3.5, and Figure 3.6) are thus directly translated into a set of strategy rules. Therefore, there is a set of strategy rules for

SYSTEM

```
[META-RULE]:

IF there are partial classification results for any blocks,
THEN select one (partially classified) block
  AND apply all classification rules for the selected block.


[FOCUS-OF-ATTENTION RULE]:

IF the grouping mode is on,
  AND a block has been selected,
THEN find all the immediate neighbors of the selected block.
```

Figure 4.3: Examples of control rules in the rule-based system.

block classification, another set for block grouping, and yet another for text block ordering.

Examples of strategy rules are shown in Figure 4.4.

4.4 The Global Data Structure

The global data structure stores the physical structure and logical structure information for the document being processed. It also facilitates the transfer of information between the rule-based system and the image processing modules. A common data area stores all intermediate computation results and other control information, and provides the framework for the construction of the trees representing the document structures.

In this system, the global data structure is divided into the following partitions:

- *Domain Data Partition:* This is the major part of the data structure, and contains

```
IF classification of the selected block is not successful,
THEN store the partial results in the global data structure
  AND modify all appropriate indicators.


IF any partially grouped units remain,
THEN apply all unit-related control rules for each of these units
       until there are no more partial units.
```

Figure 4.4: Examples of strategy rules in the rule-based system.

the data and inferences about the image blocks and units. Thus, this is the partition in which the frame-trees representing the physical and logical structures are formed and modified.

- *Control Data Partition:* This contains information about the status of the classification, grouping and read-ordering process as well as details about any partial data or results being maintained in the global data structure.

The Domain Data Partition contains the domain data (i.e., data and inferences about the image blocks and units) that is transferred by the image processing routines to the rule-based inference engine. It also contains the physical and logical structures, which are organized as a set of block frames and unit frames in a hierarchical form.

The Control Data Partition in this system is logically separate from the domain data partition, and contains flags, switches, and other variables which indicate exactly what data currently exists in the Domain Data Partition, what part of the document image was processed to obtain that data, what operations were performed, and with what parameters.

SYSTEM

It also contains information about the status of the overall classification and grouping process, and also details about any partial data or results that are being maintained in the global data structure.

4.4.1 Representation of Domain Data

As mentioned above, the domain-data partition contains data that has been extracted from the document image by the various image-processing routines. A common representation for this data is essential for the purpose of access and modification by the different components of the system. The representation chosen must reflect the original structure of the document and at the same time be flexible enough for constant modification, since data is obtained at various times during the block classification, grouping, and ordering process.

The input image data is represented initially as a list of blocks with their physical properties and their type. The physical structure tree is created from this list of blocks. Each block in the image data is represented by a frame. A frame is an attribute-value data structure that consists of a number of attribute "slots", to each of which one can associate a value. Thus, each attribute slot in a frame represents a particular characteristic of a block or unit. In our system, no slot can contain a default value; each value maintained in a frame must be explicitly placed there.

There are two kinds of frames:

1. *Block frames:* These frames represent physical characteristics of document blocks and are thus structural in nature.

2. *Unit frames:* These frames are more conceptual in nature, and represent the logical groupings of the different document blocks.

SYSTEM

Each block frame contains data about one block in the image. Different characteristics of a block are represented as slots in a block frame. The different slots in the current DeLoS system are shown in Figure 4.5 (a).

```
(1) Block number
(2) Height
(3) Width
(4) Position (X--Y coordinates of top-left and bottom-right corners)
(5) Block Type (large-text/medium-text/small-text/photo/line drawing)
(6) Block Label
            [filled in by the system after classification]
(7) Unit to which it belongs
            [filled in by the system after grouping]
(8) Block numbers of the previous and next blocks in text reading order
            [filled in by the system after read-order determination]
```

(a) Block Frame

```
(1) Unit number
(2) Number of blocks contained in the unit
            [filled in and updated during grouping]
(3) Ordered list of block numbers
            [filled in during grouping, ordered during read-ordering]
```

(b) Unit Frame

Figure 4.5: Slots in block and unit frames.

Unit frames represent logical groupings of the blocks in the image (e.g., a newspaper story, consisting of a set of text paragraphs, a photograph and its caption, and a headline). Information in a unit frame can only be filled in after the classification of all the blocks in

the document image has been made and the blocks have been grouped into logical units. The slots in a unit frame are shown in Figure 4.5 (b).

Unit frames and block frames, when the slots are all filled in, make up a tree structure of frames, since each unit is a parent to several blocks or other sub-units. It is possible to put a unit number into the "list of block numbers" slot of a unit frame, which makes that unit a child unit of the current one. Thus, a tree structure is created, whose root node is the unit frame representing the entire document page, and all other nodes are either unit frames representing logical subdivisions of the page, or block frames representing physical blocks of printed matter in the page. The above representation therefore allows us to specify logical sub-units of a given unit. For example, a given story on a particular topic with one major headline may have sub-stories under separate minor headlines. The system, by allowing unit frames to be child nodes of other unit frames, allows us to efficiently represent such a hierarchy among the stories on a document page.

Chapter 5

Development of the Knowledge Base in DeLoS

The knowledge base in the DeLoS system contains rules that encapsulate the knowledge about document structure as well as block and unit properties for documents. Now, any document image understanding system should have two main goals, namely, accuracy and flexibility. A system can achieve a high degree of accuracy in processing a particular class of documents by including in its knowledge base as much publication-specific knowledge about that document class as possible. On the other hand, a system can achieve a high degree of flexibility by including in its knowledge base as much knowledge about many different classes of documents as possible. However, if both these requirements are given high priority, the result is a knowledge base that is too big and unwieldy to achieve a satisfactory degree of computational efficiency. Therefore, we need to balance the two requirements and decide our priority in terms of whether it is more desirable to be able to accurately process a particular class of documents, or whether it is more desirable to be able to process a wide variety of documents. Thus the choice is between depth of knowledge versus breadth of knowledge.

KNOWLEDGE

The decision that has been made in the case of the DeLoS system is to concentrate on a specific class of documents, namely, newspapers, and achieve as high a degree of accuracy as possible. The reasoning behind this decision is that with a modular design for the system, an equivalent set of publication-specific rules for another document class can be substituted for the current set, and thus a comparably accurate system can be created that can process any document from that document class.

To ensure the above-mentioned modularity in the system, the knowledge base is conceptually divided into two parts:

1. *Domain, or Publication-specific, Knowledge*, i.e., knowledge that applies to a specific class of documents.

2. *Document World Knowledge*, i.e., knowledge that applies to a wide variety of documents.

As mentioned in Chapter 1, there are certain properties common to most classes of documents and some others that are specific only to a particular class of documents. Thus, the two conceptual parts of the knowledge base codify these different types of knowledge.

5.1 Domain Knowledge

Domain knowledge is that knowledge which describes specific characteristics of different classes of documents. Thus, different kinds of domain knowledge exists for newspapers, journals, magazines, forms, office documents, etc.

In the DeLoS system, the domain knowledge used to derive the logical structure of a document is encoded into knowledge rules. The domain that the DeLoS system deals with

is that of newspaper images. As mentioned in Chapter 3, newspapers conform to certain fixed formats, but even within the general class of documents consisting of all newspapers, there are variations in syntactic and semantic styles. Therefore, we have to encode in the knowledge base the domain knowledge about newspaper layout, as well as publication-specific knowledge, if any, about each type of newspaper that we want to process. In the DeLoS system, all publication-specific knowledge about *The Buffalo News* and *USA Today* have been encoded into knowledge rules in the knowledge base.

Newspaper structure is somewhat different from other documents, and newspaper pages have some unique characteristics. For example, logical units that are completely enclosed by rectangular boxes are considered to be not connected with adjoining stories. Also, every photograph in a newspaper has a caption, and in some newspapers, every story has a byline (i.e., the name and affiliation of the author of the story). In newspapers, vertical lines separate stories, and horizontal lines can separate columns and/or stories. Within a story, major headlines (i.e., large-text titles) always appear above minor headlines (i.e., medium-text titles) when both are present.

To illustrate the above, Figure 5.1 shows two rules in the DeLoS system that use domain knowledge about story bylines in newspaper articles. The first rule classifies a story byline by its position with respect to other adjacent blocks. The second rule groups the story byline with its appropriate adjacent blocks.

5.2 Document World Knowledge

Document world knowledge is that knowledge which is required for the analysis of a document but which does not specifically describe a particular document being analyzed. This

```
IF block B1 is a headline,
    AND IF block B2 is below B1,
    AND IF B2 is a text block,
    AND IF block B3 is below B2,
    AND IF B3 is a horizontal line,
    AND IF block B4 is below B3,
    AND IF B4 is a text block,
THEN block B2 is a story byline.
```

(i) Classification Rule

```
IF block B1 is a headline,
    AND IF block B2 is below B1,
    AND IF block B2 is a story byline,
    AND IF block B3 is below B2,
    AND IF block B3 is a horizontal line,
    AND IF block B4 is below B3,
    AND IF block B4 is a text block,
THEN blocks B1, B2, B3 and B4 belong to the same unit.
```

(ii) Grouping Rule

Figure 5.1: Rules using domain knowledge.

KNOWLEDGE

includes knowledge about general image characteristics common to all classes of documents, as well as information that is required for model-based analysis of any type of document image.

In the DeLoS system, document world knowledge is encoded into knowledge rules in the knowledge base, as well as incorporated into the control structure in terms of control and strategy rules. This knowledge includes details of the general properties of documents, as well as strategies for traversal of blocks within a document.

Most documents have certain common characteristics. For example, we can assume that in most classes of documents, text blocks belonging to an article or story will always be below, and never above, the title or headline describing the article or story. Also, captions are always next to the photographs that they describe (and usually directly below them). Text blocks normally do not overlap each other. All text lines are assumed to be horizontal.

To illustrate the above, Figure 5.2 shows two rules in the DeLoS system that use world knowledge about document structure. The first one indicates that if a vertical line is present between two other blocks, one of which is a headline, then those two blocks belong to different units. The second one is the simplest read-ordering rule for text blocks (i.e., read-order is top-to-bottom for text blocks within the same column).

Discussion

As seen above, it is possible to construct a knowledge base that accurately describes the structure of a specific type of document. This knowledge base contains both publication-specific domain knowledge as well as general world knowledge about documents. The accuracy of a system that uses such a knowledge base is dependant on the accuracy of the

KNOWLEDGE

```
IF block B1 is a headline,
    AND IF block B2 is right-of B1,
    AND IF B2 is a vertical line,
    AND IF block B3 is right-of B2,
THEN blocks B1 and B3 do not belong to the same unit.
```

(i) Grouping Rule

```
IF block B1 is a text block,
    AND IF block B2 is a text block,
    AND IF B2 is a neighbor of B1,
    AND IF B2 is below B1,
    AND IF B1 and B2 have equal width,
THEN B2 is next in read-order after B1.
```

(ii) Read-ordering Rule

Figure 5.2: Rules using document world knowledge.

domain knowledge, and also the modularity of the world knowledge to enable substitution of domain knowledge that describes other classes of documents.

Currently the domain knowledge contained in the DeLoS knowledge base is relevant to newspapers. Therefore, when the DeLoS system is run on a different type of document (such as a journal page), the results are not as accurate. Although the basic block categories in journal pages are the same as those in newspapers, the block and unit labels are somewhat different, and so is the physical structure. Substitution of the domain knowledge to reflect the different structure of journal pages would enable the efficient logical structure derivation of journals and other classes of documents.

Appendix A lists all the rules that comprise the domain knowledge in the DeLoS knowledge base.

Chapter 6

Implementation of the DeLoS System

This chapter describes the implementation of the DeLoS system, and includes details about the specific inputs and outputs of the system, as well as the image processing operations that are performed on the original scanned document image to prepare it for further analysis.

6.1 System Implementation

The document logical structure derivation system (DeLoS) described in this dissertation has been implemented on a Sun SPARC-2 workstation running the SunOS 4.1.3 operating system. Document images are scanned using a flatbed scanner at a resolution of 300 pixels per inch, and the resulting bitmap of the document in Sun raster format is converted into HIPS format (Landy et al. [33]) prior to any further processing.

Originally, images of newspaper pages were created by image capture by means of a digitizing Eikonix camera. The advantage of this technique was that complete broadsheet newspaper pages (i.e., 24 x 14 inches) could be digitized. The disadvantages were twofold. First, a lot of time and effort was required to properly illuminate the original newspaper page

so as to get uniform illumination on all parts of the page. Second, the intense illumination produced a "bleedthrough" effect on the page, such that the printed matter on the reverse side of the page was lightly visible through the page, thus producing a noisy gray-scale image that was difficult to segment accurately.

Using a flatbed scanner for digital image capture has two advantages over the digitizing camera. First, the illumination problem is eliminated, since there are no lights that have to be manually adjusted for each document. Second, the flatbed removes the bleedthrough effect and produces a much less noisy image. However, the largest scanner available for image capture for this system has a flatbed with dimensions of 17 inches across and about 12 inches deep. The largest page that can be scanned using this scanner is 17 inches tall and 11.7 inches wide. Thus, complete broadsheet pages cannot be scanned using this scanner.

The solution to this problem has been to scan the largest logically complete partial newspaper page (i.e., no partial blocks) that the flatbed can accommodate. Therefore, all the newspaper images of *USA Today* (scanned at 200 ppi) as well as *The Buffalo News* used to test this system are images of partial pages (the training set of *The Buffalo News* images used are the gray-scale complete page images mentioned above). Using such a partial page does not, however, significantly affect the processing of these pages, since a complete logical structure can be derived from the logically complete partial page.

Because of the propensity of different languages to allow efficient representation and manipulation of certain types of information, sub-systems in DeLoS are implemented in different languages. The rule-based control structure, being primarily a top-down backward-chaining system, is implemented in C-Prolog, which is a version of the Prolog logic programming language (Clocksin & Mellish [9]). The knowledge base, consisting primarily of first-order predicates, lends itself perfectly to the predicate nature of Prolog assertions and

rules, and is therefore also implemented in that language. The global data structure, and the routines enabling its interaction with the rule-based system, have also been implemented in Prolog. The image processing routines, which provide the inputs to the system and which require efficient and fast interaction with the actual image data, are implemented in C. The interaction between these sub-systems is handled via system routines which perform tasks such as translation of formats, consolidation of image data, and document-structure reconstruction. Because of the modularity of the system design, this system interface is implemented in a fairly efficient manner in Prolog.

The development of this system was done in the following logical sequence: First, the rule-based control system was developed and equipped with a broad set of knowledge rules that described the characteristics of newspapers. This allowed the system to accept data about different blocks on a newspaper image, classify them into appropriate block identities, and then group the blocks into self-contained logical units. The read-order determination strategy was then added to the system. The inference engine contained basic control rules and strategy rules, and was then expanded to allow finer discrimination of objects via a wider variety of rules. Next, the structural framework of the global data structure was designed and implemented. Finally, the block categorization routines were developed to extract block information from the segmentation results and process them. Also, document reconstruction routines were developed to re-create the physical structure of the document, so that the accuracy of the DeLoS output could be verified.

The acceptable codes for block types (block categories) are given in Figure 6.1 (a). The acceptable codes for block labels (block classifications) for newspapers are given in Figure 6.1 (b). As can be seen from the list, the first digit of each block label refers to its block type, thus making it easier for the system to keep track of which block types can be

legitimately translated into which block labels.

```
1 - small text,     2 - medium text,      3 - large text,
4 - half-tone,      5 - line-drawing,     6 - other,
7 - vertical line,  8 - horizontal line,  9 - line rectangle.
```

(a) Block types

```
11 - text paragraph,    12 - caption,      13 - story byline,
21 - minor headline,
31 - major headline,
41 - photograph,        42 - page banner,  43 - thick horiz. band,
51 - line drawing,
61 - unknown,
71 - column separator,
81 - row separator,
91 - enclosing rectangle.
```

(b) Block labels for newspapers

Figure 6.1: Acceptable block types and block labels for the DeLoS system.

6.2 System Inputs

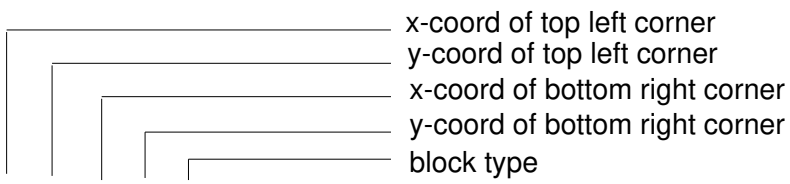
Document pages are scanned in as 300 ppi gray-scale images. After conversion to HIPS format, each image is then binarized, filtered to remove noise, and corrected for skew. Binarization is done using an adaptive thresholding method. (The partial pages mentioned above are scanned in as binary images, and are then converted to HIPS format and bit-unpacked.) Next, connected component analysis is done on the binary image. Segmentation is then performed, and text/graphics discrimination is performed. From the block-segmentation information, a block pre-processing program performs block-type categorization. The out-

put of the block-type categorization program consists of a list of the blocks in the document showing their coordinates and type. This block list is the primary input to the system. Figure 6.2 shows a sample input for the DeLoS system.

6.3 System Outputs

The primary output from this knowledge-based system is a data representation of the logical structure of the document, i.e., a tree showing all the classified blocks in the document image, giving all the relevant extracted feature details for each block, and an indication of the logical unit to which the particular block belongs, as well as an ordered listing of all the text blocks in each unit. This data structure is output in the form of an editable text file that contains information about the identity, position, and other relevant properties of all the logical blocks in the image, as well as the logical groupings of these blocks into logical units. In addition to this, the output also contains pointers between blocks to represent the reading order of the text blocks within each unit. Figure 6.3 shows a sample output file from the DeLoS system.

The data structure showing the different classified blocks and their logical groupings in a document is eminently suitable for transmission to another location, where the document can be reconstructed from its description. Also, such a data structure can be used by a higher-level interactive system that accepts user queries and attempts a semantic interpretation of the document, including text reading and natural language understanding.




```
156 18 1941 153 3
153 183 384 249 2
153 264 378 342 2
156 351 417 420 2
153 450 318 492 1
153 531 486 2034 1
516 189 1599 960 4
522 963 1596 1029 1
519 1056 1230 1596 4
522 1593 1227 1665 1
525 1692 858 2031 1
894 1695 1230 2034 1
1263 1059 1596 2031 1
1632 189 1893 264 2
1638 282 1956 348 2
1632 360 1941 420 2
1632 456 1833 501 1
1635 537 1965 2037 1
1632 2067 1965 2391 4
1629 2415 1965 2697 1
153 2061 1569 2154 2
150 2166 336 2208 1
153 2247 489 2691 1
519 2166 855 2442 4
522 2463 855 2691 1
894 2175 1230 2694 1
1260 2172 1599 2697 1
```

Figure 6.2: Sample input for DeLoS.

```
block(1,[dim(1785,135),pos(156,18),type(3),blk_lbl(31)]).
block(1,[prevtxt(0),nexttxt(0),unit_lbl(1)]).
block(2,[dim(231,66),pos(153,183),type(2),blk_lbl(21)]).
block(2,[prevtxt(0),nexttxt(0),unit_lbl(1)]).
block(3,[dim(225,78),pos(153,264),type(2),blk_lbl(21)]).
block(3,[prevtxt(0),nexttxt(0),unit_lbl(1)]).
block(4,[dim(261,69),pos(156,351),type(2),blk_lbl(21)]).
block(4,[prevtxt(0),nexttxt(0),unit_lbl(1)]).
block(5,[dim(165,42),pos(153,450),type(1),blk_lbl(13)]).
block(5,[prevtxt(0),nexttxt(0),unit_lbl(1)]).
block(6,[dim(333,1503),pos(153,531),type(1),blk_lbl(11)]).
block(6,[prevtxt(99),nexttxt(11),unit_lbl(1)]).
block(7,[dim(1083,771),pos(516,189),type(4),blk_lbl(41)]).
block(7,[prevtxt(0),nexttxt(0),unit_lbl(1)]).
block(8,[dim(1074,66),pos(522,963),type(1),blk_lbl(12)]).
block(8,[prevtxt(0),nexttxt(0),unit_lbl(1)]).
block(9,[dim(711,540),pos(519,1056),type(4),blk_lbl(41)]).
block(9,[prevtxt(0),nexttxt(0),unit_lbl(1)]).
block(10,[dim(705,72),pos(522,1593),type(1),blk_lbl(11)]).
block(10,[prevtxt(0),nexttxt(0),unit_lbl(1)]).
block(11,[dim(333,339),pos(525,1692),type(1),blk_lbl(11)]).
block(11,[prevtxt(6),nexttxt(12),unit_lbl(1)]).
block(12,[dim(336,339),pos(894,1695),type(1),blk_lbl(11)]).
block(12,[prevtxt(11),nexttxt(13),unit_lbl(1)]).
block(13,[dim(333,972),pos(1263,1059),type(1),blk_lbl(11)]).
block(13,[prevtxt(12),nexttxt(99),unit_lbl(1)]).
block(14,[dim(261,75),pos(1632,189),type(2),blk_lbl(21)]).
block(14,[prevtxt(0),nexttxt(0),unit_lbl(2)]).
block(15,[dim(318,66),pos(1638,282),type(2),blk_lbl(21)]).
block(15,[prevtxt(0),nexttxt(0),unit_lbl(2)]).
block(16,[dim(309,60),pos(1632,360),type(2),blk_lbl(21)]).
block(16,[prevtxt(0),nexttxt(0),unit_lbl(2)]).
block(17,[dim(201,45),pos(1632,456),type(1),blk_lbl(13)]).
block(17,[prevtxt(0),nexttxt(0),unit_lbl(2)]).
block(18,[dim(330,1500),pos(1635,537),type(1),blk_lbl(11)]).
block(18,[prevtxt(99),nexttxt(20),unit_lbl(2)]).
block(19,[dim(333,324),pos(1632,2067),type(4),blk_lbl(41)]).
block(19,[prevtxt(0),nexttxt(0),unit_lbl(2)]).
block(20,[dim(336,282),pos(1629,2415),type(1),blk_lbl(11)]).
block(20,[prevtxt(18),nexttxt(99),unit_lbl(2)]).
block(21,[dim(1416,93),pos(153,2061),type(2),blk_lbl(31)]).
block(21,[prevtxt(0),nexttxt(0),unit_lbl(3)]).
block(22,[dim(186,42),pos(150,2166),type(1),blk_lbl(13)]).
block(22,[prevtxt(0),nexttxt(0),unit_lbl(3)]).
block(23,[dim(336,444),pos(153,2247),type(1),blk_lbl(11)]).
block(23,[prevtxt(99),nexttxt(25),unit_lbl(3)]).
block(24,[dim(336,276),pos(519,2166),type(4),blk_lbl(41)]).
block(24,[prevtxt(0),nexttxt(0),unit_lbl(3)]).
block(25,[dim(333,228),pos(522,2463),type(1),blk_lbl(11)]).
block(25,[prevtxt(23),nexttxt(26),unit_lbl(3)]).
block(26,[dim(336,519),pos(894,2175),type(1),blk_lbl(11)]).
block(26,[prevtxt(25),nexttxt(27),unit_lbl(3)]).
block(27,[dim(339,525),pos(1260,2172),type(1),blk_lbl(11)]).
block(27,[prevtxt(26),nexttxt(99),unit_lbl(3)]).
unit(1,blocks(13),blocknum([1,2,3,4,5,6,7,8,9,10,11,12,13])).
unit(2,blocks(7),blocknum([14,15,16,17,18,19,20])).
unit(3,blocks(7),blocknum([21,22,23,24,25,26,27])).
```

block dimensions, position, type, and label.

unit to which block belongs + pointers to previous & next blocks in read-order

unit information (number of blocks + list of block numbers)

Figure 6.3: Sample output from DeLoS.

6.4 Image Processing

The image processing modules directly access the document image to extract various types of information about the document, primarily the intrinsic properties of the printed blocks and the block type for each block. This information is stored in the global data structure and used as input data for the rule-based system.

The basic image processing operations performed on the scanned gray-level document image include binarization, connected component analysis, image segmentation, and block categorization. Figure 6.4 shows the steps in the transformation of the original document into the format required for input into the DeLoS system.

The original, gray-level document image obtained by scanning the physical document is first converted from the original Sun raster format to the HIPS format. It is then binarized using an adaptive thresholding algorithm (Liu et al. [38]). Then, connected component analysis is performed on the binary image (DeSilva [13]). Next, segmentation is performed on the image using the connected component data, by a method known as "docstrum" (O'Gorman [60]), which has been implemented locally at CEDAR by a fellow graduate student (DeSilva [13]).

The segmentation technique utilized here deserves further explanation, since it is critical to obtaining accurate block information for the DeLoS system. The "docstrum" method of image segmentation uses K-nearest-neighbor clustering of connected components. For each connected component in the image, K nearest neighbors are found by investigating the neighborhood around it. A neighbor is defined by the distance between the centroids of the component and the neighbor, as well as the angle to the horizontal of the line connecting the centroids. Then, by analyzing these distances and angles for the whole image, values

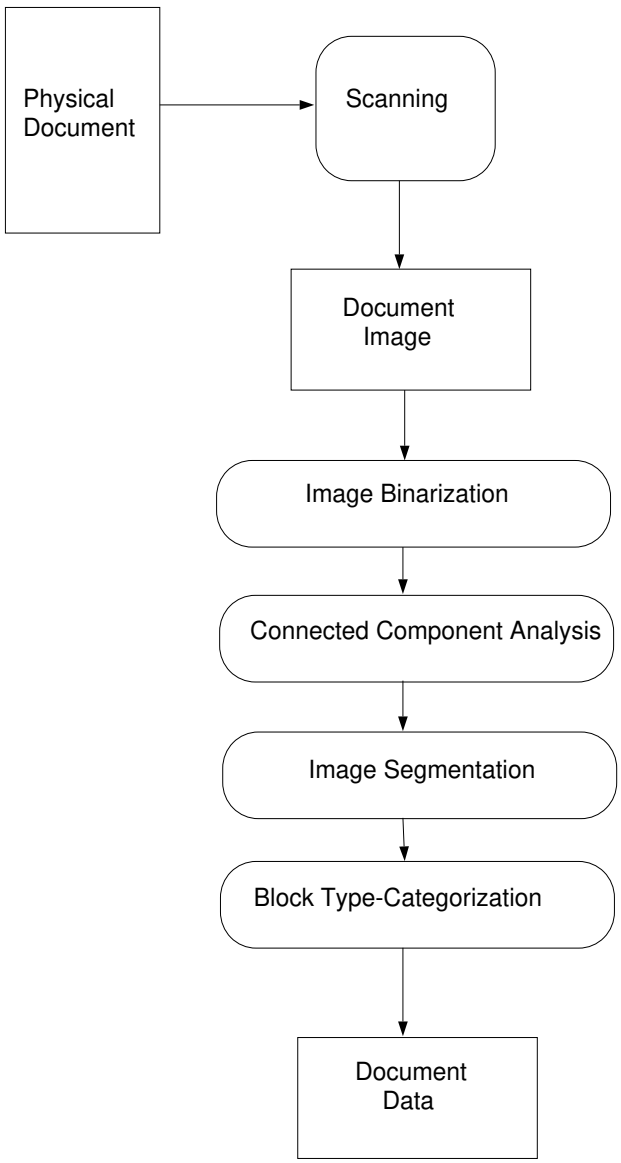


Figure 6.4: Image processing steps for documents.

of the skew, between-text-line spacing, and within-text-line spacing are obtained. These parameters are then used to merge the connected components into text lines, and text lines into text blocks. The main advantages docstrum has over other bottom-up approaches to image segmentation are its relative independence from *a priori* knowledge of layout rules and structure, and the concurrent determination of skew and block layout. This independence of *a priori* knowledge makes it possible for docstrum to operate over a wide range of documents. A limitation is that occurrence of small amount of noise between blocks, unless filtered out, might merge blocks together.

Because the "docstrum" program relies on neighborhood clustering, it has to be run several times, each time removing the identified text blocks from the connected components list. This prevents the large text blocks (e.g., major headlines) from being type-categorized as graphics, which is what happens if "docstrum" is only run once on the entire image. In order to allow a more accurate analysis of the larger components, the connected component size thresholds in "docstrum" are changed appropriately after the removal of the smaller connected components at the end of each run.

The result of segmentation is that each of the printed blocks on the image is isolated and categorized as text or graphics. Further processing of the blocks information, using block size and connected component characteristics, results in the categorization of each block into a block *type*, i.e., "small-text", "medium-text", "large-text", "line-drawing", or "half-tone". The list of all the blocks in the image, giving the coordinates of their enclosing rectangle as well as the basic block type, is the input to the knowledge-based document logical structure derivation system.

An alternative approach to block-type categorization that was originally used by the DeLoS system is the method of texture analysis (Wang & Srihari [78]), because text areas

can be considered to have some common textural characteristics. For this, homogeneous rectangular blocks are first segmented out of the image using methods such as run-length smoothing and recursive horizontal/vertical cuts. The type categorization approach is based on statistical textural features and feature space decision techniques. The feature spaces created by these statistical features are effective in categorizing blocks into a set of predetermined document block categories.

The image processing operations used to obtain input document data for the DeLoS system, including binarization, connected component analysis, and image segmentation, as well as "docstrum", have been developed by other members of the CEDAR research group. The techniques used in these operations are described in detail in Wang & Srihari [78], DeSilva [13], and Liu et al. [38]. However, I have developed and implemented the block type-categorization technique currently being used, specifically to process output data from the "docstrum" program and prepare the appropriate input data for the DeLoS system.

Chapter 7

Experimental Results and Performance Analysis for DeLoS

DeLoS, the system for the derivation of logical structure of documents, has been developed and implemented, and has been tested on a variety of document images. This chapter presents the results that have been obtained using this system, and discusses the performance of the system on different document types.

7.1 Experimental Results

The DeLoS system has been trained using data from page images of *The Buffalo News*, and extensively tested on data from digitized images of pages from *The Buffalo News* and *USA Today* newspapers.

The training set consisted of 14 full page gray-scale images which had been scanned using an Eikonix digitizing camera. The image shown in Figure 1.3, for example, was one of the earliest samples used in the training set. When this image was binarized and then run through the system, it provided fair but not very accurate results. Using feature spaces, 46 blocks were segmented and type-categorized. Then, using the original classifi-

cation and grouping rules, the block classification process correctly labeled 39 of these 46 blocks (84.78%). Out of the 7 blocks that were mis-labeled, 3 had been wrongly segmented and/or type-categorized, and 4 were "special" blocks (page banner, etc.) that were wrongly classified. A total of 5 complete units were created by the block-grouping process, using 24 blocks. One unit was partially created (it was missing a major headline). Of the complete units created, the correct reading order was determined for each one.

The problem of wrong block segmentation was found to be the result of a combination of bad image capture (these images were originally digitized using an Eikonix camera) and "bleedthrough" of newsprint. When the image capture method was improved for the images in the test set (by using a Microtex flatbed scanner), the segmentation performance also improved significantly.

Figure 7.1 and Figure 7.2 show the results obtained from the image of a page from *The Buffalo News*. Figure 7.1 (a) shows the original newspaper image. Figure 7.1 (b) shows the segmented image, with the segmented blocks shown in white. Figure 7.2 shows the output of the DeLoS system. Each segmented and type-categorized block is represented by two lines. The details shown for each block are the block number, the block dimensions (width, height), the block coordinates (x, y coordinates of the upper-left corner), the block type, the block label, the previous and next text block in logical reading order, and the unit number to which the block belongs. This output contains the filled-in information about the blocks (i.e., block labels) and also unit details (unit number, number of blocks, and a list of blocks in the unit). A unit number of 99 represents a block that was not grouped into any specific unit. The filled in block numbers in the "prevtxt" and "nexttxt" fields create the logical links that determine the read-order of text blocks within a unit. A "prevtxt" value of 99 means that the current block is the first text block in its unit in terms of read-order, and

a "nexttxt" value of 99 means that the current block is the last text block in its unit in terms of read-order. For non-text blocks, the "prevtxt" and "nexttxt" fields have a null value (i.e., block number "0").

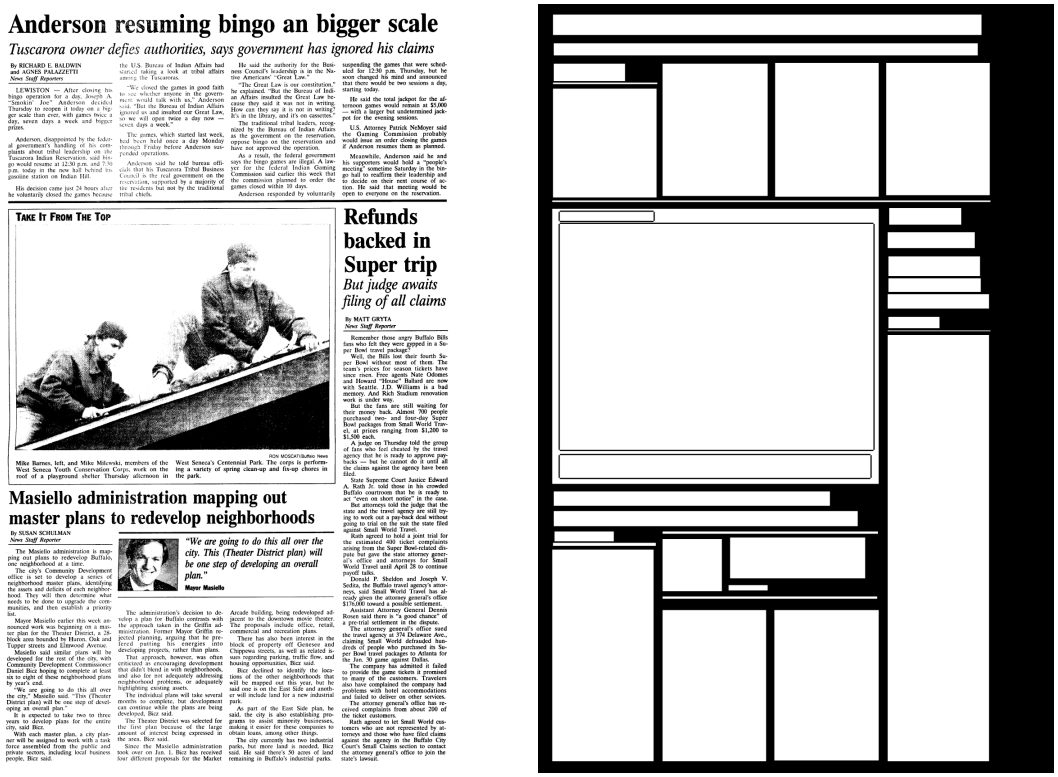
Figure 7.3 shows the reconstructed physical structure of the page, showing the different blocks in the image. As mentioned previously, the reconstructed physical structure is obtained by running a format conversion program to convert the DeLoS output to a LaTeX format file (consisting primarily of commands in the LaTeX *picture* environment), which is then processed by LaTeX to produce the output structure shown.

Evaluation of the results on each image was done by comparison with a truth file that was generated by means of manual inspection from the original image and contained the true classification, grouping and read-ordering data.

Results obtained by running the DeLoS system on a page of *USA Today* are presented in Figure 7.4, which shows the original binary document image and the segmented image, and Figure 7.5, which shows the output document data file from DeLoS. Figure 7.6 shows the reconstructed physical structure of the newspaper page obtained by using the DeLoS output data (by means of data conversion and LaTeX formatting) to reconstruct the page.

7.2 Analysis of System Performance

So far, the DeLoS system has been tested on a variety of pages from *The Buffalo News* and *USA Today*. A total of 44 binary images of newspaper pages (12 images of pages of *USA Today* and 32 images of pages of *The Buffalo News*) were analyzed through the system. Table 7.1 shows the results obtained from the *USA Today* pages, and Table 7.2 shows the results obtained from analyzing images of *The Buffalo News*.



Anderson resuming bingo an bigger scale

Tuscarora owner defies authorities, says government has ignored his claims

By RICHARD E. BALDWIN and AGNES PALAZZETTI
News Staff Reporters

LEWISTON — After closing his bingo operation for a day, Joseph A. "Smokin' Joe" Anderson decided Thursday to reopen it today on a bigger scale than ever, with games twice a day, seven days a week and bigger prizes.

Anderson, disappointed by the federal government's handling of his complaints about tribal leadership on the Tuscarora Indian Reservation, said bingo would resume at 12:30 p.m. and 7:30 p.m. today in the new facility behind his gasoline station on Indian Hill.

His decision came just 24 hours after he voluntarily closed the games because the U.S. Bureau of Indian Affairs had warned him that it, not tribal affairs agency, the Tuscarora.

"We closed the games in good faith to see whether lessons in the government would be learned," Anderson said. "But the Bureau of Indian Affairs ignored us and insulted our Great Law, so we will open twice a day now — seven days a week."

The games, which started last week, had been held once a day Monday through Friday before Anderson suspended operations.

Anderson said he told bureau officials in Tuscarora Tribal Business Council is the real government on the reservation, supported by a majority of the residents but not by the traditional tribal chiefs.

He said the authority for the Business Council's leadership is in the Native Americans' "Great Law."

"The Great Law is our constitution," he explained. "But the Bureau of Indian Affairs insulted the Great Law because they said it was not in writing. How can they say it is not in writing. It's in the library, and it's on cassettes.

The traditional tribal leaders, recognized by the Bureau of Indian Affairs as the government on the reservation, charge that the games operations are have not approved the operation.

As a result, the federal government says the bingo games are illegal. A lawyer for the federal Indian Gaming Commission said earlier this week that the commission planned to order the games closed within 10 days.

Anderson responded by voluntarily suspending the games that were scheduled for 12:30 p.m. Thursday, but he soon changed his mind and announced that there would be two sessions a day, starting today.

He said the total jackpot for the afternoon games would remain at $3,000 — with a larger but undetermined jackpot for the evening sessions.

U.S. Attorney Patrick McMoyer said the Gaming Commission probably would issue an order closing the games if Anderson resumes them as planned.

Meanwhile, Anderson said he and his supporters would hold a "people's meeting" sometime Saturday in the bingo hall to reaffirm their leadership and to decide on their next course of action. He said that meeting would be open to everyone on the reservation.

TAKE IT FROM THE TOP

RON MOSCATI/Buffalo News

Mike Barnes, left, and Mike Milcvski, members of the West Seneca Youth Conservation Corps, work on the roof of a playground shelter Thursday afternoon in West Seneca's Centennial Park. The corps is performing a variety of spring clean-up and fix-up chores in the park.

Refunds backed in Super trip

But judge awaits filing of all claims

By MATT GRYTA
News Staff Reporter

Remember those angry Buffalo Bills fans who felt they were gypped in a Super Bowl travel package?

Well, the Bills lost their fourth Super Bowl without most of them. The team's prices for season tickets have since risen. Free agents Nate Odomes and Howard "House" Ballard are now with Seattle. J.D. Williams is a bad memory. And Rich Stadium renovation work is under way.

But the fans are still waiting for their money back. Almost 700 people purchased two- and four-day Super Bowl packages from Small World Travel, at prices ranging from $1,200 to $1,500 each.

A judge on Thursday told the group of fans who feel cheated by the travel agency that he is ready to approve paybacks — but he cannot do it until all the claims against the agency have been filed.

State Supreme Court Justice Edward A. Rath Jr. told those in his crowded Buffalo courtroom that he has to act "on very short notice" in the case.

But attorneys told the judge that the state and the travel agency are still trying to work out a pay-back deal without going to trial on the suit the state filed against Small World Travel.

Rath agreed to hold a joint trial for the estimated 400 ticket complaints filed with the attorney general's office and has given the state attorney general's office and attorneys for Small World Travel until April 28 to continue payoff talks.

Donald P. Sheldon and Joseph V. Sedita, the Buffalo travel agency's attorneys, said Small World Travel has already given the attorney general's office $1 million in refunds and vouchers.

Assistant Attorney General Dereca Rosen said there is "a good chance" of a pre-trial settlement in the dispute.

The attorney general's office sued the travel agency at 376 Delaware Ave., claiming Small World defrauded hundreds of people who purchased its Super Bowl travel packages to Atlanta for the 1991 and 1992 Super Bowls.

The company has admitted it failed to provide the game tickets it promised to many of the customers. Travelers also have complained the company had problems with hotel accommodations and failed to deliver on other services.

The attorney general's office has received complaints from about 200 of the ticket holders.

Rath agreed to let Small World customers who are not represented by attorneys and those who have filed claims against the agency in the Buffalo City Court's Small Claims section to contact the attorney general's office to join the state's lawsuit.

Masiello administration mapping out master plans to redevelop neighborhoods

By SUSAN SCHULMAN
News Staff Reporter

The Masiello administration is mapping out plans to redevelop Buffalo, one neighborhood at a time.

The city's Community Development office is set to develop a series of neighborhood master plans, identifying the needs of each area in the neighborhood. They will then determine what needs to be done to upgrade the communities, and then establish a priority list.

Mayor Masiello earlier this week announced work was beginning on a master plan for the Theater District, a 24 block area bounded by Huron, Oak and Terrace streets and Main Street.

Masiello said similar plans will be developed for the rest of the city, with Community Development Commissioner Daniel Bice hoping to complete at least six to eight of these neighborhood plans by year's end.

"We are going to do this all over the city," Masiello said. "This (Theater District plan) will be one step of developing an overall plan."

It is expected to take two to three years to develop plans for the entire city, said Bice.

With each master plan, a city planner will be assigned to work with a task force assembled from the public and private sectors, including local business people, Bice said.

"We are going to do this all over the city. This (Theater District plan) will be one step of developing an overall plan."
Mayor Masiello

The administration's decision to develop a plan for Buffalo contrasts with the approach taken in the Griffin administration, which rejected neighborhood planning, arguing that he preferred putting his energies into developing projects, rather than plans.

That approach, however, was often criticized as encouraging development that didn't blend in with neighborhoods, and also for not adequately addressing neighborhood problems, or adequately highlighting existing assets.

The individual plans will take several years to complete, but the projects already can continue while the plans are being developed, Bice said.

The Theater District was selected for the first plan because of the large amount of interest being expressed in the area, Bice said.

Since the Masiello administration took over on Jan. 1, Bice has received four different proposals for the Market Arcade building, being redeveloped adjacent to the downtown movie theater. The proposals include office, retail, and housing opportunities, Bice said.

There has also been interest in the block of property off Genesee and Chippewa streets, as well as related issues regarding parking, traffic flow, and housing opportunities, Bice said.

Bice declined to identify the locations of the other neighborhoods that will be mapped out this year, but he said one is on the East Side and another will include land for a new industrial park.

As part of the East Side plan, he said, the city is also establishing programs to assist minority businesses, making it easier for these companies to obtain loans, among other things.

The city currently has two industrial parks, but more land is needed, Bice said. He said there's 50 acres of land remaining in Buffalo's industrial parks.

(a) Original newspaper page (b) Segmented image

Figure 7.1: Results of block segmentation on a page from *The Buffalo News*.

```
block(1,[dim(2855,135),pos(100,70),type(3),blk_lbl(31)]).
block(1,[prevtxt(0),nexttxt(0),unit_lbl(1)]).
block(2,[dim(2825,80),pos(105,260),type(2),blk_lbl(21)]).
block(2,[prevtxt(0),nexttxt(0),unit_lbl(1)]).
block(3,[dim(475,115),pos(105,395),type(1),blk_lbl(13)]).
block(3,[prevtxt(0),nexttxt(0),unit_lbl(1)]).
block(4,[dim(695,10),pos(100,520),type(8),blk_lbl(81)]).
block(4,[prevtxt(0),nexttxt(0),unit_lbl(1)]).
block(5,[dim(690,720),pos(100,560),type(1),blk_lbl(11)]).
block(5,[prevtxt(99),nexttxt(6),unit_lbl(1)]).
block(6,[dim(700,875),pos(830,395),type(1),blk_lbl(11)]).
block(6,[prevtxt(5),nexttxt(7),unit_lbl(1)]).
block(7,[dim(690,885),pos(1580,390),type(1),blk_lbl(11)]).
block(7,[prevtxt(6),nexttxt(8),unit_lbl(1)]).
block(8,[dim(690,875),pos(2320,390),type(1),blk_lbl(11)]).
block(8,[prevtxt(7),nexttxt(99),unit_lbl(1)]).
block(9,[dim(2920,10),pos(95,1300),type(8),blk_lbl(81)]).
block(9,[prevtxt(0),nexttxt(0),unit_lbl(99)]).
block(10,[dim(2175,1820),pos(95,1355),type(9),blk_lbl(91)]).
block(10,[prevtxt(0),nexttxt(0),unit_lbl(2)]).
block(11,[dim(625,60),pos(145,1375),type(2),blk_lbl(31)]).
block(11,[prevtxt(0),nexttxt(0),unit_lbl(2)]).
block(12,[dim(2090,1495),pos(145,1455),type(4),blk_lbl(41)]).
block(12,[prevtxt(0),nexttxt(0),unit_lbl(2)]).
block(13,[dim(2070,150),pos(145,2985),type(1),blk_lbl(12)]).
block(13,[prevtxt(0),nexttxt(0),unit_lbl(2)]).
block(14,[dim(1840,95),pos(105,3225),type(2),blk_lbl(31)]).
block(14,[prevtxt(0),nexttxt(0),unit_lbl(3)]).
block(15,[dim(2025,100),pos(105,3355),type(2),blk_lbl(31)]).
block(15,[prevtxt(0),nexttxt(0),unit_lbl(3)]).
block(16,[dim(395,65),pos(110,3490),type(1),blk_lbl(13)]).
block(16,[prevtxt(0),nexttxt(0),unit_lbl(3)]).
block(17,[dim(685,20),pos(100,3565),type(8),blk_lbl(81)]).
block(17,[prevtxt(0),nexttxt(0),unit_lbl(3)]).
block(18,[dim(675,1400),pos(95,3610),type(1),blk_lbl(11)]).
block(18,[prevtxt(99),nexttxt(24),unit_lbl(3)]).
block(19,[dim(1435,15),pos(830,3490),type(8),blk_lbl(81)]).
block(19,[prevtxt(0),nexttxt(0),unit_lbl(99)]).
block(20,[dim(395,345),pos(830,3540),type(4),blk_lbl(41)]).
block(20,[prevtxt(0),nexttxt(0),unit_lbl(99)]).
block(21,[dim(900,270),pos(1280,3530),type(2),blk_lbl(31)]).
block(21,[prevtxt(0),nexttxt(0),unit_lbl(99)]).
block(22,[dim(260,35),pos(1270,3845),type(1),blk_lbl(13)]).
block(22,[prevtxt(0),nexttxt(0),unit_lbl(99)]).
block(23,[dim(1430,15),pos(830,3920),type(8),blk_lbl(81)]).
block(23,[prevtxt(0),nexttxt(0),unit_lbl(99)]).
block(24,[dim(690,1000),pos(830,4010),type(1),blk_lbl(11)]).
block(24,[prevtxt(18),nexttxt(25),unit_lbl(3)]).
block(25,[dim(695,995),pos(1575,4010),type(1),blk_lbl(11)]).
block(25,[prevtxt(24),nexttxt(99),unit_lbl(3)]).
block(26,[dim(480,110),pos(2340,1350),type(3),blk_lbl(31)]).
block(26,[prevtxt(0),nexttxt(0),unit_lbl(4)]).
block(27,[dim(580,105),pos(2330,1510),type(3),blk_lbl(31)]).
block(27,[prevtxt(0),nexttxt(0),unit_lbl(4)]).
block(28,[dim(610,130),pos(2335,1670),type(3),blk_lbl(31)]).
block(28,[prevtxt(0),nexttxt(0),unit_lbl(4)]).
block(29,[dim(615,90),pos(2335,1815),type(2),blk_lbl(21)]).
block(29,[prevtxt(0),nexttxt(0),unit_lbl(4)]).
block(30,[dim(675,95),pos(2330,1920),type(2),blk_lbl(21)]).
block(30,[prevtxt(0),nexttxt(0),unit_lbl(4)]).
block(31,[dim(340,75),pos(2335,2070),type(1),blk_lbl(13)]).
block(31,[prevtxt(0),nexttxt(0),unit_lbl(4)]).
block(32,[dim(685,5),pos(2330,2160),type(8),blk_lbl(81)]).
block(32,[prevtxt(0),nexttxt(0),unit_lbl(4)]).
block(33,[dim(675,2820),pos(2330,2190),type(1),blk_lbl(11)]).
block(33,[prevtxt(99),nexttxt(99),unit_lbl(4)]).
unit(1,blocks(8),blocknum([1,2,3,4,5,6,7,8])).
unit(2,blocks(4),blocknum([10,11,12,13])).
unit(3,blocks(7),blocknum([14,15,16,17,18,24,25])).
unit(4,blocks(8),blocknum([26,27,28,29,30,31,32,33])).
unit(99,blocks(6),blocknum([9,19,20,21,22,23])).
```

Figure 7.2: Output of DeLoS for the page from *The Buffalo News*.

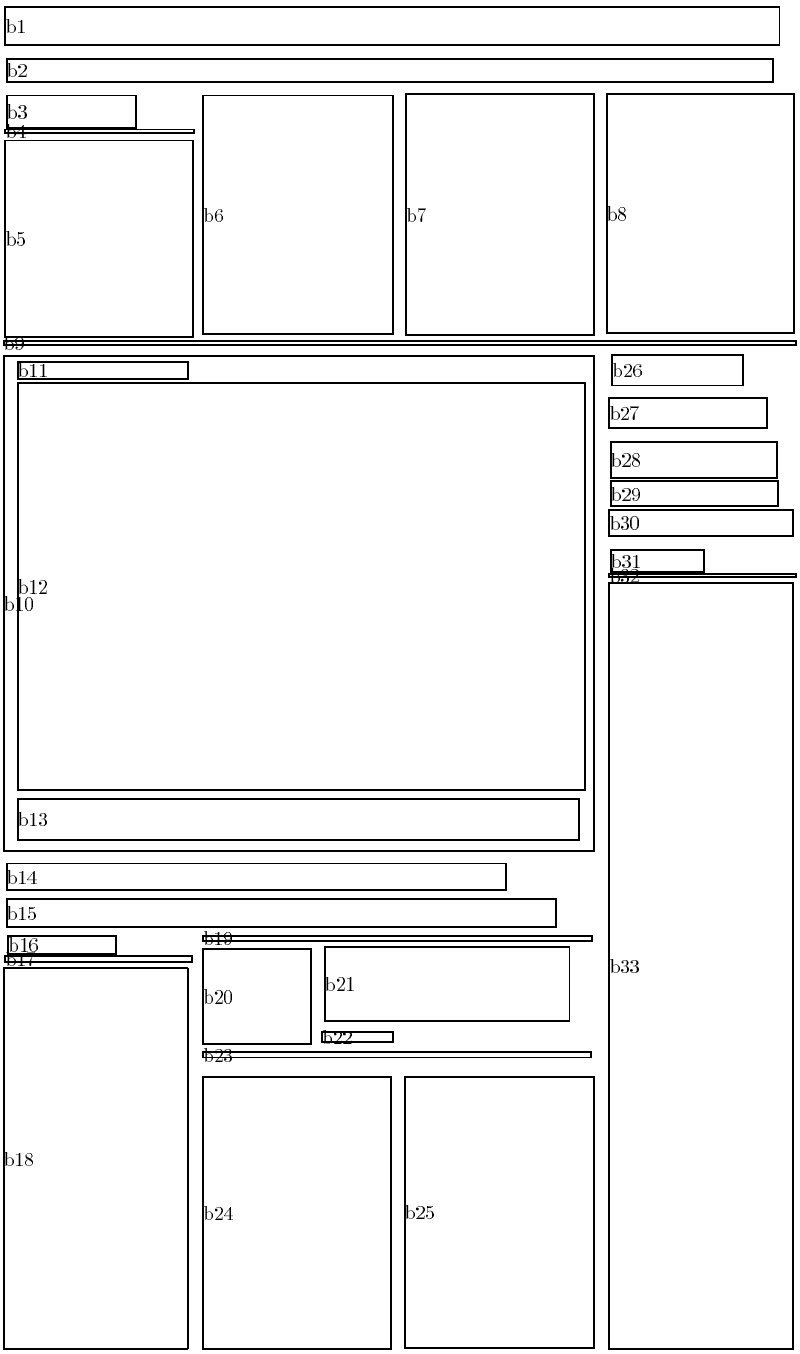


Figure 7.3: Reconstructed physical structure for the page from *The Buffalo News*.

Bomb suspect intent, animated

Rubble yielded key clue

By Carl Weiser
USA TODAY

At first, the FBI's only clue was a twisted, scorched piece of metal.

But that mangled axle from a rented Econoline van led the FBI — along with luck, and apparent ineptitude by a prime suspect — to the first arrest in the bombing of the World Trade Center.

"They did a remarkable job," said President Clinton.

The axle, collected from the perimeter of the garage's bomb crater, tipped police to the van rented by a New Jersey Muslim fundamentalist named Mohammed Salameh.

Salameh also left an obvious paper trail that clearly helped the FBI. Despite an intense ongoing investigation, police say Salameh:

▶ Doggedly tried to retrieve his $400 deposit from the company that rented him the van — the van, authorities said, which carried enough explosives to blow a crater four stories deep in the World Trade Center's parking garage.

▶ Listed on his rental agreement a phone number of a New Jersey apartment where a bomb-sniffing dog turned up bomb-making equipment.

▶ Tried to get his money back with a rental receipt bearing traces of nitrate — an ingredient in some explosives, and a substance said to be present at the blast site.

Police say as early as two hours after the van had been blown up in the blast, Salameh returned to DIB Leasing Inc. in New Jersey.

He said the van he had rented had been stolen in Jersey City. He wanted his $200 damage deposit and his $200 rental deposit back.

The rental agency demanded a police report. Salameh made a theft report with Jersey City police and returned Monday again demanding his damage deposit.

At the same time, the FBI was focusing on DIB Leasing.

They had checked parking stubs and videotapes from the garage.

But the big break came when investigators found the twisted axle, flung far from the center of the bomb's crater on a ramp two floors down.

"We were pleasantly surprised," said the FBI's Jim Fox.

Analysts of the charred metal remains yielded the chassis number of a yellow Ford Econoline, easily traced to DIB, a Ryder Truck Rental Inc. shop.

The FBI went there Wednesday night, where they learned Salameh — using his own name — had rented the van on Tuesday Feb. 23, three days before the bombing.

Salameh, a slightly built man with a closely cropped brown beard, was accompanied by another man when he rented the van, police said, but was alone when he returned.

On Thursday, Salameh called DIB yet again, insisting he get his money back.

By that time, the FBI had gotten hold of Salameh's rental agreement. Along with Salameh's signature, it bore traces of nitrate.

When he returned Thursday to get his refund, FBI agents were on the scene. One agent, pretending to be a Ryder representative, interviewed Salameh about the alleged theft.

"He didn't have a clue as to what was going on," said DIB owner Patrick Galasso. "He just wanted that money."

Salameh got only half his $400 back.

"He argued for more money," but didn't get it, Galasso told CNN, adding that Salameh did not act nervous during the transactions.

Salameh left the agency, and was arrested a block and a half away by as many as 10 federal agents carrying machine guns.

"They just grabbed him, threw him down and put the cuffs on him," said witness Dave Koslow.

Charges could carry a life term

By Bruce Frankel
USA TODAY

NEW YORK — The nation got its first glimpse Thursday of the man prosecutors say was tied to "the single most destructive act of terrorism ever committed on American soil."

And what it saw, in a 44-minute federal court hearing, was a man intent and animated — even an Assistant U.S. Attorney Gilmore Childress told of bomb-making materials found in his closet and a judge turned down his request for $5 million bail.

Mohammed Salameh, 26, whose lawyer said had been in the United States for five years, listened carefully to the Arabic interpreter at his side.

"What's in the charge?" Salameh demanded in a high-pitched, heavily accented voice.

After reading the FBI complaint, Judge Richard Owen — a gentle-toned man — summarized the charges: "You blew up the World Trade Center and killed five people."

"Why?" Salameh asked.

His lawyer, Robert Precht, said Salameh is "absolutely innocent" of the charge — which could bring a life sentence. Children said the government was investigating the possibility of seeking the death penalty — which does not exist in New York — under federal law.

The courtroom was silent during the proceeding, despite the presence of about 250 reporters, prosecutors and officials. Several members of Kahane Chai, a militant Jewish organization, also watched.

At the center of it all was the slightly built, sharp-featured Salameh, who stood in a gray jogging suit. His hair and beard were neatly trimmed, and he wore clean white sneakers.

As Salameh was led away, Precht said Salameh told him he was hungry and "faint."

Owen ordered that a meal be given to him.

ANNOUNCING THE ARREST: We have a long road ahead of us," FBI Director William Sessions, right, said Thursday in a news conference with acting Attorney General Stuart Gerson.

By Tim Dillon, USA TODAY

RENTAL COMPANY: Officials believe a van that carried the bomb was rented from this lot in Jersey City.

By Paul Hurschmann, AP

Anger, fear of backlash follow arrest

By Andrea Stone
USA TODAY

Anger, relief and apprehension came with Thursday's arrest of a suspect in last week's World Trade Center bombing.

"There isn't enough punishment for someone that did these horrible things," said World Trade Center elevator operator Curt Ronning.

"That was very good detective work to find (suspects) so quickly," said Reginald Gonzi, manager of the Sbarro Restaurant beneath the trade center. "It's a relief."

Port authority police Lt. Michael Podolak, who rescued 72 people trapped in an elevator, was "ecstatic" about the arrest but dismayed that the suspect is from the New York area.

"That's all the more insulting," he said. "This is a free country based on democratic principles, and they assault the very foundations of our country and hide behind those very same principles."

Said teacher Anna Marie Tessmers, who with her 17 kindergarten pupils from Brooklyn's P.S. 95 was in an elevator and rescued by Podolak: "I hope they caught the right person. Justice should be done."

That also was the hope of many Arab-Americans. A diverse community, U.S. Muslims are estimated at 4 million to 6 million, with biggest concentrations in New York, Los Angeles, Detroit, Chicago and Washington.

Muslim leaders feared a retaliatory backlash:

▶ "I feel personally upset. I hope the American public would understand this certainly is not representative" of Arab-Americans, said Mona Aboelnaga, a New York activist. "Such an act reflects poorly on many innocent American Arabs who would never support such activity."

▶ The media and officials should take care "not to tar and feather the entire American Arab community" as it did during the Persian Gulf war, said Albert Mokhiber, president of the American-Arab Anti-Discrimination Committee.

▶ "It irks the hell out of every sensitive person," said M.T. Mehdi, president of the American Arab Relations Committee. "But ultimately, this is an individual, crazy action."

"It's very disconcerting," said Shula Bahat of the American Jewish Committee. "Terrorism is no longer something confined to the Middle East. It can be imported" anywhere.

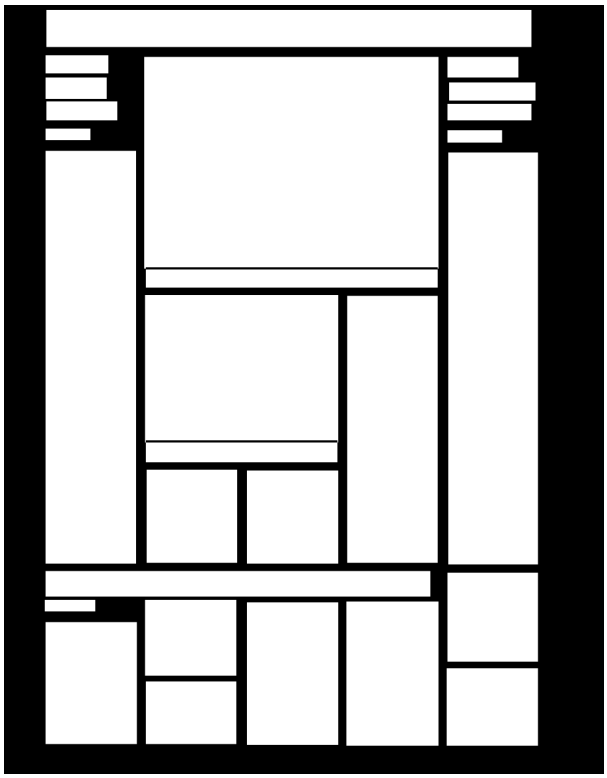
Contributing: Jessica Lee and Bruce Frankel

BOMBING COVERAGE

▶ Suspect charged, 1A
▶ Cover story, 1A
▶ Cairo follows leader, 2A
▶ Mosque denies ties, 2A
▶ Rebuilding towers, 6A
▶ Editorial, 10A

"We fear Islam and Muslims will be stereotyped as fanatics."

— Abdurahman Alamoudi, American Muslim Council



(a) Original newspaper page (b) Segmented image

Figure 7.4: Results of block segmentation on a *USA Today* page.

```
block(1,[dim(1785,135),pos(156,18),type(3),blk_lbl(31)]).
block(1,[prevtxt(0),nexttxt(0),unit_lbl(1)]).
block(2,[dim(231,66),pos(153,183),type(2),blk_lbl(21)]).
block(2,[prevtxt(0),nexttxt(0),unit_lbl(1)]).
block(3,[dim(225,78),pos(153,264),type(2),blk_lbl(21)]).
block(3,[prevtxt(0),nexttxt(0),unit_lbl(1)]).
block(4,[dim(261,69),pos(156,351),type(2),blk_lbl(21)]).
block(4,[prevtxt(0),nexttxt(0),unit_lbl(1)]).
block(5,[dim(165,42),pos(153,450),type(1),blk_lbl(13)]).
block(5,[prevtxt(0),nexttxt(0),unit_lbl(1)]).
block(6,[dim(333,1503),pos(153,531),type(1),blk_lbl(11)]).
block(6,[prevtxt(99),nexttxt(11),unit_lbl(1)]).
block(7,[dim(1083,771),pos(516,189),type(4),blk_lbl(41)]).
block(7,[prevtxt(0),nexttxt(0),unit_lbl(1)]).
block(8,[dim(1074,66),pos(522,963),type(1),blk_lbl(12)]).
block(8,[prevtxt(0),nexttxt(0),unit_lbl(1)]).
block(9,[dim(711,540),pos(519,1056),type(4),blk_lbl(41)]).
block(9,[prevtxt(0),nexttxt(0),unit_lbl(1)]).
block(10,[dim(705,72),pos(522,1593),type(1),blk_lbl(11)]).
block(10,[prevtxt(0),nexttxt(0),unit_lbl(1)]).
block(11,[dim(333,339),pos(525,1692),type(1),blk_lbl(11)]).
block(11,[prevtxt(6),nexttxt(12),unit_lbl(1)]).
block(12,[dim(336,339),pos(894,1695),type(1),blk_lbl(11)]).
block(12,[prevtxt(11),nexttxt(13),unit_lbl(1)]).
block(13,[dim(333,972),pos(1263,1059),type(1),blk_lbl(11)]).
block(13,[prevtxt(12),nexttxt(99),unit_lbl(1)]).
block(14,[dim(261,75),pos(1632,189),type(2),blk_lbl(21)]).
block(14,[prevtxt(0),nexttxt(0),unit_lbl(2)]).
block(15,[dim(318,66),pos(1638,282),type(2),blk_lbl(21)]).
block(15,[prevtxt(0),nexttxt(0),unit_lbl(2)]).
block(16,[dim(309,60),pos(1632,360),type(2),blk_lbl(21)]).
block(16,[prevtxt(0),nexttxt(0),unit_lbl(2)]).
block(17,[dim(201,45),pos(1632,456),type(1),blk_lbl(13)]).
block(17,[prevtxt(0),nexttxt(0),unit_lbl(2)]).
block(18,[dim(330,1500),pos(1635,537),type(1),blk_lbl(11)]).
block(18,[prevtxt(99),nexttxt(20),unit_lbl(2)]).
block(19,[dim(333,324),pos(1632,2067),type(4),blk_lbl(41)]).
block(19,[prevtxt(0),nexttxt(0),unit_lbl(2)]).
block(20,[dim(336,282),pos(1629,2415),type(1),blk_lbl(11)]).
block(20,[prevtxt(18),nexttxt(99),unit_lbl(2)]).
block(21,[dim(1416,93),pos(153,2061),type(2),blk_lbl(31)]).
block(21,[prevtxt(0),nexttxt(0),unit_lbl(3)]).
block(22,[dim(186,42),pos(150,2166),type(1),blk_lbl(13)]).
block(22,[prevtxt(0),nexttxt(0),unit_lbl(3)]).
block(23,[dim(336,444),pos(153,2247),type(1),blk_lbl(11)]).
block(23,[prevtxt(99),nexttxt(25),unit_lbl(3)]).
block(24,[dim(336,276),pos(519,2166),type(4),blk_lbl(41)]).
block(24,[prevtxt(0),nexttxt(0),unit_lbl(3)]).
block(25,[dim(333,228),pos(522,2463),type(1),blk_lbl(11)]).
block(25,[prevtxt(23),nexttxt(26),unit_lbl(3)]).
block(26,[dim(336,519),pos(894,2175),type(1),blk_lbl(11)]).
block(26,[prevtxt(25),nexttxt(27),unit_lbl(3)]).
block(27,[dim(339,525),pos(1260,2172),type(1),blk_lbl(11)]).
block(27,[prevtxt(26),nexttxt(99),unit_lbl(3)]).
unit(1,blocks(13),blocknum([1,2,3,4,5,6,7,8,9,10,11,12,13])).
unit(2,blocks(7),blocknum([14,15,16,17,18,19,20])).
unit(3,blocks(7),blocknum([21,22,23,24,25,26,27])).
```

Figure 7.5: Output of DeLoS for the *USA Today* page.

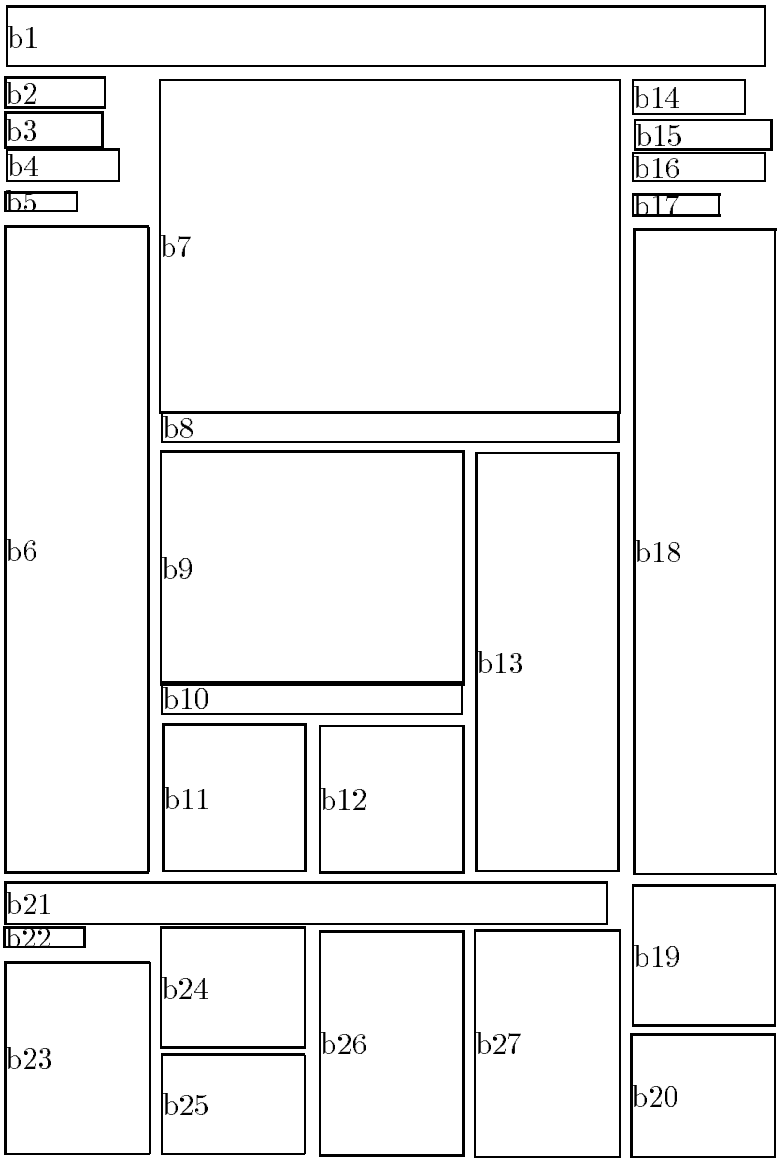


Figure 7.6: Reconstructed physical structure for the *USA Today* page.

Document ID	Original Image			Blocks Seg.	Correct Classif.	Blocks Grouped	No. of Units	Classif. As Text	Correct Rd-Ord.
	Blocks	Units	Text						
Page01	26	3	11	27	25	19	3	13	8
Page02	31	3	13	31	30	25	3	13	12
Page03	27	2	13	27	23	27	3	13	13
Page04	19	1	8	17	16	14	1	8	7
Page05	23	4	7	22	21	19	4	9	5
Page06	27	3	10	26	23	18	3	8	8
Page07	26	3	11	25	23	22	3	12	11
Page08	18	2	7	18	18	17	2	8	7
Page09	27	1	7	27	19	23	2	10	4
Page11	35	4	15	35	35	34	4	15	13
Page12	23	2	6	23	19	16	2	7	5
Page13	25	4	8	24	21	12	2	9	6

where

```
ORIGINAL IMAGE BLOCKS:  Total number of blocks in the original image
ORIGINAL IMAGE UNITS:   No. of logical units in the original image
ORIGINAL IMAGE TEXT:    No. of text blocks in the original image
BLOCKS SEG.:            No. of blocks obtained from segmentation
CORRECT CLASSIF.:       No. of blocks correctly classified by DeLoS
BLOCKS GROUPED:         No. of blocks grouped into units by DeLoS
NO. OF UNITS:           No. of logical units formed by block grouping
CLASS. AS TEXT:         No. of blocks classified as text blocks
CORRECT RD-ORD.:        No. of text blocks correctly read-ordered
```

Table 7.1: Results of DeLoS on pages of *USA Today*.

Document ID	Original Image			Blocks Seg.	Correct Classif.	Blocks Grouped	No. of Units	Classif. As Text	Correct Rd-Ord.
	Blocks	Units	Text						
BN-01	34	4	8	33	31	27	4	8	8
BN-02	33	5	8	32	32	31	5	8	8
BN-03	51	3	10	42	38	28	3	11	6
BN-04	37	4	6	34	33	31	4	6	6
BN-05	41	3	9	37	36	34	4	9	9
BN-06	35	6	10	34	32	30	6	10	9
BN-07	37	4	5	36	32	31	4	7	5
BN-08	35	3	8	34	31	29	3	8	7
BN-09	30	3	7	30	29	26	3	7	7
BN-10	33	4	11	32	30	27	4	11	10
BN-11	27	5	8	27	26	24	5	8	8
BN-12	32	4	6	31	31	28	4	6	5
BN-13	35	4	8	35	34	33	5	8	7
BN-14	29	3	7	28	25	25	3	5	7
BN-15	31	3	8	30	29	27	3	8	8
BN-16	45	5	6	43	39	36	5	6	6
BN-17	40	3	8	40	38	35	3	8	8
BN-18	40	4	8	40	39	36	5	8	6
BN-19	37	5	7	36	36	33	5	7	7
BN-20	44	5	8	43	40	35	5	8	8
BN-21	32	5	5	31	29	28	5	7	6
BN-22	46	4	16	45	44	42	4	14	12
BN-23	46	4	9	44	43	40	4	8	8
BN-24	31	3	7	30	30	28	3	7	5
BN-25	35	3	8	35	34	33	3	8	8
BN-26	37	3	9	37	36	34	3	9	8
BN-27	43	4	8	42	40	37	4	8	8
BN-28	34	5	8	32	30	28	5	8	7
BN-29	31	3	8	30	30	29	3	8	8
BN-30	47	5	9	45	43	40	5	9	9
BN-31	41	3	9	39	38	36	3	9	9
BN-32	33	4	7	32	32	30	4	7	6

See Table 7.1 for explanation of column headings.

Table 7.2: Results of DeLoS on pages of *The Buffalo News*.

RESULTS

As can be seen from Table 7.1 and Table 7.2, block segmentation and block type-categorization in the original images proved to be a deciding factor in the performance of the system. The "docstrum" program when run on the original binary images gave only the text blocks accurately, and categorized most of the headline blocks as graphics. The connected components comprising these blocks were then extracted and "docstrum" was re-run on only these components, which then yielded a more accurate categorization of medium-sized text blocks (corresponding to minor headlines). Another similar extraction of connected components and a re-run of "docstrum" yielded the large-sized text blocks (corresponding to major headlines). Also, since graphical blocks often have similar geometrical extents as rectangular enclosing rectangles, the latter were categorized by computing the block's pixel-density *pixden*, which is defined as the ratio of the number of pixels in the block to the area of the block. A very low value of *pixden* indicates an enclosing rectangle. In addition, horizontal and vertical lines were categorized by recognizing blocks which had a very small height or width respectively.

Even after further processing of the "docstrum" segmentation output, some block segmentation errors remained. In particular, touching blocks or text blocks in very close proximity (e.g., photo credits whose letters touched the photograph boundary, photo credits that were very close to the photo caption, etc.) were merged together by the segmentation process. Such errors influenced the accuracy of the DeLoS block classification process, and thus carried over into the block grouping and read-ordering processes as well. Overall, the DeLoS system performed fairly well for images from *The Buffalo News* as well as *USA Today*. Table 7.3 and Table 7.4 show the performance of the system for both these kinds of newspaper pages, in terms of percentages of the original blocks correctly classified, grouped and read-ordered. It also shows the percentages of blocks correctly classified, grouped and

read-ordered in terms of the segmented image. As we can see, the performance of the system was dependent on the performance of the block segmentation process.

Processing time for the DeLoS system to analyze one newspaper page currently varies from 12 seconds to 54 seconds of CPU time on a SPARC-2 workstation. The variation is due to the different degrees of complexity of the different pages. The above does not include the time required for the image processing operations (which are very computation-intensive, especially the "docstrum" segmentation method, each run of which takes about 30-35 minutes (physical time, not CPU time) on a SPARC-10 workstation to segment each image). The processing speed can be further improved by doing all the processing on a dedicated machine so that data is not transferred across file systems. Other ways of improving the speed include compiling the Prolog code, or converting the entire system to C/C++ .

Document ID.	Classif. (Original)	Grouping (Original)	Read-Ord. (Original)	Classif. (Blk-Seg.)	Grouping (Blk-Seg.)	Read-Ord. (Blk-Seg.)
Page01	96.1 %	73 %	72.7 %	92.5 %	70.3 %	61.5 %
Page02	96.7 %	80.6 %	92.3 %	96.7 %	80.6 %	92.3 %
Page03	85.1 %	100 %	100 %	85.1 %	100 %	100 %
Page04	84.2 %	73.6 %	87.5 %	94.1 %	82.3 %	87.5 %
Page05	91.3 %	82.6 %	71.4 %	95.4 %	86.3 %	55.5 %
Page06	85.1 %	66.6 %	80 %	88.4 %	69.2 %	100 %
Page07	88.4 %	84.6 %	100 %	92 %	88 %	91.6 %
Page08	100 %	94.4 %	100 %	100 %	94.4 %	87.5 %
Page09	70.3 %	85.1 %	57.1 %	70.3 %	85.1 %	40 %
Page11	100 %	97.1 %	86.6 %	100 %	97.1 %	86.6 %
Page12	82.6 %	69.5 %	83.3 %	82.6 %	69.5 %	71.4 %
Page13	84 %	48 %	75 %	87.5 %	50 %	66.6 %

where

```
CLASSIF. (ORIGINAL)  = % of blocks from original image that were
                              correctly classified
GROUPING (ORIGINAL)  = % of blocks from original image that were
                              correctly grouped
READ-ORD. (ORIGINAL) = % of text blocks from original image that
                              were correctly read-ordered
CLASSIF. (BLK-SEG.)  = % of blocks from segmented image that were
                              correctly classified
GROUPING (BLK-SEG.)  = % of blocks from segmented image that were
                              correctly grouped
READ-ORD. (BLK-SEG.) = % of text blocks from segmented image that
                              were correctly read-ordered
```

Table 7.3: Performance of DeLoS on pages of *USA Today*.

Document ID.	Classif. (Original)	Grouping (Original)	Read-Ord. (Original)	Classif. (Blk-Seg.)	Grouping (Blk-Seg.)	Read-Ord. (Blk-Seg.)
BN-01	91.1 %	79.4 %	100 %	93.9 %	81.8 %	100 %
BN-02	96.9 %	93.9 %	100 %	100 %	96.8 %	100 %
BN-03	74.5 %	54.9 %	60 %	90.4 %	66.6 %	54.5 %
BN-04	89.1 %	83.7 %	100 %	97 %	91.1 %	100 %
BN-05	87.8 %	82.9 %	100 %	97.2 %	91.8 %	100 %
BN-06	91.4 %	85.7 %	90 %	94.1 %	88.2 %	90 %
BN-07	86.4 %	83.7 %	100 %	88.8 %	86.1 %	71.4 %
BN-08	88.5 %	82.8 %	87.5 %	91.1 %	85.2 %	87.5 %
BN-09	96.6 %	86.6 %	100 %	96.6 %	86.6 %	100 %
BN-10	90.9 %	81.8 %	90.9 %	93.7 %	84.3 %	90.9 %
BN-11	96.2 %	88.8 %	100 %	96.2 %	88.8 %	100 %
BN-12	96.8 %	87.5 %	83.3 %	100 %	90.3 %	83.3 %
BN-13	97.1 %	94.2 %	87.5 %	97.1 %	94.2 %	87.5 %
BN-14	86.2 %	86.2 %	100 %	89.2 %	89.2 %	71.4 %
BN-15	93.5 %	87 %	100 %	96.6 %	90 %	100 %
BN-16	86.6 %	80 %	100 %	90.6 %	83.7 %	100 %
BN-17	95 %	87.5 %	100 %	95 %	87.5 %	100 %
BN-18	97.5 %	90 %	75 %	97.5 %	90 %	75 %
BN-19	97.2 %	89.1 %	100 %	100 %	91.6 %	100 %
BN-20	90.9 %	79.5 %	100 %	93 %	81.3 %	100 %
BN-21	90.6 %	87.5 %	83.3 %	93.5 %	90.3 %	85.7 %
BN-22	95.6 %	91.3 %	75 %	97.7 %	93.3 %	85.7 %
BN-23	93.4 %	86.9 %	88.8 %	97.7 %	90.9 %	100 %
BN-24	96.7 %	90.3 %	71.4 %	100 %	93.3 %	71.4 %
BN-25	97.1 %	94.2 %	100 %	97.1 %	94.2 %	100 %
BN-26	97.2 %	91.8 %	88.8 %	97.2 %	91.8 %	88.8 %
BN-27	93 %	86 %	100 %	95.2 %	88 %	100 %
BN-28	88.2 %	82.3 %	87.5 %	93.7 %	87.5 %	87.5 %
BN-29	96.7 %	93.5 %	100 %	100 %	96.6 %	100 %
BN-30	91.4 %	85.1 %	100 %	95.5 %	88.8 %	100 %
BN-31	92.6 %	87.8 %	100 %	97.4 %	92.3 %	100 %
BN-32	96.9 %	90.9 %	85.7 %	100 %	93.7 %	85.7 %

See Table 7.3 for explanation of column headings.

Table 7.4: Performance of DeLoS on pages of *The Buffalo News*.

Chapter 8

Summary and Future Directions

8.1 Summary

We have presented a computational model for the derivation of logical structure in a document by the use of domain knowledge about document layout. The discussions have related to the investigation of such issues as what domain and world knowledge is required by this model, how the knowledge is represented in a globally integrated form, and how a control strategy is devised that efficiently utilizes this knowledge base.

The design and development of a document logical structure derivation system based on this model was also discussed. It was pointed out that the integration of a multi-level rule-based inference engine that matches image features and relationships to identify different types of blocks, and a global data structure that facilitates the use of data from the document image, leads to a more efficient integration of top-down and bottom-up reasoning, and provides a flexible yet powerful control structure for solving object recognition problems in structured domains.

The system can perform mixed-mode analysis of a document, and thus is more powerful than a simple top-down or bottom-up system. The rule-based system is flexible enough to

allow the addition of more knowledge rules to facilitate the analysis of additional document features, and the global data structure can allow the addition of more complex tree frames representing documents with varying levels of complexity or the results of more detailed document understanding methods. The multi-level rule structure also ensures efficiency in computation, since the computational time and effort required to fully analyze a document image is proportional to the quality and structural complexity of the original image.

The approach described in this dissertation has been implemented in DeLoS, a system that can derive the logical structure of a document. We have shown through the development of the model and the results obtained from the implementation of the DeLoS system that:

- The geometric structure of a document page can be translated into rules.
- The logical structure of different types of documents can similarly be encoded into rules.
- The logical structure of a document page can be derived from the physical structure by the application of these structural and semantic rules.

The most obvious use of the document logical structure derivation system is as a part of a comprehensive document understanding system that uses the logical structure information output by this system to read the information in the individual units in a document. Applications include document browsers and document library archival systems.

Digital document libraries have recently gained tremendous national attention, mainly because of the U.S. Government's efforts at creating a *National Information Infrastructure*. The applications of document image understanding systems in creating intelligent digital libraries are manyfold. Systems that can extract logically linked parts of a document and

retrieve them on demand can significantly improve the usefulness and performance of document libraries. To this end, document logical structure derivation is an integral step towards the creation of an intelligent digital document library.

In conclusion, the significant accomplishments of the research presented here can be summarized as follows:

- My approach presents a novel organization of past techniques combined with new algorithms.

- It deals with complex multi-unit documents such as newspapers, unlike most earlier work.

- It defines a general process for read-order determination of text blocks as well as block classification and grouping.

- It uses spatial constraints and document domain knowledge, rather than pre-determined document templates, to derive the reading order in documents.

- It has the modularity of being adaptable to different document domains by substitution of a limited set of rules in the domain knowledge base.

- It allows for convenient modification of the logical structure and reconstruction of the document physical layout.

- It is evaluated by reconstructing the physical structure of the analyzed documents.

8.2 Future Work

The document logical structure derivation system derives as complete a logical description of a document page as is possible without performing actual text reading and semantic understanding of the document contents. In fact, spatial information alone provides a significantly large amount of knowledge that enables an effective logical structure derivation. However, a more complete understanding of document structure and content can obviously be obtained by involving text understanding in the logical structure derivation process. Therefore, any further work based on the current document logical structure derivation model needs to include text understanding.

The potential extensions to the document logical structure derivation system that are suitable to be pursued for future research work are as follows:

1. *Multi-page logical grouping.*

   My research has developed a model for deriving the logical structure of a single-page document. Often, however, multi-page documents may need to be analyzed in a similar manner. For example, a newspaper frequently contains a story that is continued onto another page. In such instances, the logical unit that represents the story should include the logical structure information from the second page as well. Further research work will therefore include multi-page logical grouping and read-order determination. This will involve the understanding of continuation keywords at the bottom and top of story segments. For example, one may find the words "see BUDGET Page 3" at the bottom of a story segment on Page 1 of a newspaper and the words "Continued from Page 1" at the top of the corresponding story segment on Page 3. These words link these two story segments together and direct the human reader's attention from

the first to the second. Thus, multi-page logical grouping should also link these two story segments and create a single grouping and read-order for the entire story. Note however that in the above example, there may be more than one story on Page 3 that is continued from Page 1, and in such cases further text understanding (perhaps of the headlines in the two physical segments of the story) may be required to disambiguate the reference.

2. *Linking non-adjacent blocks to logically grouped units.*

   This is a variation of the multi-page logical grouping issue. The vast majority of logical units in documents involve physically adjacent blocks, and my system effectively derives such units. Occasionally, however, a physically non-adjacent block may "belong" in a particular unit (e.g., a figure or table that is referenced on a different page than the one on which it appears). Such situations are virtually unknown in newspapers but are fairly common in certain types of documents, e.g., journals, magazines, etc. Complete retrieval and "understanding" of logical units should include the linking of such physically separate blocks with the rest of the unit. This will also require keyword understanding (e.g., the words "Figure 2" within the text of an article and the corresponding words "Figure 2" in the caption of the actual figure).

3. *Semantic analysis for read-order determination.*

   Read-order determination can also be helped by knowledge about the contents of the text blocks that are to be ordered. For example, hyphenated words can provide important cues in determining read-order, since it is likely that a text block that starts with the tail segment of a hyphenated word (e.g., "-ument") follows a text block that ends with the head segment of a hyphenated word (e.g., "doc-"). Work in this area will include recognizing hyphenated word segments, extracting the different hyphenated

segments, removing the hyphen, merging the two segments, and then performing word recognition using a stored dictionary. This task will involve character-level as well as word-level text recognition, and perhaps linguistic knowledge about word hyphenation heuristics used in document formatting systems.

Appendix A

DeLoS Domain Knowledge

This appendix contains the rules that encode the domain knowledge in the DeLoS knowledge base, i.e., the domain *knowledge rules* in the DeLoS system. The current domain knowledge pertains to newspaper pages.

Rules for Block Classification

```
MAJOR HEADLINE:

A block X is a major headline IF
    block X is of type large-text.

A block X is a major headline IF
    block X is of type medium-text
      AND block X is a below-neighbor of block Y
      AND block Y is not a major headline
      AND block Y is not a minor headline.

A block X is a major headline IF
    block X is of type medium-text
      AND block X is a below-neighbor of block Y
      AND block Y is of type medium-text
      AND block Y is a major headline.


MINOR HEADLINE:
```

APPENDIX A: RULES

```
A block X is a minor headline IF
    block X is of type medium-text
      AND block X is a below-neighbor of block Y
      AND block Y is a major headline.

A block X is a minor headline IF
    block X is of type medium-text
      AND block X is below block Y
      AND block Y is of type medium-text
      AND block Y is a minor headline.


TEXT BLOCK:

A block X is a text block IF
    block X is of type small-text
      AND block X is not a caption
      AND block X is not a story byline.


STORY BYLINE:

A block X is a story byline IF
    block X is of type small-text
      AND block X is a below-neighbor of block Y
      AND {block Y is a major headline
             OR block Y is a minor headline}
      AND block X is a above-neighbor of block Z
      AND block Z is a row separator
      AND height of block X is less than 150
      AND width of X is less than 500.
                                                    [The Buffalo News]

A block X is a story byline IF
    block X is of type small-text
      AND block X is a below-neighbor of block Y
      AND {block Y is a major headline
             OR block Y is a minor headline}
      AND width of X is less than 250.
                                                    [USA Today]
```

```
A block B2 is a story byline IF
    block B1 is a major/minor headline
      AND block B2 is a below-neighbor of block B1
      AND B2 is of type small-text
      AND block B3 is a below-neighbor of block B2
      AND B3 is of type horizontal-line
      AND block B4 is a below-neighbor of block B3
      AND B4 is a text block.


PHOTOGRAPH:

A block X is a photograph IF
    block X is of type half-tone
      AND block X is not a page banner
      AND block X is not a horizontal band.


CAPTION:

A block X is a caption IF
    block X is of type small-text
      AND block X is a below-neighbor of block Y
      AND block Y is a photograph
      AND block X and block Y are of approximately equal widths
      AND block X has aspect ratio greater than 5.


PAGE BANNER:

A block X is a page banner IF
    block X is of type half-tone
      AND block X is positioned on the top of the page
      AND block X has width greater than 1200
      AND block X has height greater than 100.


HORIZONTAL BAND:

A block X is a horizontal band IF
```

```
    block X is of type half-tone
      AND block X has aspect ratio greater than 8
      AND block X has height nor greater than 100.


LINE DRAWING:

A block X is a line drawing IF
    block X is of type line-draw.


COLUMN SEPARATOR (vertical thin line):

A block X is a column separator IF
    block X is of type vertical-line
      AND block X has aspect ratio of less than 0.1
      AND block X has width not greater than 25.


ROW SEPARATOR (horizontal thin line):

A block X is a row separator IF
    block X is of type horizontal-line
      AND block X has aspect ratio of greater than 10
      AND block X has height not greater than 25.


ENCLOSING RECTANGLE:

A block X is an enclosing rectangle IF
    block X is of type line-rectangle.
```

Rules for Block Grouping

```
IF a block B1 is a major/minor headline
    AND a block B2 is also a major/minor headline
    AND block B1 is an above-neighbor of block B2
    AND the left extents of the blocks B1 and B2 are approximately equal
```

```
THEN group the two blocks B1 and B2 into the same unit.


IF a block B1 is a major headline
    AND a block B2 is a minor headline
    AND block B2 is a below-neighbor of block B1
    AND the following is true:
         {the left extents of B1 and B2 are approximately equal
             OR the right extents of B1 and B2 are approximately equal}
THEN group the two blocks B1 and B2 into the same unit.


IF a block B1 is a major/minor headline
    AND a block B2 is a text block
    AND block B2 is a below-neighbor of block B1
THEN group the blocks B1 and B2 into the same unit.


IF a block B1 is a major/minor headline
    AND a block B2 is a photograph
    AND block B2 is a below-neighbor of block B1
THEN group the blocks B1 and B2 into the same unit.


IF a block B3 is a photograph
    AND a block B4 is a caption
    AND block B4 is a below-neighbor of block B3
    AND the following condition is NOT true:
               {there exists a block B5 which is of type small-text
                  and is adjacent to block B3,
                AND a block B6 which is a photograph
                  and is adjacent to block B5}
    AND the left extents of the blocks B3 and B4 are approximately equal
THEN group the blocks B3 and B4 into the same unit.


IF block B1 is a major/minor headline
    AND block B2 is a below-neighbor of B1
    AND block B2 is a story byline
    AND block B3 is a below-neighbor of B2
    AND block B3 is of type horizontal-line
```

```
    AND block B4 is a below-neighbor of B3
    AND block B4 is a text block
THEN group the blocks B1, B2, B3 and B4 into the same unit.


IF block B1 is a major/minor headline
    AND block B2 is a right-neighbor of B1
    AND B2 is a vertical line
    AND block B3 is a right-neighbor of B2
THEN *do not* group the blocks B1 and B3 into the same unit.


IF block B1 is an enclosing rectangle
    AND blocks B2 and B3 are inside block B1
THEN group the blocks B2 and B3 into the same unit.


IF block B1 is a row separator
    AND block B2 is an above-neighbor of block B1
    AND block B3 is a below-neighbor of block B1
THEN *do not* group the blocks B2 and B3 into the same unit.


IF block B1 is a column separator
    AND block B2 is a left-neighbor of block B1
    AND block B3 is a right-neighbor of block B1
THEN *do not* group the blocks B2 and B3 into the same unit.
```

Rules for Text Block Read-Ordering

```
IF block B1 is a text block
    AND block B2 is a text block
    AND B1 and B2 are in the same unit
    AND B2 is a below-neighbor of B1
    AND B1 and B2 have equal width
THEN B2 is next in read-order after B1.
```

```
IF block B1 is a text block
    AND block B2 is a text block
    AND B1 and B2 are in the same unit
    AND B2 is below B1
    AND there is NO block B3 that satisfies:
          {B3 is a text block
              AND B3 is in the same unit as B1 and B2
              AND {B3 is a below-neighbor of B1
                      OR B3 is an above-neighbor of B2}}
THEN B2 is next in read-order after B1.


IF block B1 is a text block
    AND block B2 is a text block
    AND B1 and B2 are in the same unit
    AND B2 is a right-neighbor of B1
    AND there is NO block B3 that satisfies:
          {B3 is a text block
              AND B3 is in the same unit as B1 and B2
              AND {B3 is a below-neighbor of B1
                      OR B3 is an above-neighbor of B2}}
THEN B2 is next in read-order after B1.


IF block B1 is a text block
    AND block B2 is a text block
    AND B1 and B2 are in the same unit
    AND B2 is right-of B1
    AND there is NO block B3 that satisfies:
          {B3 is a text block
              AND B3 is in the same unit as B1 and B2
              AND {B3 is below B1
                      OR B3 is above B2
                      OR B3 is left-of B2}}
THEN B2 is next in read-order after B1.
```

Functions for DeLoS rule base

```
A block P is above a block Q IF
    Ymax(P) is less than Ymin(Q)
      AND Xmax(P) is greater than Xmin(Q)
      AND Xmin(P) is less than Xmax(Q).


A block P is below a block Q IF
    Ymin(P) is greater than Ymax(Q)
      AND Xmin(P) is less than Xmax(Q)
      AND Xmax(P) is greater than Xmin(Q).


A block P is left-of a block Q IF
    Xmax(P) is less than Xmin(Q)
      AND Ymax(P) is greater than Ymin(Q)
      AND Ymin(P) is less than Ymax(Q).


A block P is right-of a block Q IF
    Xmin(P) is greater than Xmax(Q)
      AND Ymin(P) is less than Ymax(Q)
      AND Ymax(P) is greater than Ymin(Q).


A block X is an above-neighbor of a block Y IF
    block X is above block Y
      AND there is NO block Z that is directly between blocks X and Y.


A block X is a below-neighbor of a block Y IF
    block X is below block Y
      AND there is NO block Z that is directly between blocks X and Y.


A block X is a left-neighbor of a block Y IF
    block X is to the left of block Y
      AND there is NO block Z that is directly between blocks X and Y.
```

APPENDIX A: RULES

```
A block X is a right-neighbor of a block Y IF
    block X is to the right of block Y
      AND there is NO block Z that is directly between blocks X and Y.


A block X is adjacent to a block Y IF
    block X is an above-neighbor of block Y
      OR block X is a below-neighbor of block Y
      OR block X is a left-neighbor of block Y
      OR block X is a right-neighbor of block Y.


A block B is positioned at the top of a page IF
    Ymin(B) is not more than 100
      AND Ymax(B) is not more than 700.


A block B occupies the full width of a page IF
    width(B) is at least 1200.


A dimension/property of block X is approximately equal to
                the same dimension/property of block Y IF
    the two values of the dimension/property differ by less than 10%
                of the smaller value.
```

Bibliography

[1] E.C. Arnold. *Modern Newspaper Design.* Harper and Row, New York, 1969.

[2] D.H. Ballard, C.M. Brown, and J.A. Feldman. An approach to knowledge-directed image analysis. In A. R. Hanson and E. M. Riseman, editors, *Computer Vision Systems*, pages 271–281. Academic Press, New York, 1978.

[3] T.O. Binford. Survey of model-based image analysis systems. *International Journal of Robotics Research*, 1(1):18–62, Spring 1982.

[4] R.A. Brooks. Symbolic reasoning among 3-D models and 2-D images. *Artificial Intelligence*, 17(1–3):285–348, August 1981.

[5] R.A. Brooks. *Model-Based Computer Vision.* UMI Research Press, Ann Arbor, Michigan, 1984.

[6] R.A. Brooks, R. Greiner, and T.O. Binford. The ACRONYM model-based vision system. In *Proceedings of 6th IJCAI*, volume 1, pages 105–113, 1979.

[7] H. Bunke and G. Sagerer. Use and representation of knowledge in image understanding based on semantic networks. In *Proc. of 7th International Conference on Pattern Recognition*, volume 2, pages 1135–1137, Montreal, Canada, July 30–Aug. 2, 1984.

[8] G. Ciardiello, M.T. Degrandi, M.P. Roccotelli, G. Scafuro, and M.R. Spada. An experimental system for office document handling and text recognition. In *Proc. of 9th International Conference on Pattern Recognition*, volume 2, pages 739–743, Rome, Italy, Nov. 14–17, 1988.

[9] W.F. Clocksin and C.S. Mellish. *Programming in Prolog.* Springer-Verlag, 1981.

[10] A.A. Crowell. *Creative News Editing.* Wm. C. Brown Company Publishers, Dubuque, Iowa, 1970.

[11] A. Dengel and G. Barth. ANASTASIL: A hybrid knowledge-based system for document layout analysis. In *Proc. of 11th IJCAI*, volume 2, pages 1249–1254, Detroit, MI, Aug. 20–25, 1989.

[12] D. Derrien-Peden. Frame-based system for macro-typographical structure analysis in scientific papers. In *Proc. of ICDAR-91*, pages 311–319, Saint-Malo, France, Sept. 30–Oct. 2, 1991.

[13] G.L. DeSilva. Syntax-directed text segmentation in page layout analysis. Master's thesis, State University of New York at Buffalo, 1993.

[14] W. Doster. Different states of a document's content on the way from the Gutenbergian world to the electronic world. In *Proc. of 7th ICPR*, pages 872–874, 1984.

[15] L.D. Erman, F. Hayes-Roth, V.R. Lesser, and D.R. Reddy. The HEARSAY-II speech understanding system: Integrating knowledge to resolve uncertainty. *ACM Computing Surveys*, 12(2):213–253, June 1980.

[16] F. Esposito, D. Malerba, G. Semeraro, E. Annese, and G. Scafuro. An experimental page layout recognition system for office document automatic classification: An integrated approach for inductive generalization. In *Proc. of 10th International Conference on Pattern Recognition*, volume 1, pages 557–562, Atlantic City, NJ, June 16–21, 1990.

[17] F. Esposito, G. Semeraro, B. Giangregorio, G. Scafuro, and G. Ciardiello. An office document automatic classification system based on the page layout conceptual representation. In *Proc. of 1989 AICA Annual Conference*, Trieste, Italy, Oct. 4–6, 1989.

[18] J.L. Fisher. Logical structure descriptions of segmented document images. In *Proc. of ICDAR-91*, pages 302–310, Saint-Malo, France, Sept. 30–Oct. 2, 1991.

[19] J.L. Fisher, S.C. Hinds, and D.P. D'Amato. A rule-based system for document image segmentation. In *Proc. of 10th International Conference on Pattern Recognition*, volume 1, pages 567–572, Atlantic City, NJ, June 16–21, 1990.

[20] H. Fujihara and A. Mukerjee. Document recognition using qualitative reasoning. In *Proc. of Symposium on Document Analysis and Information Retrieval*, pages 316–331, Las Vegas, NV, March 16–18, 1992.

[21] M.L. Gibson. *Editing in the Electronic Era.* The Iowa State University Press, Ames, Iowa, 2nd edition, 1984.

[22] A.R. Hanson and E.M. Riseman. VISIONS: A computer system for interpreting scenes. In A.R. Hanson and E.M. Riseman, editors, *Computer Vision Systems*, pages 303–333. Academic Press, New York, 1978.

[23] B. Hayes-Roth. A blackboard architecture for control. *Artificial Intelligence*, 26:251–321, July 1985.

[24] A. Hurlburt. *The Grid: A modular system for the design and production of newspapers, magazines and books.* Van Nostrand Reinhold Company, New York, 1978.

[25] A. Hutt. *Newspaper Design.* Oxford University Press, London, 2nd edition, 1967.

[26] R. Ingold and D. Armangil. A top-down document analysis method for logical structure recognition. In *Proc. of ICDAR-91*, pages 41–49, Saint-Malo, France, Sept. 30–Oct. 2, 1991.

[27] Text ISO 8613, Information Processing and Office Systems. *Office Document Architecture (ODA) and Interchange Format (ODIF).* Geneva, 1989.

[28] Text ISO 8879, Information Processing and Office Systems. *Standard Generalized Markup Language (SGML).* Geneva, 1986.

[29] J.H. Kim, D.W. Payton, and K.E. Olin. A knowledge-based approach for object recognition in natural scenes. Technical report, Artificial Intelligence Center, Hughes Aircraft Company, Calabasas, CA, July 1985.

[30] K. Kubota, O. Iwaki, and H. Arakawa. Document understanding system. In *Proc. of 7th International Conference on Pattern Recognition*, pages 612–614, Montreal, Canada, July 30–Aug. 2, 1984.

[31] S.W. Lam and S.N. Srihari. Multi-domain document layout understanding. In *Proc. of ICDAR-91*, pages 112–120, Saint-Malo, France, Sept. 30–Oct. 2, 1991.

[32] L. Lamport. *LaTeX: A Document Preparation System.* Addison-Wesley, 1986.

[33] M. Landy, Y. Cohen, and G. Sperling. HIPS: A unix-based image processing system. *Computer Vision, Graphics, and Image Processing*, 25:331–347, 1984.

[34] J.C. Latombe and .A. Lux. Basic notions in knowledge representation and control for computer vision. In O.D. Faugeras, editor, *Fundamentals of Computer Vision: An Advanced Course*, pages 324–371. Cambridge University Press, 1983.

[35] K.A. Lemone. Document analysis using attribute grammars. In *Proc. of Symposium on Document Analysis and Information Retrieval*, pages 117–133, Las Vegas, NV, March 16–18, 1992.

[36] M.D. Levine. A knowledge-based computer vision system. In A.R. Hanson and E.M. Riseman, editors, *Computer Vision Systems*, pages 335–352. Academic Press, New York, 1978.

[37] M.D. Levine and S.I. Shaheen. A modular computer vision system for picture segmentation and interpretation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-3(5):540–556, September 1981.

[38] Y. Liu, R. Fenrich, and S.N. Srihari. An object attribute thresholding alogrithm for document image binarization. In *Proc. of ICDAR-93*, pages 278–281, Tsukuba Science City, Japan, Oct. 20–22, 1993.

[39] Q. Luo, T. Watanabe, and N. Sugie. A structure recognition method for japanese newspapers. In *Proc. of Symposium on Document Analysis and Information Retrieval*, pages 217–234, Las Vegas, NV, March 16–18, 1992.

[40] T. Matsuyama. Knowledge organization and control structure in image understanding. In *Proc. of 7th International Conference on Pattern Recognition*, volume 2, pages 1118–1127, Montreal, Canada, July 30–Aug. 2, 1984.

[41] T. Matsuyama and V. Hwang. SIGMA: A framework for image understanding - integration of bottom-up and top-down analysis. In *Proceedings of the 9th IJCAI*, volume 2, pages 908–915, Los Angeles, CA, August 1985.

[42] T. Matsuyama, V.S. Hwang, and L.S. Davis. Evidence accumulation for spatial reasoning. Technical Report CAR-Tech. Rep.-54, CS-Tech. Rep.-1381, Center for Automation Research, University of Maryland, College Park, MD, March 1984.

[43] D.M. McKeown and J.L. Denlinger. Map-guided feature extraction from aerial imagery. In *Proc. of the Workshop on Computer Vision: Representation and Control*, pages 205–213, Annapolis, MD, April 30–May 2, 1984.

[44] D.M. McKeown, W.A. Harvey, and J. McDermott. Rule-based interpretation of aerial imagery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-7(5):570–585, September 1985.

[45] M. Minsky. A framework for representing knowledge. In P. Winston, editor, *The Psychology of Computer Vision*. McGraw-Hill, New York, 1975.

[46] M. Mizuno, T. Kamimura, T. Tanaka, H. Tanaka, T. Ide, J. Tsukumo, and T. Temma. Development of a japanese document recognition system. In *Proc. of 2nd Annual Symposium on Document Analysis and Information Retrieval*, pages 233–242, Las Vegas, NV, April 26–28, 1993.

[47] G. Nagy. What does a machine need to know to read a document? In *Proc. of Symposium on Document Analysis and Information Retrieval*, pages 1–10, Las Vegas, NV, March 16–18, 1992.

[48] G. Nagy, S.C. Seth, and S.D. Stoddard. Document analysis with an expert system. In *Proc. of Pattern Recognition in Practice II*, Amsterdam, June 19–21, 1985.

[49] A.M. Nazif and M.D. Levine. Low level image segmentation: An expert system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(5):555–577, September 1984.

[50] R.P. Nelson. *Publication Design*. Wm. C. Brown Company Publishers, Dubuque, Iowa, 1972.

[51] J. Nielsen. *Hypertext and Hypermedia.* Academic Press, San Diego, 1990.

[52] H. P. Nii. Blackboard systems [Part One]: The blackboard model of problem solving and the evolution of blackboard architectures. *AI Magazine*, 7(2):38–53, Summer 1986.

[53] H.P. Nii. Blackboard systems [Part Two]: Blackboard application systems, blackboard systems from a knowledge engineering perspective. *AI Magazine*, 7(3):82–106, August 1986.

[54] D. Niyogi and S.N. Srihari. Expert systems and pattern recognition. In P.V.S. Rao and P. Sadanandan, editors, *Modern Trends in Information Technology.* Tata McGraw-Hill, Bombay. India, 1988.

[55] D. Niyogi and S.N. Srihari. A rule-based system for document understanding. In *Proceedings of AAAI-86*, volume 2, pages 789–793, Philadelphia, PA, August 15–22, 1986.

[56] D. Niyogi and S.N. Srihari. A knowledge-based expert system for document image analysis. In *Proc. of the 1986 Conference on Intelligent Systems and Machines*, Rochester, MI, March 28-30, 1986.

[57] D. Niyogi and S.N. Srihari. Using a blackboard architecture for control in a knowledge-based document understanding system. In *Proc. of SPIE : Applications of Artificial Intelligence V*, volume 786, pages 319–326, Orlando, FL, May 18–20, 1987.

[58] D. Niyogi and S.N. Srihari. Document understanding using a knowledge based methodology. In *Proc. of 1986 Expert Systems in Government Conference*, McLean, VA, October 1986.

[59] M. Numao and M. Ishizuka. A frame-like knowledge representation system for computer vision. In *Proc. of 7th International Conference on Pattern Recognition*, volume 2, pages 1128–1130, Montreal, Canada, July 30–Aug. 2, 1984.

[60] L. O'Gorman. The document spectrum for page layout analysis. In *IAPR International Workshop on Structural and Syntactic Pattern Recognition*, Bern, Switzerland, August 1992.

[61] A.J. Peels, N.J. Jansen, and W. Nawijn. Document architecture and text formatting. *ACM Transactions on Office Information Systems*, 3(4):347–369, October 1985.

[62] G.B. Porter and E.V. Rainero. Document reconstruction: A system for recovering document structure from layout. In *Proceedings of the Electronic Publishing Conference*, Lausanne, Switzerland, 1992.

[63] G. Reynolds, N. Irwin, A. Hanson, and E. Riseman. Hierarchical knowledge-directed object extraction using a combined region and line representation. In *Proc. of the Workshop on Computer Vision: Representation and Control*, pages 238–247, Annapolis, MD, April 30–May 2, 1984.

[64] D.M. Russell. Where do I look now? : Modeling and inferring object locations by constraints. In *Proc. of 1979 IEEE Computer Society Conf. on Pattern Recognition and Image Processing*, pages 175–183, Chicago, IL, August 6–8, 1979.

[65] L.G. Shapiro. The role of AI in computer vision. In *Proc. of 2nd IEEE Computer Society Conference on Artificial Intelligence Applications*, pages 76–81, December 1985.

[66] K.R. Sloan and R. Bajcsy. World model driven recognition of outdoor scenes. Technical Report TR-40, Comp. and Info. Sci. Dept., Univ. of Pennsylvania, Philadelphia, PA, September 1979.

[67] R.K. Srihari. PICTION: A system that uses captions to label human faces in newspaper photographs. In *Proc. of AAAI-91*, pages 80–85, Anaheim, CA, 1991.

[68] S.N. Srihari. From pixels to paragraphs: the use of models in text recognition. In *Proc. of 2nd Annual Symposium on Document Analysis and Information Retrieval*, pages 47–64, Las Vegas, NV, April 26–28, 1993.

[69] S.N. Srihari. Document image understanding. In *Proc. of ACM-IEEE Computer Society 1986 Fall Joint Computer Conference*, pages 87–96, Dallas, TX, November 2–6, 1986.

[70] S.N. Srihari and V. Govindaraju. Textual image analysis using the hough transform. *International Journal of Machine Vision and Applications*, 2(3):141–153, 1989.

[71] S.N. Srihari, J.J. Hull, P.W. Palumbo, D. Niyogi, and C.H. Wang. Address recognition techniques in mail sorting: Research directions. Technical Report Tech. Report 85-09, Dept. of Computer Science, SUNY at Buffalo, Buffalo, NY, August 1985.

[72] Y.Y. Tang, C.Y. Suen, C.D. Yan, and M. Cheriet. Document analysis and understanding: a brief survey. In *Proc. of ICDAR-91*, pages 17–31, Saint-Malo, France, Sept. 30–Oct. 2, 1991.

[73] S.L. Tanimoto. Paradigms for control of vision using inference in networks. In *Proc. of the Workshop on Computer Vision: Representation and Control*, pages 3–13, Rindge, NH, August 23–25, 1982.

[74] S.L. Taylor, M. Lipshutz, and C. Weir. Document structure interpretation by integrating multiple knowledge sources. In *Proc. of Symposium on Document Analysis and Information Retrieval*, pages 58–76, Las Vegas, NV, March 16–18, 1992.

[75] S. Tsujimoto and H. Asada. Understanding multi-articled documents. In *Proc. of 10th International Conference on Pattern Recognition*, volume 1, pages 551–556, Atlantic City, NJ, June 16–21, 1990.

[76] L.W. Tucker. Control strategy for an expert vision system using quadtree refinement. In *Proc. of the Workshop on Computer Vision: Representation and Control*, pages 214–218, Annapolis, MD, April 30–May 2, 1984.

[77] C.H. Wang and S.N. Srihari. Object recognition in structured and random environments: Locating address blocks on mail pieces. In *Proc. of AAAI-86*, volume 2, Philadelphia, PA, August 15–22, 1986.

[78] D. Wang and S.N. Srihari. Classification of newspaper image blocks using texture analysis. *Computer Vision, Graphics, and Image Processing*, 47:327–352, 1989.

[79] T. Watanabe, Q. Luo, and T. Fukumura. A framework of layout recognition for document understanding. In *Proc. of Symposium on Document Analysis and Information Retrieval*, pages 77–95, Las Vegas, NV, March 16–18, 1992.

[80] L. Wesley and A. Hanson. The use of an evidential-based model for representing knowledge and reasoning about images in the Visions system. In *Proc. of the Workshop on Computer Vision : Representation and Control*, pages 14–25, Rindge, NH, Aug. 23–25, 1982.

[81] T.E. Weymouth, J.S. Griffith, A.R. Hanson, and E.M. Riseman. Rule-based strategies for image interpretation. In *Proceedings of AAAI-83*, Washington, DC, August 1983.

[82] J.V. White. *Editing By Design*. R.R. Bowker Company, New York, 2nd edition, 1982.

[83] K.Y. Wong, R.G. Casey, and F.M. Wahl. Document analysis system. In *Proc. of 6th International Conference on Pattern Recognition*, Munich, W.Germany, Oct. 19–22, 1982.

[84] A. Yamashita, T. Amano, H. Takahashi, and K. Toyokawa. A model based layout understanding method for the document recognition system (DRS). In *Proc. of ICDAR-91*, pages 130–138, Saint-Malo, France, Sept. 30–Oct. 2, 1991.