

# PERFORMANCE EVALUATION OF DOCUMENT LAYOUT ANALYSIS

THESIS SUBMITTED IN ACCORDANCE WITH THE REQUIREMENTS  
OF THE UNIVERSITY OF LIVERPOOL FOR THE DEGREE OF  
DOCTOR IN PHILOSOPHY BY DAVID PAUL BRIDSON.

October 2009



# Contents

<b>Abstract</b>	<b>xiii</b>
<b>Declaration</b>	<b>xv</b>
<b>Copyright</b>	<b>xvii</b>
<b>Acknowledgements</b>	<b>xix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Overview . . . . .	1
1.2 Background . . . . .	2
1.2.1 Documents . . . . .	2
1.2.2 Digital documents . . . . .	3
1.2.3 Conversion to digital form . . . . .	4
1.2.4 Document Image Analysis . . . . .	5
1.2.5 Document Layout Analysis . . . . .	7
1.3 Research objectives . . . . .	8
1.4 Key contributions of the research . . . . .	9
1.5 Structure of thesis . . . . .	10
<b>2 Document Layout Analysis</b>	<b>11</b>
2.1 Overview . . . . .	11
2.2 Importance of Document Layout Analysis . . . . .	11
2.2.1 Page Segmentation . . . . .	11
2.2.2 Region Classification . . . . .	12
2.2.3 Reading Order Detection . . . . .	13
2.3 Problem characteristics . . . . .	14
2.3.1 Layout complexity . . . . .	14

2.3.2	General conditions affecting Layout Analysis . . . . .	15
2.3.3	Historical documents . . . . .	19
2.4	Layout analysis to date . . . . .	19
2.4.1	Top-down approaches . . . . .	19
2.4.2	Bottom-up approaches . . . . .	23
2.4.3	Hybrid approaches . . . . .	25
2.5	Maturity of the area . . . . .	29
2.6	Summary . . . . .	29
<b>3</b>	<b>Performance Evaluation</b>	<b>31</b>
3.1	Overview . . . . .	31
3.2	Performance Evaluation . . . . .	31
3.3	Performance Evaluation in Document Layout Analysis . . . . .	32
3.3.1	Desirable characteristics . . . . .	33
3.4	Review of Performance Evaluation methods . . . . .	34
3.4.1	Overview . . . . .	34
3.4.2	OCR output-based Performance Evaluation . . . . .	35
3.4.3	Pixel-based performance evaluation . . . . .	37
3.4.4	Geometric Performance Evaluation methods . . . . .	44
3.5	ICDAR Page Segmentation Competitions . . . . .	48
3.6	Discussion . . . . .	50
3.7	Summary . . . . .	51
<b>4</b>	<b>Ground-truth and Datasets</b>	<b>53</b>
4.1	Overview . . . . .	53
4.2	Ground-truth . . . . .	53
4.2.1	Definition . . . . .	53
4.2.2	Desirable characteristics . . . . .	54
4.3	Desirable criteria of datasets . . . . .	55
4.3.1	Representativeness of included document types . . . . .	55
4.3.2	Complexity . . . . .	55
4.3.3	Use of synthetic ground-truths . . . . .	56
4.3.4	Representativeness of document features . . . . .	56
4.4	Datasets . . . . .	57
4.4.1	Nartker, Rice & Lumos . . . . .	57
4.4.2	Phillips, Chen, Ha & Haralick . . . . .	57

4.4.3	Sauvola & Kauniskangas . . . . .	58
4.4.4	U.S. National Library of Medicine . . . . .	59
4.4.5	Suzuki, Uchida & Nomura . . . . .	59
4.4.6	Todoran, Worring & Smeulders . . . . .	60
4.4.7	Antonacopoulos, Karatzas & Bridson . . . . .	60
4.5	Discussion . . . . .	62
4.6	Summary . . . . .	66
<b>5</b>	<b>Interval Comparison</b>	<b>69</b>
5.1	Overview . . . . .	69
5.2	Introduction . . . . .	69
5.2.1	Examples . . . . .	70
5.3	Region representation . . . . .	71
5.4	Polygon representation . . . . .	74
5.5	Algorithm . . . . .	75
5.5.1	The Region Interval representation . . . . .	75
5.5.2	Converting a polygon to Region Intervals . . . . .	77
5.5.3	Converting the array into a Region Interval representation . . . . .	82
5.5.4	Application to a complete document . . . . .	83
5.5.5	Comparison of ground-truth and segmentation . . . . .	86
5.6	Application to a real-world document . . . . .	92
5.7	Efficiency . . . . .	93
5.7.1	Space-efficiency . . . . .	95
5.7.2	Time-efficiency . . . . .	96
5.8	Discussion . . . . .	98
<b>6</b>	<b>Evaluation System</b>	<b>101</b>
6.1	Overview . . . . .	101
6.2	Identification of errors . . . . .	101
6.2.1	Error types . . . . .	101
6.2.2	Identification of region correspondences . . . . .	103
6.2.3	Severity of errors . . . . .	105
6.3	Problems with the region-only approach . . . . .	107
6.3.1	Possible solutions . . . . .	108
6.4	Error quantification . . . . .	110
6.4.1	Area-weighting . . . . .	110

6.4.2	Application scenarios . . . . .	111
6.4.3	Pre-defined application scenarios . . . . .	113
6.5	Presentation of results . . . . .	114
6.5.1	Implementation . . . . .	115
6.5.2	Detailed error description . . . . .	116
6.5.3	Detailed region description . . . . .	117
6.6	Sample output . . . . .	117
6.7	Discussion . . . . .	119
6.8	Summary . . . . .	119
<b>7</b>	<b>Evaluation</b>	<b>123</b>
7.1	Overview . . . . .	123
7.2	Introduction . . . . .	123
7.3	Accuracy & Applicability . . . . .	124
7.3.1	Testing using artificial data . . . . .	125
7.3.2	Testing pixel-accuracy and simple errors . . . . .	125
7.3.3	Manual verification using real-world data . . . . .	127
7.3.4	Evaluation against competition results . . . . .	129
7.4	Flexibility . . . . .	130
7.5	Descriptiveness . . . . .	132
7.5.1	Error type overview . . . . .	132
7.5.2	Detailed region description . . . . .	133
7.5.3	Detailed error description . . . . .	133
7.6	Efficiency . . . . .	135
7.7	Analysis of competition results . . . . .	136
7.8	Summary . . . . .	138
<b>8</b>	<b>Conclusion</b>	<b>139</b>
8.1	Overview . . . . .	139
8.2	Review of goals . . . . .	139
8.2.1	Accuracy . . . . .	139
8.2.2	Applicability . . . . .	140
8.2.3	Flexibility . . . . .	140
8.2.4	Descriptiveness . . . . .	141
8.2.5	Efficiency . . . . .	141
8.3	Future work . . . . .	141

8.3.1	Web interface . . . . .	141
8.3.2	Adaptation to other document types . . . . .	142
8.3.3	Definition of further application scenarios . . . . .	142
8.4	Summary . . . . .	143
<b>Bibliography</b>		<b>145</b>
<b>A</b>	<b>Published paper on the PRImA Dataset</b>	<b>149</b>
<b>B</b>	<b>The PRImA Document Layout XML Format</b>	<b>161</b>
<b>C</b>	<b>Published paper on the ICDAR 2005 Competition</b>	<b>169</b>
<b>D</b>	<b>Published paper on the ICDAR 2007 Competition</b>	<b>175</b>
<b>E</b>	<b>Published paper on the evaluation method</b>	<b>181</b>





# List of Figures

1.1	An image of a real-world magazine page containing a complex layout, taken from the PRImA Layout Analysis Dataset (mp00167). . . . .	6
2.1	Two adjacent columns of text from a document page. . . . .	12
2.2	a) The simple layout of a fiction book and b) a relatively complex layout from a magazine. . . . .	14
2.3	An example magazine page containing non-rectangular regions. . . . .	16
2.4	A scanned bi-level document image with noise highlighted. . . . .	16
2.5	a) A side view of a book being scanned and b) the scanned image illustrating the effects of shear. . . . .	18
2.6	From left to right, an example document image of the title page of a journal article, the horizontal-projection profile calculated by counting the black pixels on each row and an example thresholding of the projection profile with the structure labelled. . . . .	21
3.1	The region correspondence graph which shows the ground-truth regions and their overlapping segmentations, and vice versa. The edge labels show the number of pixels involved in an overlap. . . . .	43
3.2	The maximal surrounding rectangle. . . . .	46
3.3	Example maximal, segmentation and ground-truth left boundaries. . . . .	47
3.4	The overall results from the ICDAR 2003, 2005 and 2007 page segmentation competitions. . . . .	49
4.1	The colour image and associated XML ground-truth file of magazine page 39 from the PRImA Document Layout Dataset. . . . .	65
5.1	The ground-truth and segmentation layouts used in examples in this chapter to illustrate the approach. . . . .	70

5.2	An image of a real-world magazine page containing a complex layout, taken from the PRImA Layout Analysis Dataset (mp00167). . . . .	72
5.3	A rectangular region shown (a) without skew, (b) with 0.5 degrees of skew and (c) with 5 degrees of skew, all with bounding boxes overlaid. . . . .	73
5.4	a) A large book lying on a flatbed scanner, and (b) an example of the optical distortion produced by this, known as shear. . . . .	73
5.5	A document with a more complex layout with isothetic polygon region outlines overlaid. . . . .	74
5.6	a) A polygon region outline, and b) the same region shown in the region interval representation. . . . .	75
5.7	a) A polygon region outline, b) the set of pixels “inside” that polygon and c) the equivalent region interval representation. . . . .	76
5.8	A polygon with three points: one inside the polygon ( $p_1$ ), one outside ( $p_2$ ) and a reference point which is also outside the polygon ( $r$ ). . . . .	77
5.9	a) A polygon region outline; b) the array with the outline of the polygon marked as inside; c) a point selected for checking along with the reference point $r$ and a line drawn between the two; d) the inside of the region marked as inside. . . . .	79
5.10	An individual edge from the region outline and the rasterized version of it. . . . .	80
5.11	Pseudo-code for converting an arbitrary polygon to a two-dimensional array. . . . .	81
5.12	a) An array showing the pixels inside and outside of a region, and b) the related region interval containing a region entrance, an exit and a combined (1-pixel) entrance and exit. . . . .	82
5.13	Pseudo-code for converting the two-dimensional array into a region interval. . . . .	84
5.14	The example document layout introduced at the beginning of this chapter and its region interval equivalent. . . . .	85
5.15	The example ground-truth and segmentation polygon layouts and their region interval equivalents; the band structures from each being used to form a combined band structure. . . . .	87
5.16	Overlapping ground-truth and segmentation bands from the example document layouts and the combined interval created from them. . . . .	90
5.17	The excerpted combined interval from the previous diagram used to detect overlaps between the ground-truth & segmentation. . . . .	91
5.18	The operation of the method on a real-world document image, magazine page 42. . . . .	94

6.1	A portion of an image from the PRImA Dataset with the ground-truth and a segmentation from the ICDAR 2005 Page Segmentation competition overlaid. . . . .	107
6.2	An artificial example document image, ground-truth layout, segmentation layout and the overlaps between them. . . . .	118
6.3	A sample of the region description output of the system depicting a portion of the page containing a column of text which has been merged, detected by the system as a series of allowable merges. . . . .	121
7.1	a) An example ground-truth, b) the geometric description derived from it and c) a sample segmentation of the image. . . . .	128
7.2	From the ICDAR 2007 Competition dataset, a) an example ground-truth, b) the geometric description derived from it and c) a sample segmentation of the image. . . . .	128
7.3	a) The published results of the ICDAR 2005 Page Segmentation Competition, and b) results on the same data with the new system. . . . .	129
7.4	The results from ICDAR 2005 evaluated using a) the general layout analysis scenario, and b) the indexing scenario. . . . .	131
7.5	The categories of errors which contributed most to each method's score. . .	133
7.6	The overall errors detected for each method and divided into each of the categories of error measured. . . . .	136



# Abstract

Digital documents have many advantages over their analogue equivalents. However, a significant proportion of documents were not created in digital form. In order to obtain the advantages of digital documents for existing analogue documents, it is highly desirable to be able to convert them into digital form.

An important part of the process of digitisation is detecting the layout of the document to be recognised. Failure to do this correctly has negative consequences for subsequent parts of the recognition process.

In order to spur the development of layout analysis methods, it is desirable to have a common evaluation method which can be used to evaluate the results of layout analysis on complex documents. However, previous approaches to the problem have issues when dealing with more difficult document types, such as those containing colour or complex region shapes.

This thesis presents a new approach to performance evaluation of layout analysis methods which is based on a hybrid region-based and pixel-based approach which allows an accurate evaluation to be made on complex, modern documents, whatever the colour of the contents.

The approach provides significant flexibility in allowing evaluations tailored to specific application areas and in increasing the amount of information produced in the evaluation. This information is designed to be useful in aiding developers to improve their methods.



# Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institution of learning.





# Copyright

Copyright in the text of this thesis rests with the author. The author hopes that you will find it useful and will exercise prudence in making copies of it.



# Acknowledgements

I would like to thank my supervisor, Dr. Apostolos Antonacopoulos, for his invaluable guidance, my second supervisor, Prof. Trevor Bench-Capon, and advisor, Dr. Bernard Diaz, for their support and feedback on the thesis and my parents and friends for their support and encouragement during my research and the preparation of this thesis.



# Chapter 1

## Introduction

### 1.1 Overview

There are millions of documents which exist solely in analogue form. Many of the advantages of digital documents cannot be applied to these documents. So, in recent years, there has been a significant effort aimed at converting these documents into digital form.

One of the main challenges in digitising these documents is to find the layout of the document before recognising the contents. Over the past few decades, a number of approaches have been developed which are designed to recognise the layouts of general documents.

A common feature of many of the methods introduced so far has been that they have been tested on individual application-specific datasets and using individual testing methods. This means that the results presented for different methods are difficult to compare.

A number of evaluation methods have been presented to date which aim to provide a common method for evaluation. However, the approaches proposed to date have significant room for improvement in the areas of efficiency, accuracy and flexibility.

This thesis presents research into a new performance evaluation method for layout analysis systems which allows an accurate evaluation to be made on complex modern documents. This is specifically tailored to be flexible in dealing with the various uses for layout analysis systems. It is designed not only to provide a high-level evaluation but also to provide in-depth information which may be used by developers to improve their layout analysis systems.

## 1.2 Background

### 1.2.1 Documents

For a large portion of human history, it has been necessary to keep written records in the form of documents. Some of the earliest documents which are identifiable as writing in the modern sense appeared in ancient Mesopotamia around the year 2600 B.C. in the form of cuneiform writing on clay tablets and stelæ. From the earliest times, documents were used for ceremonial purposes, for sending letters and for more mundane purposes such as shopping lists.

Since documents were created manually by etching in clay, stone or wood, there were no practical means of automated reproduction. Where a copy of such a document was required, it would be necessary to manually produce a copy in the same way that the original was made.

The mass production of printed documents did not become feasible until the invention of movable type. This was first invented in 1040 A.D. by the Chinese inventor Bi Sheng (畢昇) who produced a system of movable clay characters which could be arranged in a frame then used to print multiple copies of a document. Although this first attempt proved to be fragile, the idea was later improved upon by the Korean Chae Yun-ui (채윤의) who implemented the first metal movable type system in the early 13th century.

Johannes Gutenberg independently invented a metal movable type system in Germany in the mid-15th century. His invention proved to be more successful than previous attempts and was quickly being used to mass-produce printed documents. Among the most famous of these is the Gutenberg Bible, which consists of 1,282 pages in two volumes; in all, approximately 180 copies of the books were produced.

Prior to Gutenberg's invention, books would typically have been copied manually, meaning that copies would be available only to those who could afford to have a copyist produce a new copy of the book or produce a copy themselves. The printing press brought about a revolution in duplication. It became possible to produce large numbers of copies of books and have them widely distributed. This, in turn, had a significant impact on literacy throughout Europe.

Gutenberg's original invention proved to be durable. The concept of metal movable type survived for five centuries, with new technological advances periodically being introduced. In 1812, the first steam-powered press was introduced which greatly increased the speed of printing and, thus, the number of copies which could be produced. The rotary printing press followed in the middle of the 19th century and this contributed to a further increase in production.

Over the past five millennia of documents, and the past five centuries of printed documents, the collected knowledge of mankind has been stored in documents. The amount of information stored in documents is enormous; large proportions of these are held in the world's libraries. One common measure used by libraries to measure the size of their holdings is the number of miles of shelf space required to house them. For instance, the Library of Congress, the United States' national library, is estimated to contain about 530 shelf miles of books, manuscripts and other documents. The British Library is estimated to contain 388 shelf miles of documents. Estimates of the number of books ever published range from 50 million to 200 million with another 100,000 new books being published each year.

### 1.2.2 Digital documents

The advent of computer technology in the latter half of the 20th century has brought many improvements which can be of significant value in archiving documents. The large and growing amount of digital storage available today allows millions of documents, which in physical form might occupy many miles of shelf space, to be stored in digital form in a relatively small physical space.

Digital technology also brings with it ease of duplication. While the mass copying of paper documents requires large expenditures in photocopying and prohibitive amounts of raw materials, making digital copies is often virtually cost-free, potentially allowing millions of copies to be made without significant cost. Moreover, while analogue documents are often subject to wear-and-tear through repeated use and copying, digital copies have no such disadvantage, allowing millions of perfect copies to be made from the same master without degradation to the original or the copies.

The increasing interconnectivity of computers also provides a significant advantage for archivists and the users of archives alike. In order to consult paper documents in archives, it is often necessary to travel to the archive or, in cases where this is permitted, to arrange for the documents to be shipped across the world. In either case, this is likely to be an expensive proposition. With digital copies, however, it is possible for them to be transported around the world near-instantaneously and virtually cost-free, potentially allowing large numbers of people around the world to consult a single document without significant expense.

The wider dissemination and ease of storage of digital documents has given rise to advances in indexing. Indexes for physical documents, such as card catalogues, tend to be quite narrow in scope, allowing only for limited per-document meta-data to be stored. Digital technology makes it possible not only to store meta-data about the document but also to index its

full contents. Search engines such as Google and Yahoo make it possible to search the contents of billions of documents instantaneously, returning the documents most relevant to the user's query.

It should be noted that, despite their significant advantages, digital documents do not provide a panacæa for the problem of archiving. While they suffer from few of the disadvantages of paper documents, they have other disadvantages which pose unique problems. The long-term storage qualities of paper documents are well-known — paper documents printed on suitable materials and stored safely in appropriate conditions may survive for many hundreds of years. The long-term storage capabilities of digital media are less well-known.

While some digital media may be suitable for long-term storage, it is difficult to estimate how long a particular medium will last without thorough testing. Indeed, it is not unknown for storage media which are sold as archival media to become unreadable after only a few months due to manufacturing error. So, at best, the lifespan of the media is quite uncertain.

The low cost of digital storage and ease of copying and distribution may bring further benefits from an archival standpoint. It becomes much easier and less expensive to have multiple copies of data stored in multiple locations. Such redundancy would improve the likelihood of works surviving. Were such measures available in ancient Egypt, they may have prevented the enormous loss of data which occurred during the destruction of the Royal Library of Alexandria.

### 1.2.3 Conversion to digital form

During the millennia which constituted the pre-digital age, many millions of documents were created and these are stored in libraries and archives around the world. However, since the vast majority exist in physical form only, many of the advantages available with digital documents may not be applied to them. While digital documents can be automatically searched using a search engine such as Google or Yahoo, paper documents cannot. Digital documents may be reproduced and widely distributed with minimal cost but analogue documents may be reproduced only with a significant amount of materials and effort.

In order to gain these advantages for the millions of extant physical documents, it is necessary first to convert them into digital form in a process known as Document Image Analysis. The following subsection gives an overview of this process.

The first project aiming to digitise the world's documents, Project Gutenberg[9], began in 1971. The goal of the project was to launch the world's first digital library, making available public domain books free of charge on the internet. The project began initially with volunteers retyping books. However, advances in OCR technology made it possible to digitise



books at a far faster rate. To date, the project has digitised and made available almost 30,000 public domain books, with a further 500 being added each month.

More recently, other projects have been started with even more ambitious goals. The Google Book Project[28] was started in 2004 with the goal of digitising all of the world's books. Partnering with 20 of the world's largest libraries, including the University of Oxford's Bodleian Library and Harvard University Library, as well as a large number of publishers, they are mass-digitising books at the rate of 1 million books per year at an estimated cost of \$5 million per year.

The Million Book Project[21] run by Carnegie Mellon University with partners in India and China had digitised over 1.5 million books before the end of 2007 with a further 7,000 being scanned per day. A consortium of large libraries and technology companies formed the Open Content Alliance which aims to create a freely-available library of public domain books. At present, they are scanning 12,000 books per month.

These mass digitisation projects are currently at an early stage with many books scanned but with significant work to be done to get the scanned documents into indexable, searchable text. Recognising a large number of documents in a variety of scripts and languages and of ages varying from new to centuries old, is still a significant research problem. The following subsection gives an overview of the processes involved.

#### 1.2.4 Document Image Analysis

The process of converting paper documents into digital versions is known as Document Image Analysis. This process may be divided into a number of steps:

- Digitisation
- Layout Analysis
- Recognition

##### Digitisation

The first stage, digitisation, is the initial capture of the paper document into a digital form. This is typically performed with a hardware device such as a scanner or digital camera which captures a digital image of the document. It should be noted that the image is a first crude representation of the document, being merely a picture of the document, without further understanding of the contents.

### Layout Analysis

Documents may contain a variety of different types of information. Most documents contain text of some sort, but some may also contain pictures, graphs, drawings and equations. Some documents may be entirely composed of a single type of information, while others may contain multiple different types of information on a single page. The computer must typically use a different method to recognise each different type of information. So, before any recognition is to take place, it is necessary to separate and identify the different types of information on a page. This part of the process is known as Document Layout Analysis. This is usually further split into three stages:

- Page Segmentation
- Region Classification
- Reading Order Detection

Take, for example, the document page in Figure 1.1. The page includes several regions of text — a headline at the top, a drop capital at the beginning of the first paragraph, several paragraphs of body text arranged into columns, and a highlighted quotation. The page includes a large graphic in the centre around which the body text wraps tightly. It also contains separators to divide the two columns.



Figure 1.1: An image of a real-world magazine page containing a complex layout, taken from the PRImA Layout Analysis Dataset (mp00167).

Dividing the page into its constituent regions is known as Page Segmentation. Once these regions have been segmented, it is necessary to identify the type of each region so that it might

be passed to the correct recognition method for the given region type. This is referred to as region classification.

For applications where the textual content of the page is to be repurposed, for example to provide a web version of a printed document, it is necessary to detect the correct ordering of the regions so that text is placed in the correct order. This part of the process is termed reading order detection.

## Recognition

Once the regions have been segmented and labelled, the final stage is to recognise the contents of each region. This involves converting the image of that region into a computer-editable, human-readable representation. The process used to do this is usually different for each different type of region.

For instance, the contents of text regions are usually passed to an Optical Character Recognition method which will convert an image of the text region into computer-editable text of the contents. Similarly, graphics regions may be passed to a graphics recognition method which will attempt to extract a vector graphic representing the graphics region. Image regions may be kept intact or may have further processing applied to identify the contents of the image.

OCR is a relatively mature research area with many commercial OCR products available which achieve excellent results for clean, modern documents in latin script. However, there is still considerable research into performing OCR on degraded historical documents or for non-latin scripts such as Arabic or Chinese.

### 1.2.5 Document Layout Analysis

Document Layout Analysis is the task of segmenting the different regions on the document page and finding the type of each of those regions. While this may initially seem like a simple task, it is difficult to perform automatically given the diverse range of document types and styles. Similarly, given the large number of documents which exist only in analogue form, the documents which it may be desirable to analyse may range from clearly printed, clean modern documents to highly-degraded historical documents.

It is crucial to the Document Image Analysis process to correctly segment the regions of the page before any further recognition may take place. Recognition methods typically operate on a single type of information. For example, an Optical Character Recognition method will only give meaningful results on a region of text. So, it is necessary to separate regions

of a given type from regions of other types in order to obtain meaningful output from the Document Image Analysis process.

Over the past two decades, the area has seen extensive research with a large number of new approaches published in the literature. These are summarised in chapter 2. However, one of the problems in the area is that new methods are rarely tested using common datasets and testing methods, which makes it difficult to form an opinion of the relative strengths of individual algorithms as well as the overall maturity of the research area.

In the scope of the International Conferences on Document Analysis and Recognition, Drs. Apostolos Antonacopoulos and Basilis Gatos have run a series of international competitions in the area of Document Layout Analysis in order to test modern layout analysis methods using a single dataset and testing methodology. The first, in ICDAR 2001, was devoted solely to newspaper page segmentation[8]; the three following competitions in ICDAR 2003[6], 2005[4] and 2007[5] expanded this to deal with more general documents such as technical articles and magazine pages.

The results of the competitions, discussed in greater detail in the next chapter, showed that even modern layout analysis methods had significant difficulties even when dealing with clean, modern documents. Thus, there remains significant room for improvement in the area.

### 1.3 Research objectives

In recent times, there has been a move towards developing separate evaluation methods for layout analysis which would allow dissimilar layout analysis methods to be evaluated using a common method and using a common dataset, providing for true comparability of results. Numerous performance evaluation metrics have been proposed in the literature and these are discussed in detail in chapter 3.

The methods proposed to date have several areas for improvement. Typically, such methods were designed several years ago when research in layout analysis was concentrated more on evaluating simpler documents such as journal articles. More general documents, however, often contain more complex features, such as irregularly-shaped regions, regions wrapping tightly around other regions, and significantly greater use of colour. Layout analysis methods have been proposed in the literature to deal with such features. This necessitates an improvement in performance evaluation methods to deal with such complex documents.

Previous methods typically focused mainly on benchmarking. That is, given a series of segmentations detected by a layout analysis method, they focused mainly on calculating either a global performance metric or a small number of metrics which represent performance

numerically. While such measures may be useful in benchmarking, they are less useful to developers of layout analysis methods who require information to allow them to improve their methods. In that situation, a more detailed analysis is necessary.

Document recognition, and document layout analysis, may be used in a wide variety of scenarios. Some applications may focus on indexing documents while others may focus on obtaining a full digital replicas of documents. Previous performance evaluation methods have focused on providing an all-purpose performance metric. However, the strength of a layout analysis method may depend greatly upon the scenario involved. So, it is desirable for a performance evaluation method to be able to provide an evaluation tailored to the specific application scenario.

## 1.4 Key contributions of the research

This thesis presents a new performance evaluation method based which improves upon previous approaches in several respects:

- **Accuracy** — The new approach is based on a region interval representation which is designed to provide an accurate performance evaluation for layout analysis methods.
- **Applicability** — This complex representation allows evaluations to be performed even for modern, complex documents which may not have been analysed by previous approaches.
- **Flexibility** — The new approach recognises that Document Image Analysis methods may be used for a wide variety of applications and that any evaluation must be customisable to allow evaluations based on the needs of end-users.
- **Descriptiveness** — The new approach is designed with a view not only for performance evaluation at a high level, as might be useful to end-users, but also to provide a more detailed evaluation which will allow developers of Layout Analysis methods to find the strengths & weaknesses of their methods in order to target their development efforts
- **Efficiency** — The approach presented here places a premium on efficiency. This is important to allow results to be obtained quickly and to allow evaluations to be expanded more readily to datasets of significant size.

## 1.5 Structure of thesis

The following chapter will present an overview of the process of Document Layout Analysis and a description of some of the more notable layout analysis methods. The third chapter will describe previous approaches to performance evaluation as it relates to layout analysis. Crucial to the process of performance evaluation is the availability of suitable ground-truth datasets; these are discussed in the fourth chapter. The fifth chapter will present a novel region comparison method which is used as the foundation of the performance evaluation method presented here. The sixth chapter will present the performance evaluation method as a whole while chapter seven will provide an in-depth evaluation of the method on a real-world dataset. A conclusion will be provided in chapter eight.

# Chapter 2

## Document Layout Analysis

### 2.1 Overview

The previous chapter introduced the background and topic of the thesis. This chapter will begin by explaining document layout analysis in greater detail: the importance of layout analysis to the document image analysis process, the challenges involved in correctly segmenting a document image, a summary of the approaches which have been published to date and a discussion of the maturity and open problems of the area.

### 2.2 Importance of Document Layout Analysis

Layout Analysis is a highly-important part of the document recognition process. In the previous chapter, the Layout Analysis process was divided into the following separate stages:

- Page Segmentation
- Region Classification
- Reading Order Detection

Taken separately, each of these stages is vital to the later stages of the Document Recognition process and so they are all extremely important. This section will discuss the importance of each of these stages separately.

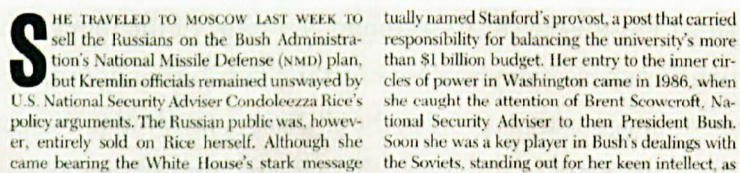
#### 2.2.1 Page Segmentation

Everyday paper documents are typically highly structured and attempting to recognise the contents of the document is likely to be impossible without first understanding its structure.

In a given document page, the logical relationship of different regions on the page is typically conveyed by the physical structure.

The physical structure is often governed by a series of rules which have evolved over the history of type and typesetting. For example, an image's caption is related to an image by being placed directly adjacent to it. If two paragraphs are vertically adjacent, then the reader will interpret them as following on from each other. If they are horizontally adjacent but slightly separated, they will be interpreted as belonging to different columns. If they are physically separated by a title, then they will be separated into logically distinct articles. In order to obtain a perfect digital representation of a document, it is first necessary to understand its structure.

In some applications, finding the document structure is particularly important. In a content indexing application, for example, it is vital to find the key items of data for indexing — particularly headings and image captions.



**S**HE TRAVELED TO MOSCOW LAST WEEK TO sell the Russians on the Bush Administration's National Missile Defense (NMD) plan, but Kremlin officials remained unswayed by U.S. National Security Adviser Condoleezza Rice's policy arguments. The Russian public was, however, entirely sold on Rice herself. Although she came bearing the White House's stark message

tually named Stanford's provost, a post that carried responsibility for balancing the university's more than \$1 billion budget. Her entry to the inner circles of power in Washington came in 1986, when she caught the attention of Brent Scowcroft, National Security Adviser to then President Bush. Soon she was a key player in Bush's dealings with the Soviets, standing out for her keen intellect, as

Figure 2.1: Two adjacent columns of text from a document page.

In addition to finding the structure of a document, segmenting the document into its constituent regions is also important for later stages of the Document Recognition process. Take, for example, Figure 2.1. This figure contains two regions of text which are logically part of the same article. However, if the article as a whole were to be passed to an Optical Character Recognition (OCR) method, then the recogniser would simply recognise the characters from the left of the image to the right, with text lines from the second column being appended to the adjacent text line of the first column. This would cause the text to be out of order. So, in order to ensure that the output from the recogniser is usable, it is necessary to separate the different regions of the page.

### 2.2.2 Region Classification

Documents often contain several different types of information. The processes by which a computer may recognise each different type of information are widely diverse and typically application-specific.

OCR methods are used to recognise text. If an OCR method is applied to an image which contains information other than text, such as a graphic, the results will typically be



non-sensical. In that instance, the OCR method will perhaps output some seemingly random characters. These will likely correspond to portions of the graphic which are *perceived* by the OCR method to be similar to text. However, the output will not be useful in gaining an understanding of the contents of the graphics region.

Given this, it is vitally important in ensuring a correct output from the document recognition process that document layout analysis is first applied to determine the types of region present in the page so that the correct recognition method may be applied in each case.

Further to this, it is important to separate different regions of different types from each other. For instance, it is quite right that a graphics region should be passed to a graphics recognition method. However, if neighbouring areas of text are passed to the graphics recognition method along with the graphic, then the results will not be an optimal recognition of the page. So, it is important to apply layout analysis to separate dissimilar regions from each other.

Given these substantial problems which would occur without performing a proper layout analysis of the page, it can be seen that the layout analysis step is a vital and important part of the document recognition process.

### 2.2.3 Reading Order Detection

Having already segmented the page into its constituent regions and classified each as being of a given type, the final stage of the Layout Analysis process is to detect the correct reading order of the page. This is particularly important in applications which involve repurposing the information from scanned documents.

Take, for example, an application involving scanning printed documents in order to make them available on the internet. For such applications, it may not be necessary to present the text in the exact layout as it appeared in the original printed document. Instead, the goal might be to re-present the text as a coherent article on the internet. If the reading order is incorrectly detected, then sections of the resulting document will appear out-of-order, rendering the digitised document incoherent.

Reading order detection may be performed by utilising knowledge of typical document structures and the historical typographical rules which form the basis of document structure. For example, a document may first be split into articles by utilising detected headings to separate articles within the document. The reading order within articles may then be detected by, in single-columned documents, simply ordering the regions from top-most to bottom-most. In multi-columned documents, columns would be ordered from left-most to right-most and text within each column from top-most to bottom-most, assuming typical conventions for

documents written in latin scripts.

### 2.3 Problem characteristics

The problem of correctly segmenting arbitrarily complex document pages is a difficult one which has not yet been solved and towards which a considerable amount of research is still being performed. This section describes some of the main features in document pages which render the task of layout analysis a difficult one.

#### 2.3.1 Layout complexity

Documents range widely in complexity from the extremely simple to the extremely complex. One example of a common document with an extremely simple layout is a fiction book — see Figure 2.2a. Such a book typically contains only one column of text with perhaps a header or footer containing page numbers and other information.

At the opposite extreme, an example of a complex document would be a magazine page — see Figure 2.2b — which may contain complex features such as text overlapping images, non-rectangular regions, text wrapping tightly around images, etc.



Figure 2.2: a) The simple layout of a fiction book and b) a relatively complex layout from a magazine.

### Varying region types

Often it is true that pages containing just one type of information, for example containing solely text, are significantly easier to segment. When a page contains only text, then the regions on the page have similar features which make it easier to find the region boundaries. In a purely textual page, the size of the connected components on the page will likely be fairly similar in all the regions and, using some knowledge of the typical structure of text and the regular feature size, it may be simple to identify the gaps between regions.

However, when a page contains multiple types of information, e.g. text and images or text and graphics, the differences between the different types of regions may make it more difficult to segment the regions. For instance, in a purely textual page, the gap between adjacent columns may be detectable by using some multiple of the typical inter-word space as a threshold. However, the presence of image regions — which typically have fewer but larger connected components — would significantly increase the complexity of such a stage.

### Complex layouts

Many of the earliest layout analysis methods often used global measures such as vertical- and horizontal-projection profiles to segment regions.<sup>1</sup> These were useful in detecting layouts in relatively simple documents which contained only rectangular regions.

However, documents are typically laid out by humans. This means that, rather than being structured using some fixed rules, documents often have highly irregular and individual layouts. Many magazine pages, such as the example image in Figure 2.3, contain non-rectangular image regions around which the text wraps closely. The global features mentioned in the previous paragraph are not useful for such images, and more complex methods must be used in order to correctly segment such pages.

## 2.3.2 General conditions affecting Layout Analysis

The previous subsection mentioned a number of layout-specific features which cause problems during layout analysis. However, there are also a number of features which are commonly found in document images which cause problems not only with layout analysis but also with other stages of the document recognition process. These are described below.

---

<sup>1</sup>Given a bi-level (black & white) image, a horizontal-projection profile is a histogram, containing one entry for each row in the image, with the value of each entry equal to the number of black pixels in the given row. As such, the plot will contain peaks on rows with many black pixels and troughs on rows with few or no black pixels. The vertical-projection profile is based upon the same concept but using columns instead of rows.



Figure 2.3: An example magazine page containing non-rectangular regions.

### Noise

Noise is a common feature in many fields, such as signal processing. Noise, in document imaging, is any feature of the recorded image which is not present in the original document. This may appear for several reasons. It may be caused by electronic noise in the scanner or digital camera sensor used for capturing the document. Likewise, it may be an artefact introduced at some stage of the document processing — for example, during binarisation.

Noise in the document image typically takes the form of single-pixel variations in colour. In a colour or grayscale image, the noise may make the pixel darker or lighter or, in a binarised image, this may cause the pixel involved to be black where it would have been white and where its neighbouring pixels are white, or the reverse. Figure 2.4 shows the effects of noise in a binarised document image. The black pixels corresponding to actual document contents have been faded to gray, leaving the pixels corresponding to noise in black.



Figure 2.4: A scanned bi-level document image with noise highlighted.

There are a large number of techniques designed to filter noise from an image. The optimal noise reduction technique for different images may be different — there is no single optimal noise reduction technique. The application of noise reduction may also cause undesired artefacts to appear in the image. Ideally, the effect of any given noise reduction technique

should be to cause fewer problems to the document recognition process than the original noise would have; otherwise, the usefulness of the noise reduction technique is questionable. Similarly, many layout analysis and recognition techniques include features designed to make them robust in the presence of noise or may specify recommended noise filtering techniques.

### Skew

When placing a document on the scanner bed or in position for being photographed, the document is rarely placed at exactly the correct orientation for capture. The deviation between the correct angle and the captured angle is known as *skew*.

In an ideal document image, the document would be positioned exactly so that the horizontal edge of the document is exactly horizontal in the image and the vertical edge of the document is exactly vertical in the image. However, in real-world document scans, this is rarely true. In virtually all captured documents, there will be some small degree of skew. In a minority of cases, there will be a more significant degree of skew in the image. The likelihood and extent of skew in a captured image may depend on the skill of the operator. In applications designed for use by the general public, a degree of robustness in the presence of skew is desirable.

### Shear

Another problem of document image analysis occurs during the capture of large books. For large books, it is typically difficult to fully open the book for capture using a flatbed scanner or digital camera. Portions of the page nearest to the spine of a large book naturally curve towards the spine when opened out, making it impossible to fully flatten the pages before scanning.

In some applications where less valuable books are involved, one solution to this problem has been simply to physically cut the book pages from the spine. This allows individual pages to be removed from the book and scanned fully flat. However, for many books, such a destructive approach is not feasible. For rare books of which only a few copies may remain — for example, the Gutenberg bibles discussed in the introduction — it is unlikely that the owner would permit such a destructive operation. The same may be true for books stored in libraries which must still be available for use after digitisation.

When intact books are scanned, the curvature of the pages towards the spine produces an artefact known as shear. The portions of the page nearest the spine will be further away from the scanner bed. This causes several effects in the scanned page which are illustrated in Figure 2.5.

The page appears curved, with the portions nearest the outer edge of the book least curved and a more dramatic curve on the edge nearest the spine. This causes problems at several stages of the image analysis process. Methods attempting to detect regions and text lines in documents containing such distortions will not work well if they make the assumption that the text lines contained within are straight. Typically, such documents will contain significant curvature which complicates the process of text line and region detection.

Since the portions of the page nearer the spine are further away from the scan-head, this causes these portions to be less well illuminated appearing slightly darker than the rest of the page, and potentially out of focus compared to the rest of the page.

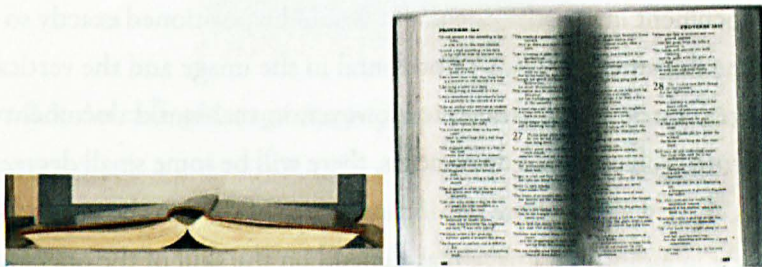


Figure 2.5: a) A side view of a book being scanned and b) the scanned image illustrating the effects of shear.

### Irregular illumination

Documents captured using a flat-bed scanner are typically well and evenly illuminated, except where shear is present, as discussed in the previous subsection. For some documents the best method for capturing an image is to use a digital camera rather than a flat-bed scanner. Some projects select digital cameras over scanners for other reasons, such as the lower labour costs and reduced physical damage to books which stem from not having to place a book flat on a scanner bed for each page to be scanned.

When using a digital camera for capture, the lens is necessarily placed some distance away from the document in order to reduce spherical distortion, whereas a scan-head would normally pass extremely close to the document. This presents the problem of lighting the page evenly from a distance. Using a scanner, both the sensor and illumination source pass extremely close to the document allowing a flat document to be illuminated evenly throughout.

When the document must be illuminated from a distance, however, this may cause some areas of the document to be further from the illumination source, resulting in the document being unevenly lit. In extreme cases, this may cause difficulties during layout analysis and other stages of the image analysis process.

### 2.3.3 Historical documents

The problems for layout analysis discussed so far in this chapter have focussed on problems which are applicable to all documents. In some more specific areas, such as historical documents, there may be other document features which complicate the Layout Analysis process.

Historical documents, which may range in age from several decades to several centuries, vary significantly more in quality than most modern documents. The quality of historical documents depends on a large range of features — the type of material on which they were printed, the age of the material, the inks used during printing and the conditions in which they were stored.

Documents which were printed on good quality materials and which have been stored in temperature and humidity-controlled conditions may often be of a quality to rival modern documents. Where documents have not been adequately stored, the paper upon which they were printed may have coloured with age, or the inks used for printing may have faded, making the text stand out less from the background.

Documents held in archives or libraries for significant periods of time are often accessed quite frequently. If a document is handled regularly over several centuries, it will receive a significant amount of wear and tear. This may cause artefacts to be present in a document image which may not appear in more modern documents. For example, page corners may be folded, pages may be creased or torn, etc.

Given these problems, Layout Analysis and Document Recognition for historical documents is significantly more difficult and is a less mature area at present.

## 2.4 Layout analysis to date

Over the past few decades, a significant number of layout analysis methods have been proposed in the literature. This section provides a description of the different types of layout analysis methods with a description of the more notable methods in each category.

### 2.4.1 Top-down approaches

Some of the earliest approaches to Layout Analysis were based on simple, top-down approaches. Methods of this type rely upon simple, global document features to detect the divisions in the document. The use of such global document features means that methods using this approach are capable of segmenting only relatively simple documents, such as technical journals.

## X-Y Cut

In 1993, Nagy, Seth & Viswanathan presented one of the earliest approaches to Document Layout Analysis[16] which used the horizontal & vertical projection profiles to decompose the document image into individual regions.

The method operates on a bi-level image of the document page. It begins by calculating one of the projection profiles. For example, assume that the horizontal-projection profile is calculated initially. An example document image together with a horizontal-projection profile is shown in Figure 2.6. This takes the form of a histogram with one entry for each row in the image, with the value of each entry being the number of black pixels present in the corresponding row of the image. This histogram will contain peaks corresponding to areas of text and troughs corresponding to horizontal gaps on the page, e.g. the gap between the page title and the body text.

Once the projection profile has been calculated, the histogram is thresholded, i.e. each value of the histogram is set to either 1 or 0 depending on whether the value is above or below the threshold used (the calculation of this threshold is not explained). Thus, the thresholded projection profile should contain a 1 for rows with some content and a 0 for rows which are mostly blank. Figure 2.6 contains an example thresholding of the horizontal projection profile.

This thresholded projection profile is then analysed to find the contiguous segments — rows of 1s or 0s. The length of each of these segments is recorded then used to allocate them to pre-defined categories which do not have any meaning at this stage.

The segmentation method requires a user-specified grammar for the particular style of page to be segmented. This grammar takes the form of labelled sequences of the categories introduced during the previous stage. So, for example, a technical article's front page might be described in the vertical direction as a long string of 0s representing the gap above the title followed by one or more long strings of 1s separated by smaller strings of 0s, representing lines in the title and the gaps between. Below that, there may be a medium-sized string of 0s representing the gap between title and body text, followed by alternating short strings of 1s and shorter strings of 0s, representing lines of body text and the gaps between them.

Once each of these strings has been allocated to a category using the horizontal-projection profile, the horizontal strips of the page related to those strings may be analysed in the other direction, using the vertical-projection profile of just that strip. By performing this process recursively, the page is segmented into its constituent regions.

This method was developed during the early 1990s and was specifically designed to operate on technical journals. Documents of that type typically have an extremely simple layout,



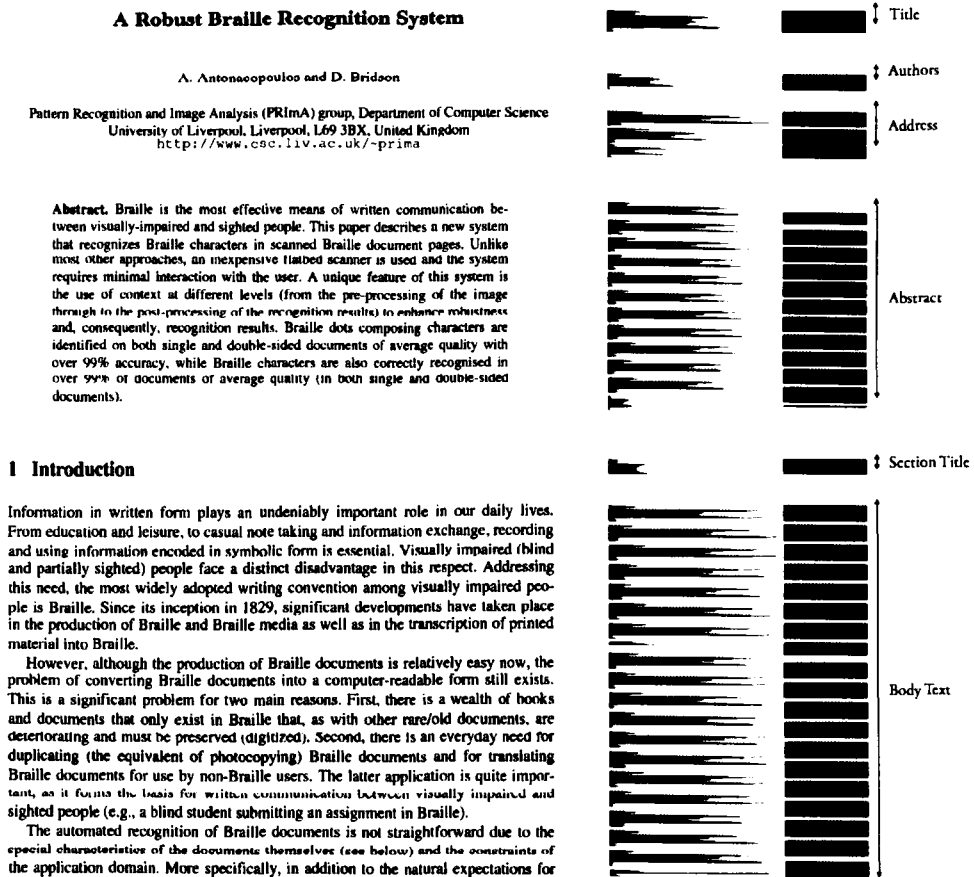


Figure 2.6: From left to right, an example document image of the title page of a journal article, the horizontal-projection profile calculated by counting the black pixels on each row and an example thresholding of the projection profile with the structure labelled.

which is often referred to as “Manhattan” given the similarity to the grid-like layout of the streets of Manhattan. Given this, the use of horizontal- and vertical-projection profiles is acceptable for this type of document. However, this means that many documents with complex layouts, i.e. those which cannot be segmented solely by horizontal and vertical cuts, may not be segmented with this method.

Similarly, the method was designed to deal with highly-regular layouts. Technical journals typically have quite strict requirements for the layout of articles. They are usually required to contain a given set of features, with each feature in a certain place on the page and of a certain size. The use of grammars is tailored for this type of layout. However, this means that documents which deviate from the expected layout, or documents which do not have a predictable layout, may not be segmented by a method of this type.

One of the disadvantages of the projection-profile-based approach is in the sensitivity to noise. Noise in the document may cause noise to be present in the projection profiles. This may mean that the detected layout may deviate from the ideal one. The researchers recommend the application of a noise-reduction technique to documents before analysis with this method. However, this also causes the removal of smaller features from the page, such as full stops and semi-colons.

The process may not be applied to images containing skew since the method uses horizontal- and vertical-projection profiles to detect the gaps between regions. Where the page is skewed, this may cause gaps between regions to be less well-defined in the projection profiles (or eliminate them entirely) and so cause problems with the segmentation. Similarly, such approaches may not be applied to documents containing regions of multiple different orientations.

### **Image transforms & texture segmentation**

Another early approach to top-down page segmentation was presented by Jain & Bhattacharjee in 1993[10]. Their approach uses whole image transforms in order to detect the texture of the different parts of the page then, using the detected textures, classify portions of the image into different region types. The page is then split into regions based by grouping connected areas of the same region type into individual regions.

The approach is based on the fact that different region types have different textures. For example, if one analyses a text region from top to bottom, one will encounter groups of black pixels corresponding to the text lines, separated by slightly smaller groups of white pixels corresponding to the gaps between text lines. If one analyses a text region from left to right, one will encounter smaller series of black pixels corresponding to individual characters, separated by very small gaps corresponding to the gaps between characters. So, one can determine the

type of a region by detecting the frequency & direction of textures in the image.

With text regions, we will encounter a rapid succession of changes between black and white in specific directions. With background regions, i.e. portions of the image with no content, one will encounter a homogeneous white texture which will be identical in all directions. Similarly, with image regions, one would expect to find less homogeneous textures but with frequencies far less uniform than is the case with text regions.

Given these observed properties of document images, this method applies a series of Gabor filters to the original image to create eight textured images. The filters are applied in four different directions in the image,  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  and  $135^\circ$  and, for each of those angles, at two different frequencies, the second twice as large as the first. Observing the same pixel in each of the eight filtered images allows the frequency & direction of the texture at that point in the image to be observed.

Pixels in the image are then clustered into three groups based on the values in each of the eight filtered images, corresponding to text regions, uniform regions (background and images with homogeneous textures) and boundaries of uniform regions. The pixels which were classified as text are then isolated and grouped into regions by performing connected component analysis.

One of the advantages of this method is that, since it uses filters in a variety of directions, the method will operate as designed even in the presence of severe levels of skew.

Although the method typically extracts actual text regions quite well, it is prone to detecting erroneous text regions from non-text portions of the image where the texture of that region, or part of region, is similar to that of a text region. Although the method performs quite well when it is only desired to detect between text and non-text/background regions, if one desires to segment a larger number of region types, the quality of the results will inevitably deteriorate significantly.

This method, and others of the same type, have also drawn criticism that they are relatively time consuming since they require the application of eight full-image transforms for each document to be processed, requiring not only a large amount of processing time but also a large amount of memory to store eight additional transformed images.

### 2.4.2 Bottom-up approaches

A major criticism of the top-down approaches mentioned in the previous sections is that, due to the use of simple global features, they are unable to identify complex-shaped regions. In order to increase the range of documents which may be segmented and the accuracy with which the segmentation can be performed, a new class of methods was introduced which,

rather than using global features, operate by combining small local features from the document such as connected components and combine these iteratively to find the regions of the document. The smaller, local scale allows such methods to gain a more detailed understanding of the document's structure.

### Document Spectrum, or Docstrum

In 1993, Dr. O’Gorman[18] published the *Document Spectrum* approach to page segmentation which uses connected components to segment the page in a bottom-up manner. In a bi-level image, a connected component is a contiguous area of the page which is black. So, for example, a single letter “P” would be a single connected component whereas a semi-colon would be comprised of two connected components, one for each dot.

Before processing of the docstrum begins by finding the connected components of the page, a pre-processing step of noise reduction is performed using a filter designed to remove noise while leaving text intact. Once this is done, a connected component analysis is performed on the whole page. Recognising that pages may contain text of differing sizes which may need to be treated separately, a histogram of component sizes is calculated in which peaks are detected. These are then used to separate the connected components into groups which are processed separately in the following stages.

Operating on a single group of connected components, the k-nearest neighbouring connected components of each connected component are detected, where the distance between components is the Euclidean distance between their central points. A value of 5 is suggested for k when performing layout analysis. The angle and distance between each connected component and its nearest neighbours is recorded. A scatter plot is made of the nearest neighbour angle against distance. This plot will display clusters which reveal the angles of text lines, the average inter-character spacing, the angle between adjacent text lines and the average inter-line spacing.

From these data, histograms of the nearest neighbour angle and nearest neighbour distance are produced. The peak of the nearest neighbour angle histogram is used to obtain a preliminary estimate of the skew angle of the page. Once the orientation of the page is known, separate histograms are produced by separating the nearest-neighbour relationships into within-line and between-line ones. These may be used to estimate the average inter-character gap and inter-line gap, respectively.

The nearest neighbour relationships which are close to the inter-line angle are then used to detect the text lines. A linear regression is performed on the centre of the characters in each text line to gain a more accurate estimate of the skew angle of the page.

The text lines are then grouped into blocks using features such as whether or not they are parallel, the perpendicular distance between them and the degree of overlap.

One particular advantage of the method is that it was designed particularly to be completely skew-independent. In other words, a document scanned at any angle should be segmented correctly. Given that skew angles are found independently for each text line, this also allows images containing portions with multiple different skew angles to be segmented correctly.

One problem with this method is that, during the  $k$ -nearest neighbour stage, the value of  $k$  to be used depends greatly on the contents of the page. Using a higher value of  $k$  will increase the required processing time while maximising the chance that between-line relationships are found correctly. However, this comes at the risk of accuracy. Using a smaller value of  $k$  will increase the accuracy but decrease the chance that inter-line angles are found.

Another problem which is not addressed by the author is the separation of non-textual elements. It is recommended that the docstrum method be applied separately to different groups of connected components based on size, it does not address whether or not this allows graphics of a similar connected component size to be merged with text regions.

### 2.4.3 Hybrid approaches

Although the bottom-up approaches described above allow a more detailed description of the document's structure to be obtained, there can be some problems making decisions solely on a local scale. Whether or not two components belong in the same region is not just a function of the components themselves but of the spacing between them. In order to gain a more detailed understanding of the document's structure, methods have been proposed which use both global features — such as the spacing between regions — and local features — such as the individual connected components which make up regions — to segment the page with a greater level of accuracy.

#### White Tiles analysis

In 1994, Dr. Antonacopoulos et al.[1] presented a new document layout analysis method which views the document regions as being separated by background space so builds a description of the background space of the document as a way of identifying the regions of the document.

The method divides background space into two categories. There is white space which separates regions and white space which is part of regions, for example, that which separates

two vertically-adjacent text lines. The former is considered as interesting while the latter is ignored.

The method differentiates between the two by size, reasoning that vertical gaps between adjacent regions ought to be larger than vertical gaps between adjacent text lines. So, the method estimates the text-line gap by analysing the distance between peaks in the horizontal-projection profile.

This distance is then used to perform a basic image transform termed *smearing*. This involves reading the document image from top to bottom and setting any continuous runs of white which are lower than the detected threshold to black. This has the effect of merging vertically-adjacent text lines while preserving larger gaps such as might separate two vertically-adjacent regions.

Once the gaps between text-lines have been eliminated, the method begins to build a description of the background of the page. It does this by reading the page from left to right and top to bottom to find connected *runs* of white pixels. This begins with the first row on the image. When the second row is analysed, the runs are compared with the runs from the rows above. If a given run is very similar to one in the previous line, i.e. they are at a similar position and size, then the two are merged to form a white tile. If a run overlaps a run from the previous row but is of a dissimilar width, then a new white tile is created to hold it. This process continues for the whole page until the bottom is reached. Once the process is completed, a description of the background has been built up by fitting white tiles into the blank space of the document.

Once the description of the background has been built, the borders of regions are surrounded by white tiles. The outline of the region can be reconstructed by creating a graph of these white tiles. There is a node for each white tile and an edge between the nodes of vertically-adjacent white tiles. The method uses a novel tracing method to identify minimum cycles in the graph which correspond to the outlines of regions. Once a cycle has been detected in the graph, a geometric description of the region outline can be recovered from it.

Although the method relies upon vertical smearing operations & detecting horizontal white runs and white tiles, the method incorporates flexibility which allows non-rectangular regions to be identified. The outlines of complex regions are often decomposed into a large number of extremely small, perhaps 1-pixel-high white tiles. This allows the outline of non-rectangular regions to be detected accurately.

This feature which allows the accurate segmentation of non-rectangular regions also enables the method to segment moderately skewed images with good accuracy. In skewed images, the size of the inter-line gap needed for the smearing stage often increases. For documents with a larger degree of skew, the larger smearing value required to join adjacent text-lines may also cause neighbouring regions to be merged. So, the method works best with low-to-moderate amounts of skew, with performance degrading at larger skew angles.

The usage of global smearing and minimum white tile width values causes difficulty for document pages which contain regions of text of significantly different sizes or of significantly different line spacing. Take, for example, a document page with a title in a large font and a larger amount of body text in a normal sized font. When detecting the inter-line gap, the value will usually be optimal for the body text. However, this may cause adjacent lines of text in the header not to be merged and so they may be detected by the method as being of different regions.

### Segmentation using the Area Voronoi diagram

In 1998, Dr. Kise et al.[12] presented a new approach to layout analysis which attempted to improve upon the docstrum approach described in the previous section. One of the problems with the docstrum approach is that it requires the value of  $k$  to be specified for the  $k$ -nearest neighbour processing, but the value of  $k$  required depends on the layout of the page. This newer approach eliminates the need for this to be specified.

The Voronoi Diagram, in mathematics, is a diagram which, given a set of points in a 2-dimensional space, divides the space into a set of regions, one for each point, where the region corresponding to each point represents the portions of the space which are closer to that point than any other. Each of these regions has a boundary which represents the points which are equidistant to two points.

The page segmentation method uses a modified version of this diagram called the *Area Voronoi Diagram* which is based on connected components rather than points. Initially, the connected components of the page are calculated in a similar way to the docstrum method discussed earlier. Once the connected components are found, a series of points on the outlines of connected components are selected. These are used to build a conventional Voronoi Diagram.

Once the conventional voronoi diagram has been constructed, it is converted into an area voronoi diagram by deleting from it all edges which separate areas belonging to the same connected component. The resulting diagram splits the page into regions around each connected component. In this graph, the region around each connected component will border

a number of other regions. For example, a region corresponding to a letter in a word will automatically border the regions of the adjacent letters in the word, as well as some letters in the words above and below. Thus, an analogue of the  $k$ -nearest neighbours from the docstrum approach has been found, without the need to specify  $k$ .

The edges in the Voronoi Diagram all represent the boundaries between regions. Some edges, such as those between adjacent characters in the same word, are not interesting from a page segmentation perspective. Other edges, such as those separating adjacent columns of text, or those separating a title from the body text, mark the boundaries between regions. Thus, the task of segmenting the document image becomes the task of deleting uninteresting edges from the area voronoi diagram. If this can be accomplished successfully, the remaining edges will be the boundaries of regions.

The method uses two measures when deciding whether the edge between two connected components should be deleted. The first is the area ratio of the two connected components. This allows edges between components of widely varying sizes, such as the edge between an image and a character of body text, to be retained, while edges between more similarly-sized components may be deleted.

The second measure is the minimum distance between the two connected components. This is used to differentiate between much closer connected components which should be part of the same region, such as characters in the same word, and much more distant ones, such as two characters in adjacent text columns, which should be separated. This uses a threshold calculated from a histogram of connected component distances, similar to that used for the docstrum method.

Once the selected edges have been deleted, the remaining edges relate only to the boundaries of regions.

This method is a novel approach to the page segmentation problem which allows good results to be obtained from documents regardless of the skew angle. The segmentations obtained at extremely different skew angles are often virtually identical due to the operation of the method. The method performs extremely well on images containing areas with different skew angles.

The authors tested the new method on the University of Washington dataset. They report that the method achieves excellent quality segmentations of body text regions but they report much lower results on other text regions (headings, etc.) and non-text regions. The lower results for these types of regions may be due to the use of global thresholds for splitting connected components based on distance, which would apply less well to regions of atypical size.



## 2.5 Maturity of the area

This area has been the focus of significant research for a long period of time. The recent approaches described above take into account many of the features found in modern printed documents.

There have been a series of independent assessments of layout analysis methods published in the literature[8][6][4][5] which have measured the maturity of modern methods on a range of complex modern documents. These found that modern methods perform quite well on relatively simple document pages but still have significant problems when dealing with non-textual regions and more complex features.

Given this, there is still a significant amount of ongoing research in the area which aims to increase the quality of layout analysis systems and the range of documents to which they may be applied.

## 2.6 Summary

This chapter has given a description of document layout analysis and its place in the document image analysis process. Some of the problems faced in layout analysis were discussed then a description of the different types of layout analysis methods was given along with detailed descriptions of some of the notable methods of each type. Following this, a brief description of the maturity of the area at this stage was given. The following chapter discusses performance evaluation.



# Chapter 3

## Performance Evaluation

### 3.1 Overview

The previous chapter contained an overview of the area of Document Layout Analysis and a brief description of the types of Layout Analysis method in the literature. This chapter discusses Performance Evaluation, its potential uses, Performance Evaluation methods in Document Image Analysis in general and more specifically previous approaches to performance evaluation related to Document Layout Analysis.

### 3.2 Performance Evaluation

The area of Document Image Analysis has been the focus of a large amount of research over the past several decades and a number of different areas are currently considered to be relatively mature, e.g. OCR of modern printed documents, layout analysis of clean, simple documents.

Since Image Analysis is an area of ongoing *scientific* research, when new methods are presented for publication, they invariably must be accompanied by some evaluation to give an idea of the particular advantage of the given method, be it in terms of an increase in recognition accuracy or an increased range of document types which may be recognised.

When research in the area is published, it is usually accompanied by such an evaluation. However, a recurring theme among such evaluations is that they are often performed on custom-produced, application-specific datasets and using custom performance evaluation methodologies. These two facts mean that it is extremely difficult to compare the results of any two methods in a given area against each other.

The use of common representative datasets for evaluating different Image Analysis methods is extremely important. If two Image Analysis systems are evaluated using a common

testing method but using different datasets then the results will not be comparable since the relative *difficulty* of the two datasets is unknown. The topic of common datasets is discussed further in the next chapter.

Similarly, if two Image Analysis systems are compared using a common dataset but separate evaluation methods, then the results again will not be comparable. So, it is important in any area to have a common testing methodology to ensure that the results presented for different methods may be compared.

Performance Evaluation may be required for a number of different purposes:

### **Benchmarking**

The simplest application for Performance Evaluation is in providing benchmarks for given methods. This entails providing, for a given method, either a statistic or group of statistics which quantitatively summarise the performance of the given method.

### **Selecting methods for specific uses**

When selecting a Image Analysis methods for a given purpose, the best method to be used depends upon the particular use to which it will be put. For one application or specific type of document, a given method may produce better results, while for a different application, it may perform worse than other methods. When selecting a method for a given use, it is necessary to produce a higher level of detail which details the specific problems with a given method and the areas for which it is most suited.

### **Targeting development effort**

Finally, and perhaps most importantly, is the task for developers of Document Image Analysis methods, of improving their methods. For such a task, a wide range of detailed information is desirable. It is necessary to know which are the most significant categories of errors produced by a method at the current stage of development, the single features which cause the most errors in the method, the documents with which the method has the most difficulty, etc. If a developer has such information, then it is possible to target the development of the method where it is most needed and, ultimately, improve the performance of the method.

## **3.3 Performance Evaluation in Document Layout Analysis**

This section discusses the desirable characteristics of Performance Evaluation methods for Layout Analysis systems then describes the prior approaches in the area.

### 3.3.1 Desirable characteristics

There are several characteristics which are important when comparing performance evaluation methods:

#### Accuracy

When evaluating the performance of an Layout Analysis system, it is important that the evaluation be performed with a high degree of accuracy. The accuracy of a system is reflected in the particular methodology used and the specific data types used. Where evaluation is not performed accurately, the results may be misleading, potentially causing problems to be found where none exist and none to be found where problems do exist.

The accuracy depends to a significant extent on the data structures used. A large number of documents contain complex-shaped regions so the region representation chosen should reflect this. However, some performance evaluation methods still rely on bounding box representations for region outlines.

Similarly, the accuracy of an evaluation depends on the accuracy of the ground-truth used. The ground-truth should be the perfect digital representation of the detected layout. So, it is necessary to ensure that the schema used for the ground-truth is capable of describing accurately the regions of the page. Some datasets still rely on bounding-box representations which will negatively affect the accuracy of evaluations based on them for documents containing non-rectangular regions.

#### Applicability

It is similarly important for performance evaluation methods to be applicable to as broad a range of documents as is feasible. As described in the section on accuracy above, it is important for the methodology and data structures chosen to be capable of reflecting all the complex features which may be encountered in a modern document page. Where a method cannot accurately describe the full contents of a given document, then it cannot provide an accurate evaluation of that document.

This is important in evaluating Layout Analysis systems to the fullest extent. If a performance evaluation method cannot operate on complex documents, then it will be possible only to evaluate Layout Analysis systems on simpler documents. This will result in the system to be evaluated receiving less scrutiny than is desirable, potentially giving the system a higher score when a lower one might be more appropriate and missing problems in the system due to the lack of more complex features in the dataset used for evaluation.

### **Descriptiveness**

Earlier in this chapter, the potential uses of performance evaluation methods were discussed. Rather than simply providing a single statistic for an evaluation, as might be adequate in a benchmarking scenario, it is often desirable to provide a more detailed assessment of a system, in terms of the different types of errors made, the most severe errors, the documents which pose the greatest difficulty, etc. This is particularly the case when the system is being used by developers who require information to further the development of their algorithms. So, it is a desirable feature of any performance evaluation system for the output to be as descriptive as possible.

### **Flexibility**

Document Layout Analysis methods, and Document Image Analysis methods in general, are often used for a wide variety of applications in a wide variety of organisations. Each of these different applications may place different requirements upon the methods involved. For instance, in an indexing application, it may be important to be able to accurately detect the titles and bylines of articles, while other parts may be less important. In an image indexing application, it may be important to segment images, their captions and copyright details while ignoring other features. In an archival situation, it may be important to capture all aspects of a document correctly.

Given the wide variety of applications for Layout Analysis methods and document types upon which they operate, it is likely that no specific Layout Analysis method will be optimal for all of those applications and document types. Instead, some methods may display strengths in particular areas and thus be more adapted for a specific application area. So, it is desirable for a performance evaluation method to take this into account and be able to evaluate documents according to how well they meet the requirements for specific application areas.

## **3.4 Review of Performance Evaluation methods**

### **3.4.1 Overview**

There have been a wide variety of Performance Evaluation methods proposed in the literature. The previous approaches may be divided into three categories: OCR output-based, image-based and region-based.

The earliest methods were intended to operate on the layout analysis modules of commercial OCR systems. They operated on the text of the complete output of the OCR process. Since they operated on text only, they were unable to provide a full evaluation of pages containing non-text regions.

Image-based performance evaluation methods were proposed in order to expand performance evaluation to documents containing any type of region. These methods describe regions in binary documents as the set of black pixels contained inside them. The main problem with such methods is that they operate accurately only for pages consisting of black content on white backgrounds. For documents containing full colour images or other colours of text and background, such methods may not perform optimally.

The final category of methods are called region-based. Such methods rely solely on geometric comparisons of regions from the ground-truth and segmentation, without referring to the image. This allows evaluation to be expanded to documents containing all region types and operate as intended regardless of the contents of the regions. However, region-based approaches are currently less well-developed and are largely based on bounding box representations, which preclude the use of existing methods for evaluating more complex documents.

This section gives a description of each of these types of method and gives an in-depth description of the specific methods in each category.

### 3.4.2 OCR output-based Performance Evaluation

The earliest methods for performance evaluation were designed for evaluating the Layout Analysis modules of commercial OCR systems. Since the systems were only available to researchers as a *black box*, the performance evaluation could only be performed on the OCR output of the systems. While these have the advantage that they may be used to evaluate the black box commercial systems for which they were designed, they do not provide a direct assessment of Layout Analysis, so the results may be affected by the subsequent OCR stage. Since they rely upon OCR systems, they are not capable of evaluating standalone layout analysis methods, as may be designed by researchers. The use of text matching precludes the correct evaluation of any page containing non-textual elements or containing a script for which no OCR method is available.

#### Kanai, Rice, Nartker & Nagy

One of the earliest attempts at performance evaluation specifically related to document layout analysis was the system developed by Kanai, Rice, Nartker and Nagy at the University of Nevada at Las Vegas[11].

The focus of the research was the evaluation of the layout analysis modules of commercial OCR systems. Given that they were dealing exclusively with commercial OCR systems, they did not have any access to the internal workings of the systems. Rather, the systems could be treated only as black boxes, meaning that all results had to be inferred from the textual output of the complete process of Layout Analysis followed by OCR.

The OCR systems under evaluation allowed the OCR process to be run either with an automated layout analysis step or allowed a human operator to supply a manually-entered layout for the page. The researchers took advantage of this fact to allow the layout analyser itself to be evaluated.

The input to the evaluation system is a pair of strings, representing the segmentation and the ground-truth. The segmentation string is the textual output of the OCR system when run using the *automatic* layout analyser, while the ground-truth string contains the textual output of the system when the layout is *manually* supplied by a human operator.

Since the focus of the research is mainly OCR systems, the evaluation is tailored towards this goal. The idea behind the system is that the goal of OCR is to extract from an image the correct text in the correct order. Using automated segmentation, where the segmentation does not perform perfectly, the text string output from the OCR system will differ in some way from the ground-truth string. Some parts may be omitted, some non-textual parts may have been inadvertently included in the OCR process and parts of the text may appear in an incorrect order.

Assuming that the goal of OCR is to recover the text from a document without error, then any imperfections during OCR process (and the layout analysis process in particular) will require some further corrections by a human operator, which means incurring some cost. This evaluation system measures the performance of a method in terms of the editing operations which must be made by a human operator in order to correct the segmentation text string. These corrections are broken down into several different operations:

- Moving portions of text to a different position in the document
- Deleting some text which does not belong in the document
- Inserting text into the document which has been mistakenly omitted

For each of the operations mentioned above, the system allows the user to supply an associated cost which means the system can be evaluated in terms of the real-world cost of correcting mistakes made during the layout analysis step.

In order to find the cost of correcting the segmentation text into the ground-truth, the system begins by finding the longest common substring of the two strings. This is marked as



the first match. It then continues to find the next longest common *unmatched* substrings of the two strings and continues until all the common unmatched substrings have been found. Then, the only unmatched substrings in the ground-truth correspond to substrings which need to be manually re-typed in the segmentation string. Likewise, the remaining unmatched substrings in the segmentation string (i.e. which have no equivalent in the ground-truth) correspond to strings which need to be deleted from the segmentation string. So, these are recorded as insertions and deletions, respectively.

One of the chief advantages of this type of system is that, since it operates on the textual output of an OCR system, it is capable of evaluating the performance of commercial OCR systems from which it is not normally possible to extract the results of page segmentation. In that respect, this may provide the most detailed evaluation possible of commercial OCR systems which do not provide details of the page segmentation process.

This method also has a number of flaws. Given that it operates solely on the textual output, it is not capable of evaluating segmentation of non-text regions such as images, graphics and separators. On pages containing such contents, the only evaluations possible with such a system will be incomplete ones.

Another problem arising from the evaluation based on text output means that the system can only be used on complete OCR systems. Given the aims of the research, i.e. gaining an understanding of the relative strengths of commercial OCR systems, this is understandable but it precludes the system from being used to evaluate methods produced by researchers which do not necessarily come with an attached OCR system. Given the reliance on the OCR stage, such methods are unable to provide evaluations for textual documents containing scripts which are not OCRable.

Another potential criticism is that the focus on manual editing operations is now slightly outdated. The original expectation was that document image analysis systems were to be used in conjunction with human proofreaders. Today, however, the focus of document image analysis has shifted towards large-scale digitisation projects such as those mentioned in chapter 1. When dealing with the digitisation of millions of books, human post-correction becomes economically unviable. The de-emphasis of manual post-correction means that evaluating layout analysis methods in terms of the cost of post-correction no longer provides a useful measurement of performance.

### 3.4.3 Pixel-based performance evaluation

In order to address the problems with the OCR output-based methods described above, a new class of performance evaluation methods was developed which focused on evaluating

the Layout Analysis stage directly rather than making inferences from OCR results. These methods take the view that, when a document is described as a bi-level image, the actual contents of the page are stored in the black pixels while the white pixels of the image represented the document background. When the primary focus of Document Image Analysis was simpler documents such as technical journals which consisted, predominately, of black text on a white page, this was understandable. However, for more complex documents containing text and backgrounds of differing colours and multi-colour images, such approaches may not be as accurate as might be desired.

### Yanikoglu & Vincent

In 1997, Yanikoglu and Vincent[29] presented the Pink Panther system for ground-truthing and performance evaluation of Document Layout Analysis. The performance evaluation aspect was based on an image-based approach.

In contrast to the previous approaches based on string matching of OCR results, this method was based on the contents of the regions. The system takes the view that the page's useful content is stored in its black pixels, while the white pixels comprise the page background. A region in the ground-truth or segmentation is described not by the geometrical outline of the region but by the set of black pixels included within.

The method begins by making a reduced-resolution *region map* which is a labelled image specifying to which regions in the ground-truth and segmentation a given pixel belongs.

Region correspondences are detected by scanning through the region map for each segmentation and ground-truth pair whose bounding boxes overlap and calculating the number of black pixels involved in the overlap. A *match score* is calculated for each pair which is calculated as the percentage of the ground-truth region's black pixels covered by the segmentation region minus the percentage of black pixels of the segmentation region which fall outside of the ground-truth region. This provides a measure of how well a ground-truth region is matched by a given segmentation region.

Once the region correspondences have been detected, they are allocated into categories: wrongly-detected, missed, horizontally split, vertically split, horizontally merged, vertically merged or mislabelled.

The user has the opportunity to specify weightings for each type of error and each type of region. Similarly, the costs of each error can be selected by the user as being weighted either just by a count of regions, the height of the involved regions and the on-pixel area of the regions involved.

This method is one of the most advanced performance evaluation methods presented to

date. It allows significant flexibility so the user can receive an evaluation which is tailored to a specific application area. Similarly, the method allows results to be weighted by the black pixel area of the regions involved, meaning that errors involving larger regions will be more highly weighted than those involving smaller regions.

The principal disadvantage to this method is that it relies upon black pixel matching. Black pixel matching makes the assumption that the page's contents are stored in the black pixels of the image. Take, for example, a full colour image of a sunset. Typically, there would be a red sky with a glowing sun in the centre and the landscape in green beneath it. When such an image is binarised, different parts of the image will become black or white. For instance, the sun would probably become white. However, the sun was a useful part of the image, potentially even the most important part. However, if that portion of the image becomes white in the binarisation, it will be ignored by methods using on-pixel matching due to the assumption that the useful content of the page is black. As Document Image Analysis moves more towards operating on colour images and more complex colour documents, such assumptions make less sense. For documents containing more than black text on white backgrounds, methods based on black-pixel matching may cause the results to be weighted too heavily towards a particular region type or fail to detect genuine errors.

### **Thulke, Märgner & Dengel**

In 1998, Thulke et al.[25] proposed an image-based approach to performance evaluation of segmentation results. Their proposed method was generalised in that it could be applied to the full range of segmentation tasks, including character segmentation, as well as page segmentation.

Similar to the previous approach, the method uses an image-based approach in order to perform a true evaluation of layout analysis, compared with earlier text-based approaches.

The matching begins by dividing the image into disjoint sets of black pixels, one set for each ground-truth region. The process is then repeated for the segmentation regions. Correspondences between regions may then be discovered by finding intersections between the ground-truth sets and the segmentation sets. Where the intersection between a ground-truth set and a segmentation set is empty, the regions have no overlap. Where the intersection between the two is not empty, then there is some overlap.

The overlaps are classified into a number of groups corresponding to one-to-one matches where exactly one ground-truth region overlaps a given segmentation region, many-to-one

matches where a ground-truth region is split into multiple segmentation regions, one-to-many matches where several ground-truth regions are merged by a single segmentation region and many-to-many matches where multiple ground-truth and segmentation regions are overlapping. The method also detects ground-truth segments with no overlapping segmentation segments (missed) and segmentation segments which overlap no ground-truth regions (wrongly detected).

These errors are then divided into 19 separate classes of various combinations of merges, splits, wrongly detected background, wrongly detected noise, etc. Statistics are then output based on the number of regions falling into each of these categories.

As one of the earliest image-based performance evaluation methods, this method provided a distinct improvement over the prior text-based evaluation methods. With such methods, it became possible to evaluate pages containing non-text regions such as separators, graphics and images. However, the system was intended to be a general approach and is designed accordingly. Therefore, the system only differentiates between background, noise and content regions. Regions are not divided into different categories such as text and images. Therefore, it is impossible for the method to judge the magnitude of errors from a page segmentation viewpoint. So, merges are all considered as merges regardless of whether they involve regions of different types or the same type or whether merges between text regions occur within columns or across columns.

The output statistics are based solely on a count of regions falling into each category. Unfortunately, the system does not classify errors according to the size or importance of the regions involved. A merge between two small regions would be weighted exactly the same as a merge between two large regions. There is also no provision for flexibility for the system by allowing application-specific weightings to be specified by the user.

The use of the black pixel region contents is adequate to deal with simpler documents which consist solely of black information on a white background. The usefulness of this is brought into question when modern colour documents are involved. For pages with non-textual regions, the method will give results which may not take into account the full content of the page. Images in particular are likely to contain a variety of light and dark parts which will variously be binarised into white and black. So, useful contents of the page may be ignored by this method if they binarise to white. Additionally, in many modern documents, it is not unusual to find pages which have differing background and text colours. Such methods may not be able to cope with this.

### Gatos & Antonacopoulos

A series of competitions in the area have been run by Drs. Gatos and Antonacopoulos[8][6][4][5] in the scope of the International Conferences on Document Analysis and Recognition which aim to determine the current state of the area.

The competitions use a pixel-based matching algorithm which begins by creating a *match score* table for each pair of ground-truth region and segmentation region in the documents to be compared. For each of those pairs, the relevant cell in the match score table is set to a value which is the number of black pixels in the intersection of the two regions, divided by the number of black pixels in the union of the two regions. Effectively, this match score will range from 0 when the regions have no intersection at all to 1 when the two regions overlap perfectly.

This match score table is then used to detect a number of matches between the ground-truth and segmentation regions. The number of one-to-one matches is detected, the number of many-to-one matches, one-to-many matches and many-to-many matches. These statistics are calculated from both the ground-truth perspective and the segmentation perspective and these are used to calculate a detection rate for the ground-truth and a recognition accuracy for the segmentation by multiplying the percentage of regions involved in the given type of error by a user-specified weight for that type of error. The two statistics are then combined to produce an overall error detection metric (EDM) for each region type and then an overall segmentation metric which is an average of the region-specific EDMs weighted by the number of regions of that type in the dataset. The latter two statistics are those which are used for judging the competition.

Since the competition is based on a pixel-matching approach, the same criticisms which apply to other pixel-matching approaches, also apply to it. One should note that there is a certain degree of inaccuracy in the competition results. Firstly, when the match score table is being used to detect matchings between regions, a threshold is used to detect overlaps which are considered to be less important. These are discarded for the final evaluation, meaning that regions could potentially be merged or split but ignored by the evaluation method.

When producing the final statistics, the errors are weighted only by the number of regions of that type in the dataset, rather than by the area of those regions. This can lead to results being inappropriately weighted towards regions which are less important.

This section has dealt exclusively with the evaluation method used for the page segmentation competitions. More details on the running of the competition, as well as results, are shown in section 3.5.

**Shafait, Keysers & Breuel**

In 2006, Shafait, Keysers and Breuel[23] presented a new approach to performance evaluation which is similar to the previous types of on-pixel matching techniques but simplifies the process by using standard image files to store ground-truth and segmentation data.

Rather than having a direct description of the region outlines, their region description takes the form of a colour image which shows, for each pixel, the region to which it belongs. Each region in the ground-truth is allocated a unique colour. Then, each pixel belonging to a given region is set to the given colour. Where a pixel is not part of any region, it is set to white.

Such an approach has several attractive features. The use of standard, previously-existing image formats allows existing software to support the format with little effort, making use of widely-available libraries. The format also allows complex region shapes to be described.

The ground-truth format does, however, have some drawbacks which reduce its usefulness for performance evaluation purposes. Since it uses a standard image format for storage, this means it cannot contain any document- or region-level meta-data. Even simple region types (e.g. text, graphic or separator) are not included which precludes using the format for evaluating region labelling, a crucial part of the document layout analysis process. The absence of region-level meta-data also precludes using the document format for a more fine-grained analysis.

Since both the ground-truth and segmentation are described as images, the detection of overlaps between ground-truth and segmentation regions is as simple as checking each pixel in the ground-truth image and comparing it to the same pixel in the segmentation image. During this process, a weighted bipartite graph is constructed. This has a node on the left side for each region in the ground-truth and a node on the right side for each region in the segmentation. Then, an edge is drawn between the ground-truth node on the left and the segmentation node on the right for each pair of ground-truth region and segmentation region which overlap. The edge is assigned a weight which represents the area of the overlap as measured by the number of black pixels. An example of the constructed graph is shown in Figure 3.1 .

The construction of this graph may be performed in a single pass over the two images so the time complexity of the graph construction is approximately linear. However, an average image is likely to have around 6 million pixels, meaning that the graph construction will likely take significantly longer than performance evaluation methods operating on more compressed document representations representing only region outlines.

Once the graph has been constructed, the method calculates several metrics from the graph which are used to describe the quality of the segmentation:

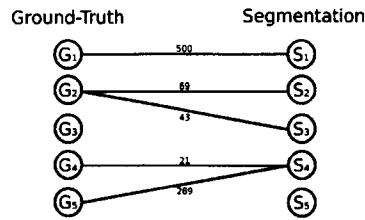


Figure 3.1: The region correspondence graph which shows the ground-truth regions and their overlapping segmentations, and vice versa. The edge labels show the number of pixels involved in an overlap.

- Total *Over-segmentations* — the number of significant edges connected to ground-truth nodes, minus the number of ground-truth nodes
- Total *Under-segmentations* — the number of significant edges connected to segmentation nodes, minus the number of segmentation nodes
- Over-segmented components — the number of ground-truth components with more than one significant edge
- Under-segmented components — the number of segmentation components with more than one significant edge
- Missed components — ground-truth nodes matched only against the background
- False alarms — segmentation nodes matched only against the background

This method provides a novel approach to region representation which allows complex region shapes to be described. The choice of images for the region representation provides both advantages and disadvantages. The images used are parsable by widely-available libraries so it would be trivial for existing applications to add support for this format.

The use of images also has a major disadvantage in that they do not contain any further metadata which is important for making a flexible and accurate evaluation. There is no region-level metadata available, nor is any information on the correct region orders available. So, for instance, the method cannot distinguish merges between different region types from merges of identical region types.

The use of black-pixel matching, albeit in a somewhat novel form, means that this method suffers from the same disadvantages of other pixel-based evaluation systems.

### 3.4.4 Geometric Performance Evaluation methods

The image-based performance evaluation methods described previously are not ideal for evaluating documents which do not consist solely of black text on a white background. In order to improve accuracy and increase the domain of documents to which they may be applied, a third category of performance evaluation method was introduced which evaluate performance based on the actual regions involved rather than their contents. These methods may be applied equally well to colour documents and bi-level ones but, so far, little research has been performed in this area. Current methods are incapable of dealing with the complex region shapes present in modern documents.

#### Liang, Phillips & Haralick

In 1997, Liang, Phillips & Haralick[13] presented a new performance evaluation metric designed to evaluate layout analysis methods with the University of Washington dataset which is discussed further in the following chapter.

This method accepts a ground-truth as a series of bounding boxes and a segmentation of the page also given as a series of bounding boxes. The method begins by comparing each pair of ground-truth and segmentation regions. Two metrics are computed which represent the size of the match, if any, between them. These are calculated as the area of the overlap between the ground-truth region and the segmentation region divided by the area of the ground-truth and segmentation, respectively. These are used to build two tables which specify the correspondences between all the regions in the ground-truth and the segmentation.

These tables are then used to check, using imperfect matching, whether regions from the ground-truth are well-matched in the segmentation (correctly detected), missed, split or merged with another ground-truth region or whether the segmentation contains any regions which do not correspond to anything in the ground-truth.

A series of weights are then used to weight the numbers of each type of error to form a weighted average which is used as a cost function.

This marks one of the earliest pure region-based evaluation methods to be presented. The method is intentionally kept relatively generic so that it may be applied to a variety of tasks, including word segmentation, line segmentation and text block segmentation.

However, the generic nature means that it is missing some key features which are required for performing a detailed evaluation of layout analysis. The method does not differentiate between regions of different types. So, for instance, if a text region were to be misdetected as an image region, the method would record this only as a correct identification.

When calculating the error metric, the method only takes into account the raw number



of errors. The method does allow different weights to be specified based on the direction of merges and splits but does not make any further attempt to judge the significance of each error.

### **Antonacopoulos & Brough**

In 1999, Antonacopoulos and Brough[3] published a method which introduced a region description to performance evaluation of layout analysis. Previous region-based evaluation methods had been based solely on bounding box representations of regions. Comparisons between such regions are extremely efficient but they are lacking in the ability to represent complex-shaped regions.

The region representation proposed by the authors is termed region intervals. This involves dividing each non-rectangular region into a series of rectangles. The method uses a *global* interval structure which splits the rectangles from each region so that the top and bottom edges align with similar edges belonging to other regions. This allows relatively complex documents to be described using a simple representation.

The paper proposes two approaches to comparison. The first is called the maximal polygons approach. For each region, a maximal polygon is constructed around it such that it fills the surrounding background space. In order to check whether or not a segmentation region matches the given ground-truth region, it is necessary to check that the segmentation region falls within the maximal bounding polygon. If it does, then the segmentation region has not merged the ground-truth region with any others. It also presents a reverse approach which is based on the same principle but attempts to match ground-truth regions to segmentation regions.

This presents a proposal for a method but the paper describes the novel region representation and the method of region comparison. Since the region representation and comparison technique were proposed, no further work was done on the project. Although development on this particular method did not continue, the concept of the global region interval is at the foundation of the method presented in this thesis.

### **Peng, Chen, Liu, Ding & Zheng**

In 2001, Peng, et al.[19] at Tsinghua University presented one of the most recent pure region-based evaluation methods. The method is region-based because, unlike the methods described earlier which use the on-pixel contents of the regions to perform their evaluation, this method performs its evaluation geometrically, referring solely to the ground-truth and segmentation regions.

The region description used for the evaluation is the bounding rectangle. This has several advantages over more complex region descriptions. The representation is extremely space-efficient as, for each region described, it requires just the co-ordinates of the top-left and bottom-right corners to be stored.

Similarly, the simple representation makes region comparisons very efficient. Given the bounding boxes of two regions, it is almost trivial to check whether one is contained within the other, or whether part of a region is not included in the other, and the extent of any mistakes. The simple representation also reduces the cost of ground-truthing, either reducing the amount of time needed to ground-truth a given number of documents, or increasing the number of ground-truths which can be produced in a given time.

The chief disadvantage of the bounding box representation is in terms of its flexibility. Modern documents contain a large number of regions which cannot be accurately represented by bounding boxes alone. The authors specifically select the bounding box representation because it aims to evaluate popular Chinese OCR products. When it comes to evaluating the more advanced segmentation methods which are not state of the art in research labs, bounding boxes are somewhat inadequate.

The method begins by finding the 1:1 matchings by exhaustively searching each ground-truth-segmentation region pair for a match. The algorithm incorporates some amount of flexibility. For each ground-truth region, a maximal surrounding region is generated which is the largest bounding box which will fit in the area surrounding the ground-truth region without encroaching on any other regions. This is illustrated in Figure 3.2.

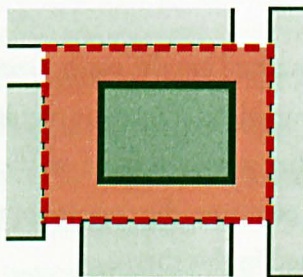


Figure 3.2: The maximal surrounding rectangle.

Once the maximal surrounding rectangle has been calculated, then the method checks each segmentation region to find if it falls within the maximal surrounding area and then to see if the segmentation boundary falls inside the ground-truth region proper. This is performed quite easily. Take, for example, the leftmost boundaries of the regions depicted in

Figure 3.3.

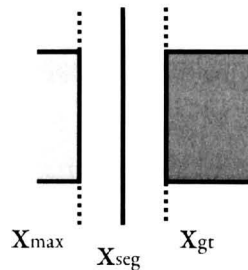


Figure 3.3: Example maximal, segmentation and ground-truth left boundaries.

The check may be performed simply by comparing the x-co-ordinates of the three left boundaries. If the left segmentation boundary falls between the left maximal surrounding ground-truth boundary and the left ground-truth boundary itself, then it has completely encompassed the region from this side. If the other three sides appear similar, this is termed an *equal match* — the segmentation region encompasses the whole of the ground-truth region and no parts of any other. Once all of the equal matches are detected, a count is taken of the number of *composite-match* regions, ie. segmentation regions which overlap more than one ground-truth region and ground-truth regions which are split into multiple segmentation regions.

Based on these results, several statistics are calculated which describe the performance of the segmentation method, the equal and composite matches as a percentage of the number of ground-truth or segmentation regions.

The pure region-based approach of this method is attractive in that it may be performed efficiently. However, the choice of a bounding-box representation as the foundation for the method limits its application to a small subset of documents.

The output from the method consists only of four statistics which may prove useful in benchmarking but with provide little help for those seeking to find the strength and weaknesses of particular methods.

While the evaluation detects correctly recognised and composite matched regions, it does not make any analysis of the types of region involved. For example, a recognition of a text region as a text region is treated the same as a text region erroneously recognised as a graphic.

No attempt is made to quantify the differences involved. For example, two split text regions will always be weighted identically even if one is a drop-capital and the other is a full column of text. There is also no possibility for users to customise the evaluation based on their specific requirements.

Overall, the method is a useful early attempt at pure region-based performance evaluation. However, the choice of region representation prevents it from being used to evaluate a large portion of existing documents. The lack of useful developer-oriented information prevents it from being used outside of benchmarking applications.

### Lucas, Panaretos & Sosa

In the scope of the International Conferences on Document Analysis and Recognition in 2003 and 2005, Lucas et al.[15][14] ran two competitions in text locating. Text locating is similar in some ways to the task of layout analysis although the methods by which the two are accomplished are quite different. While layout analysis involves finding the outline and location of various types of region on a relatively plain document page, text locating involves finding text in natural scenes, which typically have more complex surroundings than the document page.

The competition operated on the word level. Given a natural scene containing some text, a ground-truth was created containing the bounding boxes of the words in the scene. Participants were offered a training dataset which gave them an opportunity to train their methods on images similar to those used for testing.

When the competition was opened, entrants were invited to submit executables which would take as input an image and output the detected locations of words in the scene.

Since locating text in a natural scene is somewhat more difficult and necessarily less precise, it is necessary to incorporate a degree of flexibility into the evaluation system. So, the match between a segmentation region and a ground-truth region is calculated as the area of the intersection of the two divided by the area of the minimal bounding box which fully encloses both.

Measures of precision and recall are then calculated from these and an average of the two is calculated to find a overall metric to represent the quality of the text locating ability of the method.

The system relies upon a bounding box representation since this is the most natural fit for single words in a natural scene. Unfortunately, this, as well as the different nature of the problem, makes such a method unsuitable for application to Layout Analysis.

## 3.5 ICDAR Page Segmentation Competitions

In the context of the International Conferences on Document Analysis and Recognition from 2001–2007[8][6][4][5], Drs. Antonacopoulos and Gatos ran a series of page segmentation

competitions which aimed to find the maturity of the area.

The first competition, held in 2001, was dedicated to newspaper page segmentation. However, the following three competitions, held in 2003, 2005 and 2007, evaluated performance on a more general set of documents, containing magazine pages, technical articles and advertisements.

For these competitions, researchers developing layout analysis methods were invited to participate. The competition ran in an off-line mode. Participants were initially given access to a training dataset to allow problems with methods to be detected. Then, one week before the final submission date, participants were given access to the test dataset of around 32 images. The authors then ran their layout analysis methods on the given data then submitted the results.

The results were then evaluated, using a performance evaluation method discussed in section 3.4.3, against a manually-prepared ground-truth segmentation for each document. Regions which were merged, split, correctly detected or missed were measured and a segmentation metric was calculated to measure the performance of each layout analysis method on the whole dataset, with the values expressed as a percentage. The results of the three segmentation competitions for general documents are displayed in Figure 3.4.

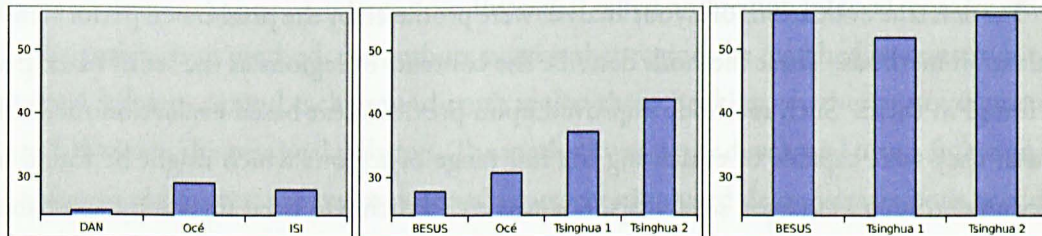


Figure 3.4: The overall results from the ICDAR 2003, 2005 and 2007 page segmentation competitions.

The datasets used for the competitions were different each year to prevent previous entrants from having any unfair advantage. However, they were selected so that the perceived difficulty of the documents used each year was broadly similar. So, although the data used each year were different, the method used to select the data should allow some relative conclusions to be drawn.

The results from the competitions show that the performance of state-of-the-art methods has improved. However, there remains significant room for improvement even in the methods submitted to the most recent competition. Furthermore, the segmentation metric results here present the weighted average performance over all region types. When looking at the results divided by region type, most methods perform significantly better on text regions than

they do on other types of region.

### 3.6 Discussion

The text-based performance evaluation methods were a novel approach to evaluating layout when direct information on the layout was unavailable. They provided evaluations based solely on the text output from an OCR method. Such methods have significant problems. Since the evaluations are based on the textual output of OCR, errors from that OCR are inter-mingled with errors resulting from layout analysis. Attempts are made to separate the two but there is no way to do this reliably since the two types of error are not independent.

The reliance on OCR output has several other disadvantages. Since it requires that all the data to be evaluated is textual, there is no way to evaluate pages containing non-textual content such as images or line-art. Even for text, the use of OCR means that evaluation can only take place if the text on the page is in a script which may be OCRed reliably. For other scripts for which OCR works less well or is not available yet, such documents may not be analysed.

The first true evaluations of layout analysis were produced by the pixel-based performance evaluation methods. These methods describe the contents of regions as the set of black pixels found in them. Such methods improved upon previous text-based evaluation methods in that they were capable of evaluating the full range of regions which might be found in a document, such as images, separators, line-art, etc. which had been missed by text-based evaluation methods.

The description of regions as their black pixel content was introduced because the typical documents and challenges in layout analysis at the time involved mainly simpler, textual pages which typically contained black content on a white background. Layout analysis methods at the time focussed more on simpler layouts. However, there has been a trend over the past decade of moving away from recognising black & white documents towards documents containing more colour. Also, in layout analysis, there has been a trend towards designing methods for segmenting more complex documents.

Given these trends, the assumption of pixel-based evaluation methods that the document's useful contents are contained in the black pixels of the image and the white pixels contain just useless background space, is less useful today. For documents which contain images, those images typically contain a full range of colours and the result of binarisation of such images is highly unlikely to lead to the useful content of the image being converted solely

into black pixels. It is more likely that the useful content of the image will be distributed between white and black pixels. Given this, the use of performance evaluation methods which rely solely on black & white images and assume that the only useful contents of the page are found in the black pixels, is likely to lead to misleading evaluation results.

Despite the problems with pixel-based performance evaluation methods, some of them contain useful ideas which are independent of the region representation. The approach published by Yanikoglu & Vincent allows a degree of flexibility in allowing the user to specify region type-specific weights. The method also permits area weighting which allows errors to be weighted according to the importance of the regions involved in the document.

A more recent development has been the region-based performance evaluation methods. Rather than access the image to determine whether or not something is useful content, such methods rely solely upon geometric matching of the ground-truth and segmentation. However, most of the methods published so far have been based on a bounding-box region representation. Bounding-box is insufficient to correctly represent the layouts of modern documents so performance evaluation methods based on it are limited so a subset of documents with less complex layouts.

One of the region-based methods, that proposed by Antonacopoulos and Brough, did not rely on the bounding box representation. Instead, it relied on a region interval representation. For the comparison method, the authors suggest that regions be matched by constructing a maximal polygon around each ground-truth region then checking whether or not segmentations fall within the maximal polygon. The method was never developed into a full comparison system. However, the region interval representation used demonstrates both accuracy and efficiency.

### 3.7 Summary

This section has described the past approaches to performance evaluation of layout analysis methods. The initial methods were based on a text-matching approach. Although these allowed the evaluation of the layout analysis modules of commercial OCR systems, the evaluation was not a truly independent evaluation of the layout analysis.

Following on from this, a number of image-based performance evaluation methods were described. These methods provided the first true evaluations of layout analysis by matching regions based on the black pixel contents of those regions. Such methods, however, are susceptible to changes in binarisation methods used and may ignore some useful parts of the document which are not represented as black. Such methods are less useful on pages are not

solely black and white or which contain images.

The final group of performance evaluation methods were the region-based methods. These enable evaluations to be performed even on colour documents. However, such methods are currently at an early stage of development. Many are based on a bounding box representation which is not suitable for describing modern, complex documents. One method introduced the concept of region intervals as a means of describing complex pages but still allowing for an efficient evaluation. The method was not completed but the global region interval concept which it introduced forms the basis of the new performance evaluation method presented in this thesis.



# Chapter 4

## Ground-truth and Datasets

### 4.1 Overview

In the introduction, it was stated that for any comparison to be made between the results of two different Layout Analysis methods, it is necessary for two things to be in place — a common evaluation method and a common dataset. The previous chapter discussed the previous performance evaluations methods which have been proposed. This chapter discusses the other pre-requisite — common ground-truth and datasets.

### 4.2 Ground-truth

#### 4.2.1 Definition

The area of performance evaluation in general involves evaluating how well a particular Image Analysis method performs on a given set of data. Typically, there exists some idea of what the perfect output of an Image Analysis method should be. The perfect output is termed the ground-truth. The task of performance evaluation then becomes the task of measuring how the actual output of the method compares against, and deviates from, the idealised output.

The ground-truth is often a human-entered perfect description of the optimum output of the Image Analysis method. In the case of evaluating Optical Character Recognition methods, this typically takes the form of a perfect text which is often typed separately by two different operators and the results combined to ensure the final result is free of human errors. In the case of Layout Analysis, the ground-truth contains the ideal layout of the page.

The availability of an accurate ground-truth is of fundamental importance for performance evaluation. For a true performance evaluation, the ground-truth must contain an accurate representation of the actual layout of the page as perceived by a human. Once this is in place, performance evaluation can be used to identify portions of a segmentation which deviate from this. Any deviations from this perfect representation must be errors.

## 4.2.2 Desirable characteristics

This section highlights some of the most desirable characteristics of a good ground-truth format.

### Accuracy

One of the chief characteristics desired in a ground-truth is its accuracy. This accuracy stems from two different areas. First, the format used for the ground-truth must be flexible enough to contain an accurate representation of the page. Secondly, the actual ground-truth as entered must represent the page accurately.

When designing a ground-truth format, there are some important technical considerations to make such as the format used for the region outlines. Some datasets in the past have simply used a bounding-box representation for regions. This may be adequate for relatively simple documents such as those found in older scientific journals.

When it comes to describing more complex regions, it is necessary to select a more complex region representation in order for the format to be able to accurately represent those pages. More modern ground-truth formats and datasets have settled on a polygon-based approach which is suitable for representing the majority of modern documents.

### Flexibility

As well as describing the geometric layout of the page, it is also desirable for a ground-truth format to be capable of storing additional meta-data about the page. Having just the geometric layout allows a performance evaluation method to detect merges and splits but does not enable a higher-level understanding to be gained.

On the simplest level, the ground-truth format should describe the type of each region. To enable more advanced performance evaluation, including region metadata such as the sub-type of the region (headline, drop capital, body text, etc.), the colour of the region contents and background, the size and orientation of text, etc.

In addition to providing for a more accurate evaluation, such additional metadata assist

in making a more application-specific evaluation. For instance, if a user is working on a document indexing application, then the regions of the page which will be most important are headlines, image captions, page number and by-lines. Given this, a tailored evaluation must allow the desired regions to be weighted more heavily than others, implying that any ground-truth which forms the basis of a flexible performance evaluation must contain such meta-data.

### 4.3 Desirable criteria of datasets

It has been mentioned that publicly-available common datasets are extremely important for a variety of reasons. For the results of different Layout Analysis methods to be comparable, it is necessary that the results are based on a common dataset. Similarly, in order to make an assessment of the maturity of the area, it is necessary that the datasets on which Layout Analysis methods are evaluated, are made publicly-available. If a developer evaluates their system using a private dataset, then the true performance of the system cannot be known. However, if the content of the dataset is known, along with its relative complexity, it is possible to make a useful assessment of the system's capabilities.

This section discusses some of the features desired in a Layout Analysis dataset then evaluates some of the currently-available datasets against these criteria in order to select a dataset upon which the performance evaluation method described in chapters 5 & 6 will be based.

#### 4.3.1 Representativeness of included document types

In evaluating general Layout Analysis methods, those which are aimed at segmenting general documents, it is an important feature of a dataset that it contains a wide selection of different types of documents which reflects the variety of documents which will be encountered in the real-world.

Some datasets focus mainly or exclusively on a single type of document, such as technical journal articles. While this may be useful for the specific purposes for which the dataset was produced, but may render the dataset unsuitable for evaluating on more general documents.

#### 4.3.2 Complexity

Datasets should contain documents of complexities similar to those likely to be found in real-world applications. Some datasets consist solely of articles from technical journals. However, such articles are typically much more restrictive in layout than other types of document and they rarely contain the complicated features which are more common in magazine pages.

Similarly, if a dataset contains only relatively simple documents which do not contain complex features, then the Layout Analysis method will not be tested to the full extent desired, meaning it may appear to perform better than it would when dealing with real-world documents and the method will not be tested on the features found in more complex documents.

### 4.3.3 Use of synthetic ground-truths

The process of ground-truthing large numbers of documents is a time-consuming and costly one. It is costly largely because the ground-truthing of scanned real-world documents requires the layout information to be input manually. Since the quality of the ground-truths greatly influences the quality of any evaluation based on them, it is typically necessary for the work to be performed or supervised by more experienced staff who are familiar to the area.

In order to reduce this cost, some have proposed methods to generate synthetic documents and accompanying ground-truths using a digital typesetting system such as L<sup>A</sup>T<sub>E</sub>X. Typically, such methods involve having a number of pre-designed layouts which are automatically filled with text. These methods have the advantage that they can be used to automatically generate large numbers of documents and, because the layout already exists in the computer, the ground-truth layout for the documents is already known. Such systems can vastly reduce the cost of ground-truthing since they eliminate the need for human operators manually zoning images. Alternatively, a much larger dataset could be produced for the original cost.

However, the problem with such approaches is that everyday documents are typically laid out manually by human operators. Given this, they often contain a wide variety of different features, some of which may be unique to a particular page. For instance, many magazine pages contain complex-shaped images around which the surrounding text wraps tightly. Given this, any system which automatically generates document layouts is unlikely to produce output which matches the complexity of documents found in the real world.

It should be noted that automatically-produced documents may be useful for some purposes. For instance, they may be useful as a low-cost source of data which may be used in testing for deficiencies in Layout Analysis methods. However, they will not allow for a true evaluation of all aspects of a Layout Analysis system.

### 4.3.4 Representativeness of document features

One of the key features desired of a dataset is its representativeness of real-world documents. This is essential in gaining a performance evaluation which reflects the true real-world ability

of a Layout Analysis method. Representativeness implies that the range of features will be similar to those found in the real world — both complex and simple features should be found as they are in general documents — and the range of documents should be similar to those which users are likely to want to digitise.

## 4.4 Datasets

This section includes a description of the available datasets in the area of Layout Analysis.

### 4.4.1 Nartker, Rice & Lumos

The ISRI OCR Performance Toolkit is a set of wide-ranging evaluation tools and a large dataset[17]. The dataset grew out of the work by the authors in evaluating OCR systems in the mid-1990s.

Today, the dataset contains 2,889 pages from a variety of different document types including Magazines, Newspapers, Business Documents and Annual Reports, and there are typically a large number of documents in each category. Each of the pages is scanned in 300dpi bi-level with 200dpi and 400dpi and 300dpi greyscale available for a majority of the documents. A small portion of the documents have also been faxed.

For each page, manually-entered layout information is available in the form of bounding boxes, and the complete ASCII text of each of the pages is available for researchers in OCR.

The dataset is over ten years old at the time of writing and some aspects of the dataset make it unsuitable for evaluating modern layout analysis methods. The ground-truths are only available in bounding-box representation which means they cannot accurately describe many modern documents. Similarly, there are no colour scans available.

### 4.4.2 Phillips, Chen, Ha & Haralick

One of the most widely-used document datasets in image analysis is the University of Washington CD-ROM Document Dataset, presented by Phillips, et al.[20].

Rather than being developed solely for use in page segmentation, the dataset was aimed more generally at researchers in the field of Document Analysis. The dataset contained information on the layout of the document for developers of Layout Analysis methods as well as textual ground-truths of the text regions contained in the database, for developers working on OCR-related research.

The dataset is also one of the largest datasets in the area, having around 1,600 journal pages in English and a smaller number in Japanese. The documents were taken from a variety of categories, although there is a significant emphasis on journal articles.

The source of the documents is varied. Part of the dataset consists of images scanned from physical documents. However, a significant part of the dataset was produced from synthetic documents using  $\text{\LaTeX}$ . For the reasons mentioned earlier, synthetic documents are less useful for performance evaluation as they are unlikely to contain the complex variety of features which appear in scanned paper documents.

A notable feature of the dataset is that a significant proportion of the documents in the dataset are also available in deliberately degraded form. This makes available a significant supply of data for researchers who are undertaking research into degraded documents. The documents of the dataset, if sourced from physical documents, have been photocopied or faxed. Where the document was synthetic, a printed copy was photocopied and faxed in order to obtain the degraded version.

The dataset contains ground-truths of regions, text lines and individual words using the bounding-box representation. This is ideally suited to the less complex technical articles which make up the majority of this dataset, but it does not allow for more complex documents to be represented. At the page and region level, the format used for the dataset contains a large amount of metadata about the content of the regions.

The dataset is quite widely used among researchers. However, the usefulness of the data in evaluating Layout Analysis methods on more modern documents is called into question by the age of the dataset and by the reliance on relatively simple document types such as technical articles and the presence of synthetic document images.

### 4.4.3 Sauvola & Kauniskangas

In 1999, Sauvola and Kauniskangas[22] published a CD-ROM document database known as the “MediaTeam Document Database II.” The database contains 500 scanned document images from 1978 and earlier and documents are taken from a wide variety of sources. Non-traditional sources include music and maps, which are quite dissimilar from standard documents in terms of layout analysis.

The wide breadth of document types comes at the expense of depth. Around half of the database is made up of journal articles but no other category contains a significant number of documents. Bounding boxes are used as the region representation for the dataset. There is a minimal amount of additional metadata for each document. However, they do include reading order information.

The dataset relies exclusively on older documents (from 1978 and before). There is a wide variety of different document types but the choices do not coincide with the types of documents which are usually analysed. The use of bounding boxes prevents more complex documents from being represented correctly.

#### 4.4.4 U.S. National Library of Medicine

As a project for the U.S. National Library of Medicine[27], a ground-truth dataset for layout analysis called Medical Article Records Groundtruth was created. This ground-truth database is aimed specifically at the digitisation of biomedical journal articles and improving the automated indexing capability. Given this, the dataset contains only the page/pages of articles which contain the abstract.

The dataset contains document images with ground-truths for both the page text and the layout. The layout has been ground-truthed at the character, word, line and zone level but all items are described by a bounding box only.

The selection of purely the first, and possibly second, pages of articles from biomedical journals implies that the contents will be quite simple in layout and, given this, the dataset relies on the bounding-box representation for regions. The simple, homogeneous layouts are unlikely to be able to highlight significant numbers of flaws in modern layout analysis algorithms.

#### 4.4.5 Suzuki, Uchida & Nomura

Between 2005 and 2006, Suzuki, Uchida and Nomura[24] presented a series of datasets of technical articles in the field of Mathematics, known collectively as InfyCDB. Rather than containing full document images, the articles were scanned and then segmented into individual characters or mathematical symbols. The images were then divided into individual images of each character or symbol for storage in the dataset.

The dataset comprises around 70 mathematical articles in English, German and French and a selection of other documents of various types including some Japanese. These images are intended to be used in developing character recognition methods for mathematical documents. Given this, each image has a ground-truth character and links to other characters in the same word.

The dataset is primarily aimed for use in OCR and, given this, does not include features which are necessary to allow use in layout analysis performance evaluation, such as region-level groundtruths or representations of non-textual regions.

#### 4.4.6 Todoran, Worring & Smeulders

In 2005, Todoran, Worring and Smeulders[26] presented a new dataset which was designed to provide a dataset for the growing area of colour image analysis. They point out the existence of datasets designed for black & white image analysis, such as the University of Washington dataset discussed earlier, but the lack of similar datasets for colour document analysis.

In order to fill this gap, they constructed a dataset of 1,000 pages which was formed solely from colour documents which were scanned from a variety of magazines.

For representing each document, an XML-based file format is used. Regions may be represented in a variety of ways from lines to polygons, depending on what is required for the region involved. The dataset contains some metadata such as the type, sub-type, colour and orientation of each region.

One notable feature of the dataset is that it extends the document representation into three dimensions by adding the concept of layers into the region representation. This allows for the representation of documents which cannot be decomposed into a discrete two-dimensional representation.

#### 4.4.7 Antonacopoulos, Karatzas & Bridson

In 2006, the author and Drs. Antonacopoulos & Karatzas[7] announced a new ground-truth format and dataset specifically designed for performance evaluation of layout analysis methods.

The ground-truth format used for the dataset is an XML-based format which has been specifically designed for the purpose of performance evaluation on complex, modern documents. Regions are represented by isothetic polygons<sup>1</sup>, which allow regions with complex shapes to be stored accurately. Each region has an associated region type which may be:

- Text
- Maths
- Image
- Graphics
- Line Drawing
- Separator

---

<sup>1</sup>Polygons having only horizontal and vertical edges.



- Noise

Those types are further subdivided into more specific types, such as heading, caption, drop capital and paragraph for text regions. Each region has an extensive amount of region-level metadata associated with it such as font size, region orientation, foreground colour, background colour, text language and script.

The metadata provided were selected specifically with a view to enabling more precise performance evaluation to be carried out using the data. Region location, types, sub-types and font size which enable performance evaluation methods based on the data to take these into account when evaluating the problems in a given segmentation.

The dataset was initially presented in 2006 and is scheduled for public release in 2009. The dataset currently includes several hundred documents of various types including magazine pages, technical articles and advertisements. All images have been scanned from real-world documents. The dataset is currently under development and sponsorship by Google is currently funding the scanning of a further thousand documents in a variety of types.

The dataset has been used, prior to and following its announcement, as the foundation of the ICDAR 2003, 2005 and 2007 Page Segmentation Competitions.

An example colour image and accompanying XML ground-truth file from the dataset are given in figure 4.1. The XML file has been abridged to omit multiple regions of the same type and long lists of region outline co-ordinates. In the included portions of the XML file, the document metadata as well as the descriptions of six individual regions can be seen. At the top are document-level metadata concerning the number of pages described in the XML file (1), the filename of the document image, a count of the regions of each type and the size of the document image on which the ground-truth is based.

The individual regions displayed correspond to visible regions in the document image. The first region, of type `Separator`, corresponds to the thick horizontal line at the very top of the document image. The next region is of type `Text` and sub-type `Header` and corresponds to the lone word in the header of the document image, "CRIME." The following region is also of type `Text` but this time is of sub-type `Paragraph` and corresponds to the very first paragraph of text in the page. The next region, of sub-type `Caption` describes the main portion of the caption under the image in the document, the following region the image itself and the final region the page numbers located at the bottom-left corner of the document.

The amount of metadata available can be seen in the figure, with 10 items of meta-data given for each text region in the image, 4 for each image region and 3 for each separator region. It can be seen that the format allows, and the dataset provides, an extensive amount of metadata specifically tailored for use in evaluating Document Layout Analysis methods.

A published paper is reproduced in Appendix 1 which gives more detail on the design of the ground-truth format and the issues involved in creating the dataset. In Appendix 2 is an XML Document Type Definition for the ground-truth format.

## 4.5 Discussion

The above has given a description of all of the known datasets in the area of Layout Analysis. Some of the datasets were designed specifically with Layout Analysis in mind, while others were aimed more generally at Image Analysis. This section will give a description of how well the existing datasets meet the criteria specified above.

When evaluating general Layout Analysis algorithms, it is desirable to have a selection of different document types which represent well those that will likely be encountered in the real-world. Some of the datasets discussed are specific to a given document type. The University of Amsterdam dataset, for instance, contains only Magazine Pages while the InftyCDB and MARG datasets contain exclusively technical articles from their fields, Mathematics and Medicine.

Another significant factor in the selection of a dataset is the complexity of the documents involved. When one of the motivations of performance evaluation is to highlight deficiencies in existing layout analysis methods, it is necessary to use a dataset containing documents which are likely to contain the complex features which will highlight problems with existing methods. So, it is desirable for datasets to have a significant proportion of more complex documents such as magazine pages. Again, the InftyCDB and MARG datasets consist largely of technical articles so are unlikely to contain many more complex features.

The accuracy of the ground-truths is a crucial factor in performance evaluation. A significant factor contributing to accuracy is the region representation used for the dataset. Some of the older datasets, such as the ISRI and University of Washington datasets, rely on the bounding-box representation. This means that either the documents selected must be simple enough to fit into such a schema or, if more complex documents are included, then the ground-truths must not be accurate.

One of the datasets mentioned above, that proposed by Todoran, Worring & Smeulders, allows for the use of layers in the document representation. Some pages from more complex documents contain regions which overlap, meaning that the documents cannot be fully segmented into a two-dimensional representation. The addition of layers allows such complex documents to be represented. This will be useful for many more complex magazine pages. However, no known Layout Analysis method can detect or output such representations.

## CRIME

videotape a woman using the toilets. He was charged with a misdemeanor and fined \$75.

After Obara's arrest in connection with Lucie's disappearance, a sharper image of his personal life emerged. In contrast to his occluded public persona, Obara's private obsessions are delineated in excruciating detail. He wrote journals and dictated audio diaries on cassette tapes starting in the early 1970s. Police have leaked some of Obara's most incriminating entries to Japanese reporters like Mamoru Kadowaki of the *Weekly Shincho* magazine. According to Kadowaki one of Obara's most troubling entries, presented in vaguely poetic form, includes the lines, "Women are only good for sex. I will lie to them. I will seek revenge. Revenge on the world."

In 1983 his journals make their first references to "conquer play," a euphemism, prosecutors say, Obara used to describe his assaults on women. Journals between 1983 and 1995 include the names of more than 200 women, beside which Obara wrote code words, 29 of which, investigators believe, refer to drugs. Police recovered more than a dozen different varieties of drugs from Obara's homes—from sleeping pills to chloroform to human growth hormone. In his diaries, he mentions drugs frequently, at one point declaring, "I am so bored with pot, hash and LSD." But if there were any doubts about his main interest, these were dispelled by an entry in which he stated, "I can not do women who are conscious."

WHEN POLICE ARRESTED OBARA in early October, he initially denied knowing who Lucie Blackman was. Police found blond hairs that matched Lucie's in one of Obara's seacoast condominiums, then a roll of film that contained pictures of her taken near the same dwelling. But without a body, they were unable to bring charges against him. Police culled Obara's videos and journals for other victims. The three foreign hostesses agreed to cooperate with the prosecution, and Obara was charged with several counts of rape.

In a rambling November letter to the media, Obara countered: "These ladies who are supposed to be victims are all foreign hostesses or sex club girls. Many took cocaine or other drugs in front of me, and all of them agreed to have sex for money." The women told a different story. He met them in hostess clubs, invited them on *dohans*, drove them to the sea and lured them into his condominium using a variety of methods. He invited one woman over, offering to cook her dinner. He asked another to accompany him to a party later in the evening. In the meantime they could watch a Mariah Carey concert on TV at his condo. Another, he simply drove to his building and asked to help him carry up some boxes from his car.

Once he got them inside, he would keep the conversation light. Inevitably he would urge them to try a rare wine which he would tell them came from India or the Philippines. To account for the funny taste of this drug-laced beverage, Obara told his victims it contained special herbs. There was one victim he coaxed into making a "good luck" toast that required her to down the en-

tire glass in a single gulp. If she didn't drink it all, he warned her, she wouldn't have good luck.

Videotapes then tell the rest of the story. According to court documents filed by the prosecution, the tapes show Obara lugging unconscious women onto his bed. He must have struggled with some. Lucie was a good 5 cm taller than he was. Police have leaked details of his having tied some of the women down, penetrating them with foreign objects and sodomizing many of them. He would assault most victims for 12 hours or more. To insure they remained unconscious, he would place a cloth soaked in a drug, known to be chloroform in at least one case, over their mouths. He captured his assaults on tape using professional video equipment and lights. One of his victims sustained burns when he left a hot light too close to her body.

Obara's women would awaken 24 or even 48 hours later, sick and disoriented from the drugs. Chloroform is toxic to the liver and can be fatal. Each of the women recounted waking up vomiting, being unable to stand, crawling on her hands and knees to the bathroom. Few had any idea what had happened. Obara would sometimes dress them back in their own clothes before they regained consciousness. Then, he would always have a story. He told one woman: "You are such a fun girl. You drank an entire bottle of vodka." He told another there had been a gas leak. The woman with the burned skin, who had been unconscious on and off for more than 36 hours, was told she had become drunk and fallen over.

In addition to the witnesses against Obara, police discovered hospital receipts linking him to a former Roppongi hostess, an Australian named Carita Ridgeway. In 1992 he took a gravely ill Ridgeway to Hideshima hospital, telling nurses she had eaten bad shellfish. Ridgeway was erroneously diagnosed as suffering from liver failure as a result of eating seafood tainted with the

virus that causes hepatitis. After she died a few days later, Obara even comforted her parents when they came to take her body home. Due to an administrative fluke, Ridgeway's liver had been preserved at Tokyo Women's Hospital, where the autopsy had originally been performed. Last autumn, after Obara came under investigation for Lucie's disappearance and his other assaults, medical examiners tested Ridgeway's liver for chloroform, which proved to be present in toxic levels. Obara was charged in connection with her death.

If anything, the arrest of Obara proved even more agonizing for the Blackmans. In addition to what they had learned about his assaults on other women, police leaked disturbing details of his activities during the first days of Lucie's disappearance. Late on the night of July 2, Obara called area hospitals asking how to treat a victim of a drug overdose.

On July 3 Obara purchased a chainsaw, cement mix and other tools from a hardware store. That afternoon, the manager of Obara's seaside condominium in Miura called police to report a tenant who was behaving suspiciously. Even in the terse language of police reports leaked to the media, the scene that afternoon at Obara's apartment has a Hitchcock-like cast. Obara had



**A FAMILY'S DESPAIR**  
Tim Blackman, with daughter Sophie, displays a poster with Lucie's picture at a press conference shortly after she vanished

```

<?xml version="1.0" encoding="UTF 8" standalone="no"?>
<!DOCTYPE document SYSTEM "groundtruth.dtd" []>

<document>
  <document_summary no_pages="1"/>
  <page page_id="0" image_filename="mp00039bw.tif">
    <page_summary no_text_regions="18" no_image_regions="1"
      no_line_drawing_regions="0" no_graphic_regions="0"
      no_table_regions="0" no_chart_regions="0"
      no_separator_regions="6" no_maths_regions="0"
      no_frame_regions="0" no_noise_regions="1"/>
    <page_pixel_size width="2331" height="3135"/>
    <separator_region id="0" sep_orientation="0.000"
      sep_colour="Black" sep_bgcolour="White">
      <coords no_coords="4">
        <point x="2324" y="107"/>
        ...
      </coords>
    </separator_region>
    <text_region id="1" txt_orientation="0.000"
      txt_reading_orientation="0.000"
      txt_reading_direction="Left_To_Right" txt_type="Header"
      txt_colour="Black" txt_reverse_video="No" txt_indented="No"
      txt_primary_language="English" txt_primary_script="Latin"
      txt_bgcolour="White">
      <coords no_coords="4">
        <point x="1344" y="143"/>
        ...
      </coords>
    </text_region>
    ...
    <text_region id="3" txt_orientation="0.000"
      txt_reading_orientation="0.000"
      txt_reading_direction="Left_To_Right" txt_type="Paragraph"
      txt_colour="Black" txt_reverse_video="No" txt_indented="No"
      txt_primary_language="English" txt_primary_script="Latin"
      txt_bgcolour="White">
      <coords no_coords="4">
        <point x="1168" y="238"/>
        ...
      </coords>
    </text_region>
    ...

```

```

<text_region id="9" txt_orientation="0.000"
  txt_reading_orientation="0.000"
  txt_reading_direction="Left_To_Right" txt_type="Caption"
  txt_colour="Black" txt_reverse_video="No" txt_indented="No"
  txt_primary_language="English" txt_primary_script="Latin"
  txt_bgcolour="White">
<coords no_coords="4">
  <point x="1724" y="1919" />
  ...
</coords>
</text_region>
...
<image_region id="13" img_colour_type="24_Bit_Colour"
  img_orientation="0.000" img_emb_text="No"
  img_bgcolour="Red">
<coords no_coords="4">
  <point x="1729" y="1097" />
  ...
</coords>
</image_region>
...
<text_region id="23" txt_orientation="0.000"
  txt_reading_orientation="0.000"
  txt_reading_direction="Left_To_Right" txt_type="Page_Number"
  txt_colour="Black" txt_reverse_video="No" txt_indented="No"
  txt_primary_language="English" txt_primary_script="Latin"
  txt_bgcolour="White">
<coords no_coords="4">
  <point x="182" y="3009" />
  ...
</coords>
</text_region>
...
</page>
</document>

```

Figure 4.1: The colour image and associated XML ground-truth file of magazine page 39 from the PRImA Document Layout Dataset.

It is important for a significant amount of meta-data to be present in ground-truths since this is crucial in performing certain kinds of evaluation and, where more extensive metadata is included, may permit more accurate evaluations to be made. For instance, for an evaluation method to identify misclassified regions, it is necessary that the ground-truth contains region type information at the very minimum. To give a more complex example, if it is desired to know the severity of a merge between text regions, it is necessary to know the direction of the text involved as well as the region order.

When selecting a dataset to be used for public evaluations, it may be desirable to select one which is currently in active public use. The most widely-used of the datasets described here are the University of Washington and the PRImA Datasets. The University of Washington dataset has been available for a long time and is widely-used among researchers. The PRImA Dataset is newer than the University of Washington dataset but has already been used for the ICDAR Page Segmentation Competitions.

Given these characteristics, the first four datasets mentioned in this chapter, ISRI, University of Washington, MediaTeam OULU and MARG, are not suitable for a general layout analysis performance evaluation context since they rely solely on bounding box representations, meaning they cannot describe documents with the accuracy desired. The InftyCDB dataset is unsuitable because it contains only technical articles in mathematics so lacks the representative selection of documents which are desired. Similarly, the University of Amsterdam dataset contains only magazine pages so, although it would include more complex documents, it would not be representative of the real-world applications of general document Layout Analysis methods.

The PRImA dataset is the only dataset which meets most of these criteria, since it was designed recently and specifically for layout analysis on general documents. The selected documents are of a variety of types. The region representation used is the isothetic polygon, allowing complex features to be described. Moreover, the format used contains a variety of meta-data selected with this application in mind. Additionally, the dataset has seen some public use in three ICDAR Page Segmentation Competitions, will be publicly released shortly and is under ongoing development, meaning that the range and number of documents are growing.

## 4.6 Summary

This chapter has given an overview of the desirable characteristics of ground-truths and datasets. Following this, a description of the available datasets in the area was given and a discussion

was presented of how well the existing datasets match up to these criteria, leading to the selection of a dataset upon which the performance evaluation method described in the following two chapters will be based.





# Chapter 5

## Interval Comparison

### 5.1 Overview

The previous chapter contained an overview of ground-truths and datasets, the desirable features of each, a discussion of the available datasets in the area and a discussion of how these datasets meet the needs of performance evaluation. This chapter presents a new means of accurately comparing polygonal ground-truth and segmentation representations. This is expanded upon in chapter 6 to provide a full performance evaluation method with the aim of fulfilling the goals laid out in chapter 1.

### 5.2 Introduction

In chapter 3, the previous approaches to Performance Evaluation are discussed in detail. One of the problems highlighted of pixel-based evaluation systems is that they rely on the assumption that in a black & white image, page contents are black and the page background is white. In documents containing regions in different colours on backgrounds of different colours, however, this is not the case. Furthermore, where pages contain images, light portions of those images will be discarded as useless background information, while only the darker portions will be evaluated.

The pure region-based evaluation methods were presented as an improvement on this situation. However, most of the region-based evaluation methods presented to date rely on a bounding-box region representation which allows comparisons to be made relatively efficiently but which prevents pages containing more complex layouts from being evaluated accurately. One region-based method was presented which was based on a region interval

representation. The particular method was never developed into a complete evaluation system. However, the region interval representation used there has the possibility to be used as the basis of an efficient but accurate performance evaluation method. This is presented here.

This chapter presents the region comparison aspect of the system. The chapter begins with a discussion of the different region representations which are available, giving the advantages and disadvantages of each including the region representation selected for the method described here. Following that, a description is given of the algorithm for converting a polygon representation into a region interval one and then the algorithm for comparing two document layouts once they are in region interval format. To conclude the chapter, the efficiency of the comparison method is discussed.

### 5.2.1 Examples

The method described in this chapter is designed to operate efficiently on the full document layouts of real-world documents with complex layouts. In order to best describe the operation of the algorithm, however, the diagrams included in this chapter will initially focus on smaller, artificial layouts, allowing the diagrams to be clear while still illustrating the workings of the algorithm. At the end of the chapter, similar diagrams are given showing the system operating on the full complex documents for which it was designed.

The system described here is one which compares an automatically-segmented document layout against a manually-entered ground-truth (or perfect) layout for the document. So, the diagrams of this chapter will use as a running example the artificial ground-truth and segmentation layouts depicted in Figure 5.1.

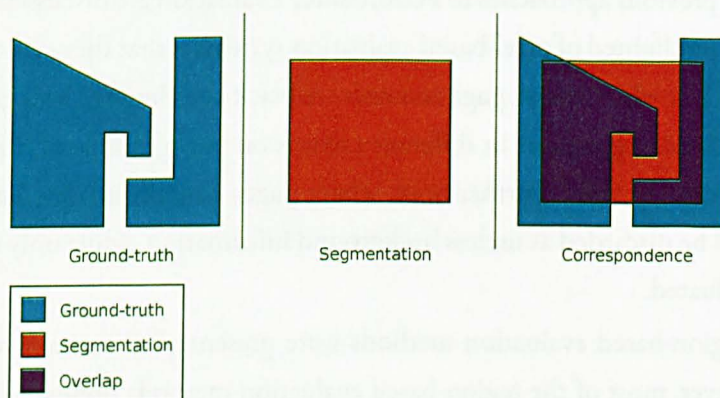


Figure 5.1: The ground-truth and segmentation layouts used in examples in this chapter to illustrate the approach.

The colours shown in Figure 5.1 are used in the other figures in this chapter (unless otherwise noted) with blue representing the ground-truth region alone, red representing the segmentation region alone and purple representing areas where the ground-truth and segmentation are overlapping.

At the end of the chapter, many of the figures will be repeated using full document examples rather than the artificial example depicted here. The same colours are used in those larger examples.

### 5.3 Region representation

Essential to the accuracy of performance evaluation of layout analysis is the region representation used. This affects many aspects of the performance evaluation process including the accuracy of the evaluation and the speed which which it can be performed.

Many of the earliest performance evaluation methods relied on a bounding box representation of regions. This representation has several advantages and disadvantages. Foremost among the advantages is that it provides an extremely efficient means of comparison. Similarly, its simplicity allows methods using it to be implemented in a relatively short amount of time.

However, the chief disadvantage of the bounding box representation is its lack of accuracy or, put another way, the relatively small proportion of the set of all document pages which could be accurately represented using this representation. The bounding box representation would be most suited to older documents with relatively simple layouts.

The bounding box representation, however, is much less suitable for describing documents with more complex layouts. Considering particularly modern magazine pages, the layouts are often manually created using computer typesetting systems with a variety of complex features. Consider the example document in Figure 5.2, taken from an issue of *Time* magazine. The figure shows a page which is largely text-based but which contains an irregularly-shaped image around which the text is made to flow. Such a layout could not be represented accurately using a bounding box representation.

A bounding box representation may also have problems dealing with documents containing less complex layouts which might ordinarily have been represented using bounding boxes but which are rendered unsuitable due to the presence of artefacts introduced during the scanning process, such as skew and shear.

When a document image is said to be skewed, this means that a page which is ordinarily



Figure 5.2: An image of a real-world magazine page containing a complex layout, taken from the PRImA Layout Analysis Dataset (mp00167).

straight has been inadvertently scanned at a slight angle. There are a variety of document image analysis methods, and document layout analysis methods in particular, which have been specifically designed to operate well even with the presence of skew in an image. For instance, the layout analysis method presented by O’Gorman [18] uses the angles between neighbouring connected components to detect the skew angle of the page and takes this into account during the later steps of the process.

Indeed, in some applications, it is necessary for methods to operate with quite large levels of skew. In the careful mass scanning of documents by trained staff, it is likely that some small amount of skew will be present in almost all documents scanned, though large amounts of skew will be rare in such circumstances. In other applications, such as the mass automated scanning of postal mail for postcode recognition, significant levels of skew are likely to be commonplace.

Given that skew represents a significant challenge in document image analysis, it is desirable for any region representation scheme where accuracy is a priority to allow for the representation of documents containing skew. The bounding box representation, for smaller amounts of skew, may be able to provide a close approximation of the true layout. However, as skew angles become non-negligible, it is unlikely that a bounding box representation would be able to fully represent individual regions without missing any portion and without also including some part of a neighbouring region. This is illustrated in Figure 5.3.

A similar issue occurs in the presence of shear. Shear is another artefact which may be introduced during the scanning process. This occurs during the scanning of a large book where



Figure 5.3: A rectangular region shown (a) without skew, (b) with 0.5 degrees of skew and (c) with 5 degrees of skew, all with bounding boxes overlaid.

the pages of the book curve in towards the spine, making it impossible to place the entire page flat on the scanner bed. This produces an optical distortion as the portions of the page close to the spine, being further away from the scan-head, appear smaller in the scanned image than the outer portions of the page which lie flat on the scanner glass.

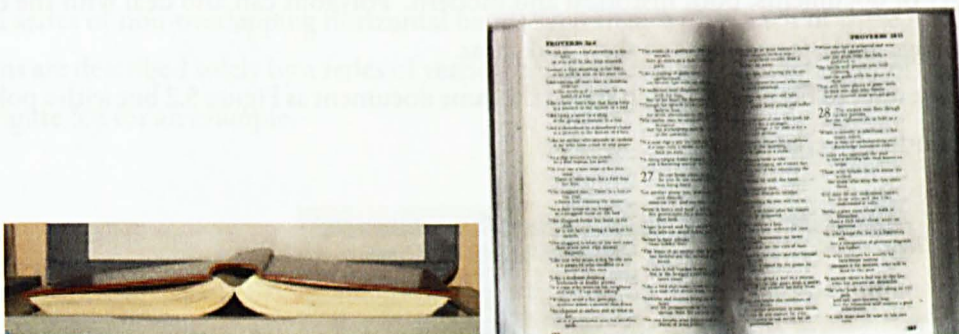


Figure 5.4: a) A large book lying on a flatbed scanner, and (b) an example of the optical distortion produced by this, known as shear.

Given that the problem of shear is limited to bound books, one approach taken by some has been to simply remove the pages of the book from its binding, eliminating the problem entirely. However, this is not an option for books of historical value or books with significant value to their owners, particularly those in libraries.

Some have proposed a modified bounding box representation which allows the box to be rotated by specifying, as well as the position and size of the box, a skew angle. Such a representation would enable bounding boxes to be used in documents where skew is present while providing the same representative capability as they do on documents without skew. Such representations do not improve descriptiveness in the presence of shear and complex

layouts and have not become widely used in layout analysis.

Given the inability of bounding boxes to represent the significant proportion of documents containing non-trivial layouts and their inability to deal with the problems of skew (in their unmodified form) and shear, it is clear that the bounding box representation is quite unsuitable as the basis of a modern ground-truth format or performance evaluation method.

## 5.4 Polygon representation

Given the shortcomings of the bounding box representation detailed in the previous chapter, there was a movement in the late 1990s away from using a bounding box representation towards more detailed region representation schemes.

Polygons suffer from none of the drawbacks of bounding boxes mentioned in the previous section. They can provide a much more accurate representation of regions rather than merely an approximation.

Where bounding box representation struggles to accurately represent the more complex layouts which have become more frequent in recent decades, polygons can describe the vast majority of documents, both historical and modern. Polygons can also deal with the document image analysis problems of skew and shear.

Please refer to Figure 5.5 which shows the same document as Figure 5.2 but with a polygon region outline overlaid.

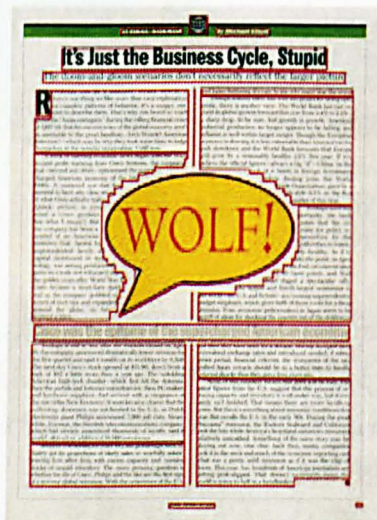


Figure 5.5: A document with a more complex layout with isothetic polygon region outlines overlaid.

It should be noted that, although polygons have a superior expressive capability, they also

necessarily complicate the performance evaluation task as comparing arbitrary polygons is significantly more difficult a challenge than comparing two rectangles. Given this, there have been no known attempts to date to present a region-based performance evaluation method based on a polygon approach.

## 5.5 Algorithm

The direct comparison of two arbitrarily complex polygons is a difficult task, such that no known region-based performance evaluation method has so far been published which uses such an approach. This section presents a method which converts polygon image regions into an intermediate region representation called region intervals then uses this representation as the basis of a comparison between the ground-truth and segmentation layouts.

### 5.5.1 The Region Interval representation

Region intervals provide a slightly unusual layout representation whereby the page is split into a series of non-overlapping horizontal bands such that, within each of these bands, the regions are described solely by a series of vertical bands taking up the full height of the band. See Figure 5.6 for an example.

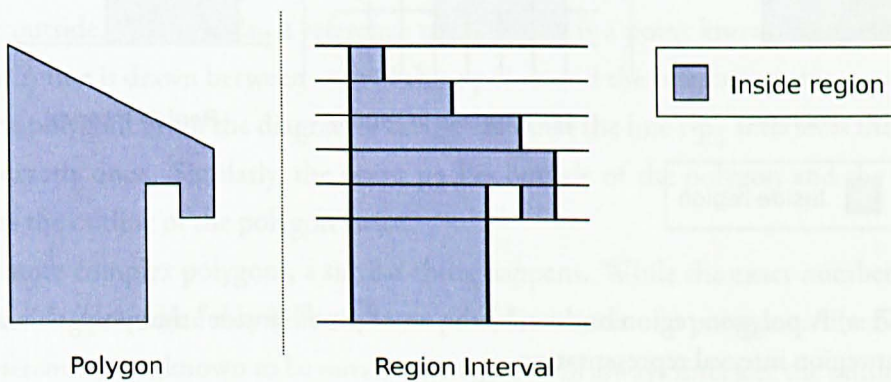


Figure 5.6: a) A polygon region outline, and b) the same region shown in the region interval representation.

Region intervals have several advantages over polygons. Chiefly, from a computational point of view, it is significantly more efficient to check whether a given point lies within a region using a region interval representation than the same region represented using a polygon.

Similarly, the comparison of two arbitrary polygons is non-trivial, while the comparison of two region interval representations may be performed efficiently as is detailed in this section.

At a first glance, it may appear that the region interval representation is not as flexible as a polygon representation in terms of the number of regions which can be accurately represented. Naturally, in a continuous domain, this would be correct. The use in the region interval representation of horizontal bands split into vertical sections precludes the representation of non-horizontal or vertical boundaries which may be present in an arbitrary polygon.

Images, however, are not a continuous domain; they are inherently discrete. A polygon region describing an area in an image can be said to describe the set of pixels contained within. Given this, it is possible to construct a region interval representation which contains exactly the same set of pixels as any polygon.

As an example, please refer to Figure 5.7.

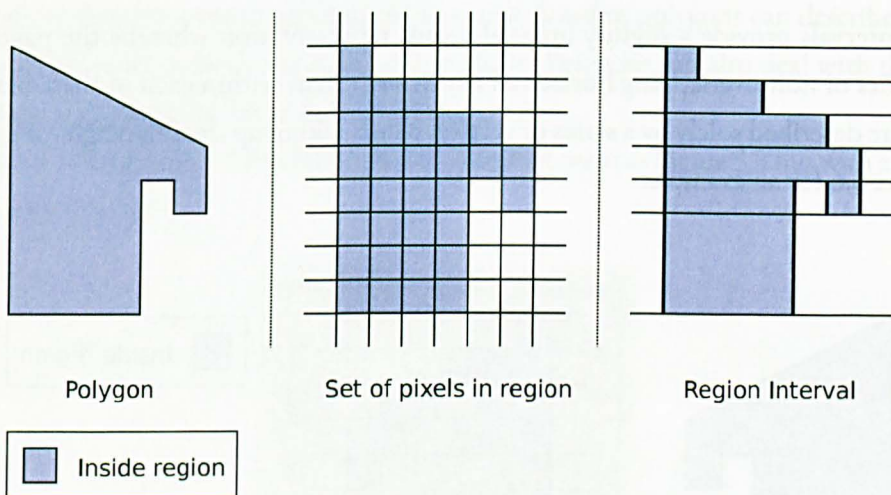


Figure 5.7: a) A polygon region outline, b) the set of pixels “inside” that polygon and c) the equivalent region interval representation.

The next section describes a method of converting any region’s polygon into its equivalent region interval representation. Following that, an efficient method is presented for comparing two region interval representations representing the ground-truth and segmentation. This comparison will be used in the next chapter as the basis of the performance evaluation method presented herein.



### 5.5.2 Converting a polygon to Region Intervals

The polygon or a similar type, such as isothetic polygons or bounding boxes, is used in most major layout analysis methods. Typically, this is stored as an ordered list/array of co-ordinates in the case of a polygon or isothetic polygon. For a bounding box, typically only the co-ordinates of the top-left and bottom-right corners are stored, from which the remaining two corners can be calculated.

Given this representation alone, it is difficult to check whether given areas lie within the polygon. However, there exists a method to check whether a given point lies within the polygon. Take Figure 5.8 as an example.

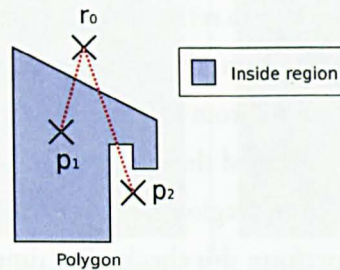


Figure 5.8: A polygon with three points: one inside the polygon ( $p_1$ ), one outside ( $p_2$ ) and a reference point which is also outside the polygon ( $r$ ).

The figure shows a polygon and a pair of points for comparison, one inside the polygon and one outside. There is also a reference point which is a point known to be outside the polygon. A line is drawn between each of these points and the reference point.  $p_1$  is a point inside the polygon. From the diagram it can be seen that the line  $r$ - $p_1$  intersects the polygon outline exactly once. Similarly, the point  $p_2$  lies outside of the polygon and the line  $r$ - $p_2$  intersects the outline of the polygon twice.

For more complex polygons, a similar thing happens. While the exact number of intersections with the outline may differ, when a point lies inside of a polygon, a line between it and a reference point known to be outside the region will always intersect the outline an odd number of times. Similarly, a line between a point outside of a polygon and a reference point also outside of the region will intersect the outline of the polygon an even number of times or not at all.

This fact can be used to detect whether any given point lies inside or outside of a polygon. In the remainder of this section, we will make use of this to convert a polygon region into its region interval equivalent.

Since it is possible to find, for any given point, whether this point lies inside or outside of

a polygon, then it is possible to construct a brute-force approach which performs this test for every pixel in an image to determine the complete set of pixels which lie inside the polygon.

Such an approach would not be efficient, however. Assuming an average document image contains around 6 million pixels, then checking for each pixel whether it intercepts each of the edges in (on average) half of the regions in the document is likely to be an extremely time-consuming process.

However, by making use of the principal discussed earlier, it is possible to greatly improve upon this brute-force approach.

The region boundaries mark the entrances to and exits from regions. The approach described above for detecting whether or not a point lies within a polygon depends upon this fact for its efficacy. If a line is drawn from a reference point outside the region to an unknown point, then all the points on that line from the reference point to the first region boundary must lie outside the region. All points from the first region boundary to the second must lie inside, and so on. So, if it is ascertained that a given pixel lies within a given region, then all of the neighbouring pixels up to the region boundary must also lie inside of the region. Therefore, it is necessary only to perform this check a few times, then all of the neighbouring pixels may be marked as belonging to the same region. The following describes how this is performed in the method.

Please refer to Figure 5.9. Initially, a two-dimensional array is constructed which is two pixels wider than the region and two pixels taller. This allows the region to be centred in the array with a one-pixel border around the region which allows the area surrounding the region to be filled in one pass, reducing the total amount of computation needed. Each pixel in this grid is capable of taking one of three values: outside, inside and unknown, with each pixel being initially set to unknown.

Next, the region outline is rasterized into the grid and marked as inside. This is performed by looping through each of the vertices in the polygon and drawing a line between each vertex and the next. In the case where this line is horizontal, i.e. the y-co-ordinates of the two vertices are equal and the x-co-ordinates are different, the pixels between (and including) these two points are marked inside. A similar operation occurs where the edge joining the two vertices is vertical.

Where the edge formed between the two vertices is neither horizontal nor vertical, a slightly different algorithm is used. First, a check is performed to determine whether the line is closer to vertical (i.e. more than  $45^\circ$  from horizontal) or closer to horizontal (i.e. less than  $45^\circ$  from horizontal).

In the case where the line is more horizontal, as illustrated in Figure 5.10, the method

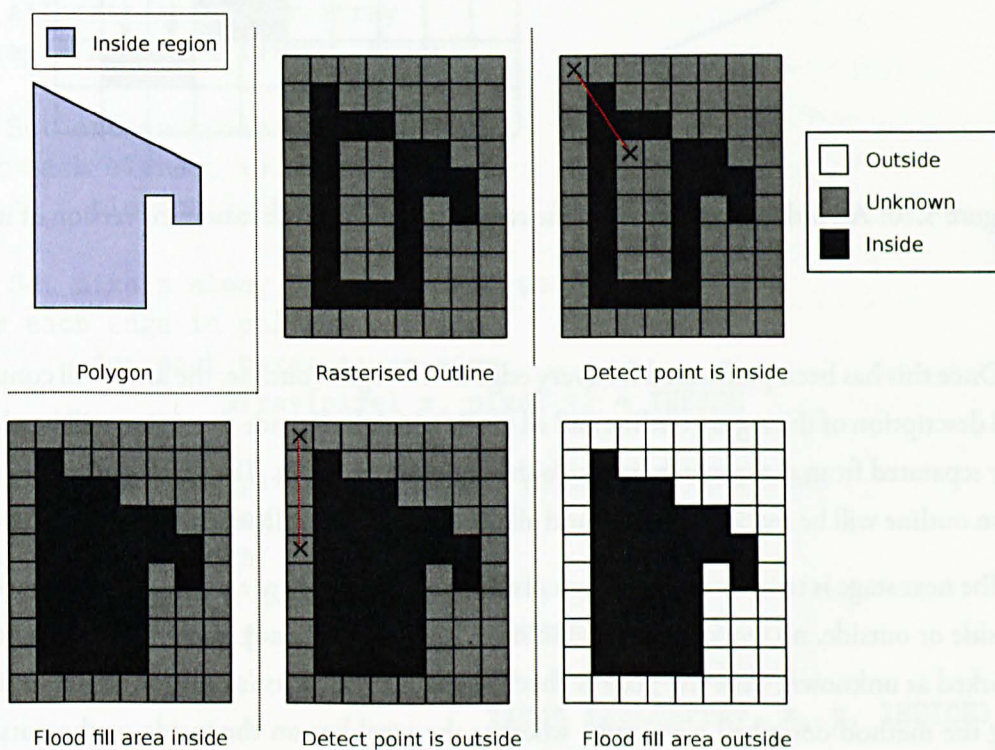


Figure 5.9: a) A polygon region outline; b) the array with the outline of the polygon marked as inside; c) a point selected for checking along with the reference point  $r$  and a line drawn between the two; d) the inside of the region marked as inside.

cycles through each column in the x-direction, calculates the corresponding y-position of the line and marks the closest pixel to this.

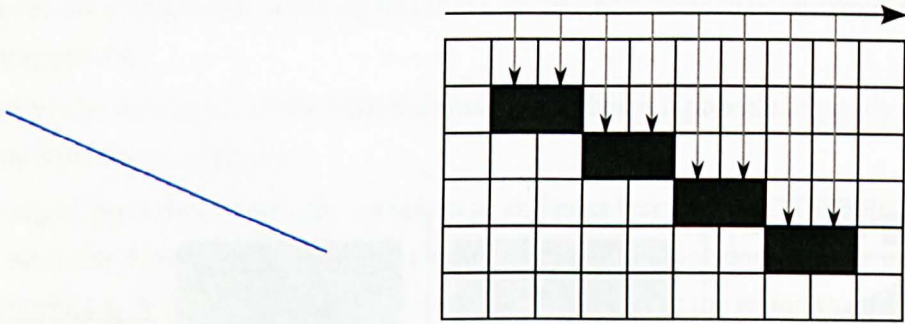


Figure 5.10: An individual edge from the region outline and the rasterized version of it.

Once this has been performed for every edge in the region outline, the array will contain a full description of the region outline and all pixels which lie outside the region will be physically separated from those which lie inside the region by this line. The pixels making up the region outline will be marked as inside and all other pixels will still be marked as unknown.

The next stage is to cycle through each pixel in the array. Where a pixel is already marked as inside or outside, no work need be done since its status is already known. Where a pixel is marked as unknown, then the pixel is checked against the region's polygon to determine, using the method described previously, whether the pixel lies on the inside or the outside of the region. The result is then marked into the array not only for that pixel but, using a recursive process, for all the pixels connected to it. This process is then repeated for all of the other pixels in the image.

The result of this is that all of the pixels in the array will be marked as being inside or outside of the region and the process will be performed efficiently by removing unnecessary computations. A pseudo-code description of the conversion of an arbitrary polygon to a two-dimensional array is given in Figure 5.11.

Referring to Figure 5.7, it can be seen that the second stage of this process has been completed, i.e. the polygon region outline has been converted into an array marking each pixel as being inside or outside of the region. Next, this will be converted into a region interval description for the single region and then for the whole document.

```
// Allocate space for array
array = new array[width, height]

// Set entire array to UNKNOWN
for each element in array
    element = UNKNOWN

// Set pixels along polygon edges to INSIDE
for each edge in polygon outline
    for each pixel along edge
        array[pixel x, pixel y] = INSIDE

// For remaining pixels, check if inside polygon then
// flood-fill using result
for x = 1 to width
    for y = 1 to height
        if array[x, y] is UNKNOWN
            if pixel x, y is in polygon
                flood fill(array, x, y, INSIDE)
            else
                flood fill(array, x, y, OUTSIDE)
```

Figure 5.11: Pseudo-code for converting an arbitrary polygon to a two-dimensional array.

### 5.5.3 Converting the array into a Region Interval representation

Initially, the array will be converted into a trivial region interval representation for this region containing one band for each row in the array. While this could be simplified at this stage, this may cause problems for reasons described later in this section, so simplification will be delayed until later in the process.

First, an empty region interval representation is created with a number of bands equal to the height of the region, each exactly 1-pixel high, but without any contents at the moment. Note that, although the array from the previous section contained a 1-pixel empty border around the region, this was added merely to speed up processing during that stage and is not carried over into the region interval representation.

The region interval representation should contain, within each band, a number of entrance and exit points which denote the beginnings and endings of regions. These can be inferred from the array produced in the previous section.

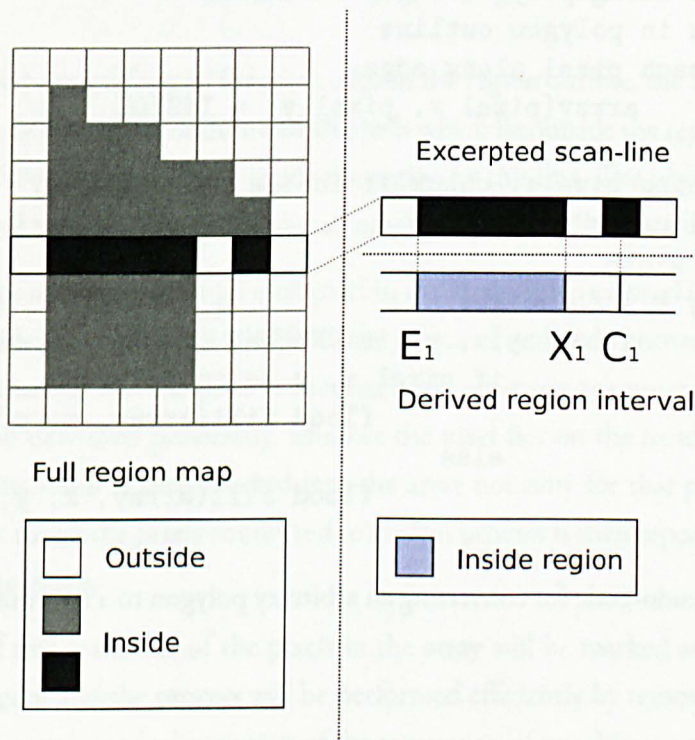


Figure 5.12: a) An array showing the pixels inside and outside of a region, and b) the related region interval containing a region entrance, an exit and a combined (1-pixel) entrance and exit.

This process begins by moving along each row in the image separately. A record of the

current state is kept. At the beginning of the row, this marker is initialised to reflect the fact that we are outside of the region. The method moves along the pixels in the row. When the first pixel marked inside is encountered, this marks the beginning of the region. So, an entrance point is added into the region interval with the x-position the same as the current pixel. The state marker is updated to reflect that we are now inside the region.

The loop continues moving along the line until a pixel marked outside is reached which represents the exit from the region, so an exit point is marked in the region interval representation at the x-position of the last inside point.

It should be noted that in some ground truths and segmentations, there may be regions which are just 1-pixel wide (for example, some very narrow vertical separator regions) or other regions, similar to that shown in Figure 5.12, which contain portions which are just 1-pixel wide. This may be considered an entrance and exit point which occur at the same x-position. Having these described as separate entrance & exit points causes later stages to be unnecessarily complicated, so a test is performed here on any region exit to check if the x-value is the same as the previous entrance and, if so, to replace this entrance and exit with a special “combined” entrance-exit point.

This process continues until the end of the line is reached, at which point the interval related to that line contains a complete representation of that line. Then, the process begins again for the next line. A pseudo-code description of the conversion of the two-dimensional array to a region interval representation is given in Figure 5.13.

In this way, the method makes a complete simple region interval representation of the polygon region.

#### 5.5.4 Application to a complete document

The previous section describes the conversion of a single region into a simple region interval representation. However, it will be rare for a document to contain just one region. In fact, even simple documents will likely have ten or more regions while average documents will likely contain significantly more.

When dealing with a whole document, the process described in the previous section must be repeated separately for each individual region. One modification is made when dealing with multiple regions in order to increase performance. Rather than allocate the two-dimensional array for each individual region, which would reduce efficiency, one single two dimensional array is allocated which is as wide as the widest region (plus two pixels for the border) and as tall as the tallest region (plus two pixels). This allows the same memory to be used for all regions rather than repeatedly allocating and deallocating memory.

```

// Make array of bands, one for each scan-line
bands = new array[region height]

CurrentState = OUTSIDE

// Loop through array and add region entrances & exits into bands

for y = 1 to region height
  for x = 1 to region width
    if array[x][y] is INSIDE and CurrentState is OUTSIDE
      CurrentState = INSIDE
      Add change point (x, entrance) into bands[y]
    else if array[x][y] is OUTSIDE and CurrentState is INSIDE
      CurrentState = OUTSIDE
      if last change point's x equals x - 1 and type was entrance
        Change type to combined
      else
        Add change point (x - 1, exit) into bands[y]

```

Figure 5.13: Pseudo-code for converting the two-dimensional array into a region interval.

Once each polygon region has been converted into a region interval format, these can be transferred into a master region interval representation which stores information for the whole document. This is similar to the individual region versions but it contains one interval for each row in the image which the document layout represents.

The entrance and exit points for individual regions may be transferred into the document region interval representation but, in doing so, two changes must be made. In the single region representation, there was no need to store any information about the region from which the change points belong since the representation contained only one region. This must be added to each change point in the full document representation to identify each change point as belonging to a given region. Secondly, the relative positions must be translated from the smaller single-region representation to the larger full-document representation since the individual region region interval representations represent only the area of the document containing that specific region.

Once these have been transferred into the master region interval representation, it is desirable to simplify this. While originally an interval of 1-pixel height was created for every row in the image, it will often be the case that one interval contains exactly the same number of region entrances and exits as the previous interval, with each point in exactly the same position as the equivalent point in the previous interval.



In this case, the second interval can be simply deleted and the first interval expanded to cover the height of the two.

This process begins by cycling through each interval and comparing it to the next. First, the number of *change points* (entrance, exit & combined points) in the two intervals is compared. If these counts are different, then the two intervals must also be different. If the numbers are the same, then a comparison needs to be made of the change points in the two intervals. They must be checked in order to test if the position, type and region are the same as the equivalent in the next region. If they are different, then the two intervals are different and the process can move onto the next pair of intervals. If they are identical, this can be repeated for the next pair of change points. If all the change points have been found identical between the two intervals, then the intervals must be identical. In that case, the second of the two intervals can be deleted and the first expanded to fill the space of both.

Once one interval has been deleted and the interval above expanded to fill its place, the expanded region is then checked against the new next interval. Once this has been repeated for each consecutive pair of intervals, then the region interval representation will have been simplified as much as is possible.

Figure 5.14 shows the artificial document layout introduced at the beginning of this chapter in its original polygon representation and then in its region interval equivalent. There are several features of note in the diagram. Note that the diagonal upper edge of the leftmost region, has been split into three separate single-pixel-high intervals,  $I_0$ – $I_2$ . Note also that interval  $I_6$  is larger than the others as a result of the merging process described in this subsection. Since this portion of the document can be perfectly described with just one interval, any more would slow down the comparison process described in the next section while resulting in no gain in accuracy.

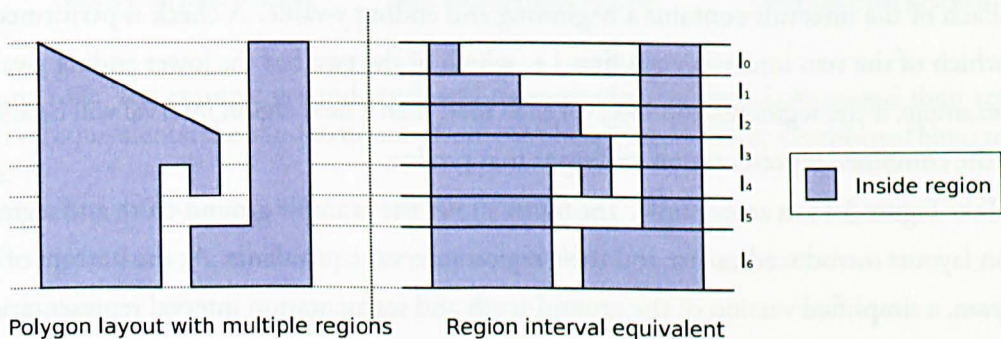


Figure 5.14: The example document layout introduced at the beginning of this chapter and its region interval equivalent.

### 5.5.5 Comparison of ground-truth and segmentation

The description so far has concentrated on the processing of a single document layout, and the conversion of that layout into a region interval representation.

In performance analysis, there will usually be two document layouts involved. One, the ground-truth is a human-entered description of the actual layout of the page. The second, the segmentation, is the automatically generated layout as detected by a document layout analysis method. The latter will likely have mistakes and it is the goal of performance analysis to locate, quantify and describe these mistakes and the process by which this begins is the comparison of the ground-truth and segmentation layouts.

The process so far describes the conversion of a single document layout into region interval representation. This needs to be performed twice, once for the ground-truth and once for the segmentation. So, at this stage, there are a pair of region interval representations, one representing the ground-truth and the other the segmentation.

Referring to Figure 5.15, it can be seen that the two interval representations are quite different. Some regions in the ground-truth have been split in the segmentation, while others have been merged. Due to these differences, it should be noted that the intervals are quite different in each, in terms of number, positions and heights. In order to compare the two, therefore, it will first be necessary to align the two sets of intervals.

A new combined interval representation will be generated which will contain a set of intervals derived from the ground-truth and segmentation intervals and which will contain the change points from both the segmentation and ground-truth.

The creation of this combined representation begins as follows. Two pointers are created which will point to the current ground-truth and segmentation intervals, respectively. Initially, they are set to point to the topmost interval in each document.

Each of the intervals contains a beginning and ending y-value. A check is performed to see which of the two intervals ends first, i.e. which of the two has the lower ending y-value. For example, if the segmentation interval ends first, then a new region interval will be added into the combined representation ending at that y-value.

Take Figure 5.15 as an example. The figure shows the example ground-truth and segmentation layouts introduced earlier and their region interval equivalents. At the bottom of the diagram, a simplified version of the ground-truth and segmentation interval representations is given showing just the horizontal bands of each.

Using this figure as an example, initially, the ground-truth pointer will be set to  $G_0$  and the Segmentation pointer set to point to  $S_0$ , since those intervals are the topmost intervals in their respective layouts. A check is performed to see which of the two ends first (i.e. nearest

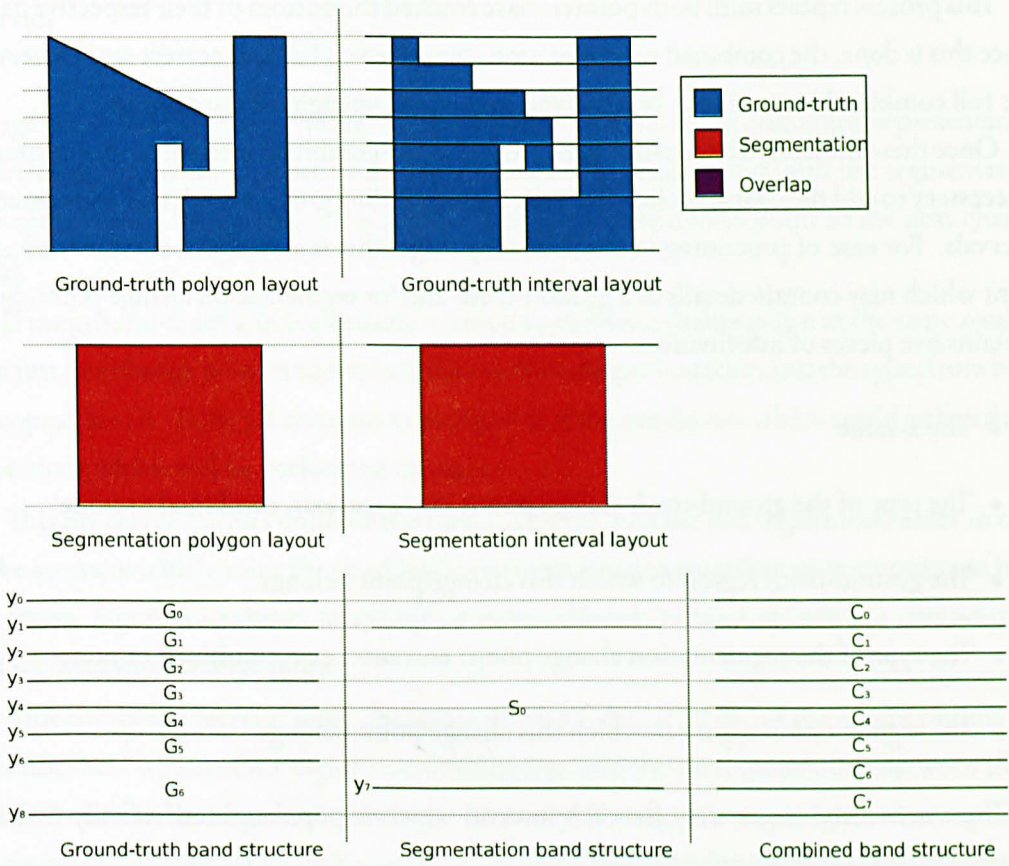


Figure 5.15: The example ground-truth and segmentation polygon layouts and their region interval equivalents; the band structures from each being used to form a combined band structure.

the top). In this case, interval  $G_0$  ends at  $y_1$ , while we are not yet inside interval  $S_0$ . So, an interval ( $C_0$ ) is added to the combined representation from the top of the page ( $y_0$ ) to  $y_1$ , and the ground-truth pointer is incremented to point to the next ground-truth interval,  $G_1$ .

The process will then repeat. After the previous step, the ground-truth pointer points to  $G_1$  and the segmentation pointer still points to  $S_0$ . Of the two,  $G_1$  ends first (at  $y_2$ ), so a new interval,  $C_1$ , is added to the combined representation ending at  $y_2$  and the ground-truth pointer is incremented to point to  $G_2$ .

This process repeats until both pointers have reached the bottom of their respective pages. Once this is done, the combined representation should have a full and correct set of intervals. The full combined intervals can be observed in the bottom-right of the diagram.

Once these horizontal bands have been created in the combined interval representation, it is necessary to add the change points from the corresponding ground-truth and segmentation intervals. For ease of processing in the following stage, this is stored as a combined change point which may contain details of a ground-truth and/or segmentation change point. So, it contains five pieces of information:

- The x-value
- The type of the ground-truth change point: entrance, exit, combined or none
- The ground-truth region to which this change point belongs
- The type of the segmentation change point: entrance, exit, combined or none
- The segmentation region to which this change point belongs

These are stored in an array for each interval which is populated individually for each interval in the manner described below.

Both the ground-truth and segmentation intervals contain an array of change points sorted by ascending x-value. The change points are added to the combined interval representation in order by reading the two arrays from left to right.

First, two pointers are created which will be used to point to the current change point in the ground-truth and segmentation intervals, respectively. These are initialised to point to the first (i.e. leftmost) change point in each of the two arrays. Each of the two will have an x-value and type (entrance, exit or combined).

A check is performed to detect which of the two has the lower x-value. If one has a lower x-value than the other, then that will be added into the combined representation first, copying the x-value and type and setting the other type to none. So, for example, if the ground-truth

Condition	Action
$x_{GT_i} < x_{Seg_j}$	Add $GT_i$ into combined interval, will Segmentation component null
$x_{GT_i} > x_{Seg_j}$	Add $Seg_j$ into combined interval, will Ground-Truth component null
$x_{GT_i} = x_{Seg_j}$	Add a single change-point into combined interval, representing both $GT_i$ and $Seg_j$

Table 5.1: Conditions met & actions performed while adding change points into combined intervals

change point has the lower x-value, a change point is added to the combined representation with the ground-truth type set to the same value as the original and with the segmentation type set to none. The ground-truth pointer is then incremented to point to the next change point.

If the ground-truth and segmentation arrays both have a change point at the same x-value, then just one change point is added into the combined representation and the types from both are copied into it. Table 5.1 contains a description of the conditions which could be faced and the actions which will be performed on each.

This process continues until both pointers have reached the last (rightmost) entry in each of the arrays, at which point the combined interval contains a complete representation of both the arrays. Figure 5.16 shows an example combined band containing both the ground-truth and segmentation change points.

This combined interval representation represents the positions of the region outline but does not show which of the regions are overlapping, and the correspondences between them. For this, one further stage of processing is necessary. During this stage, a geometric representation will be derived which will contain a complete description of each pair of overlapping regions.

The process begins by reading the intervals from the top of the page to the bottom and, within each interval, from left to right. Initially, two pointers are created which will point to the current ground-truth region and the current segmentation region, respectively. At the beginning of the page, since we must not be inside of any region, these pointers are set to null to reflect this.

Next, a loop is performed over all the change points in the current interval, from left to right. The first change point reached must not be an exit since we are already outside of all regions at the beginning of the page. So, the first change point must be either an entrance or a combined entrance-exit and it may be from either the ground-truth, the segmentation

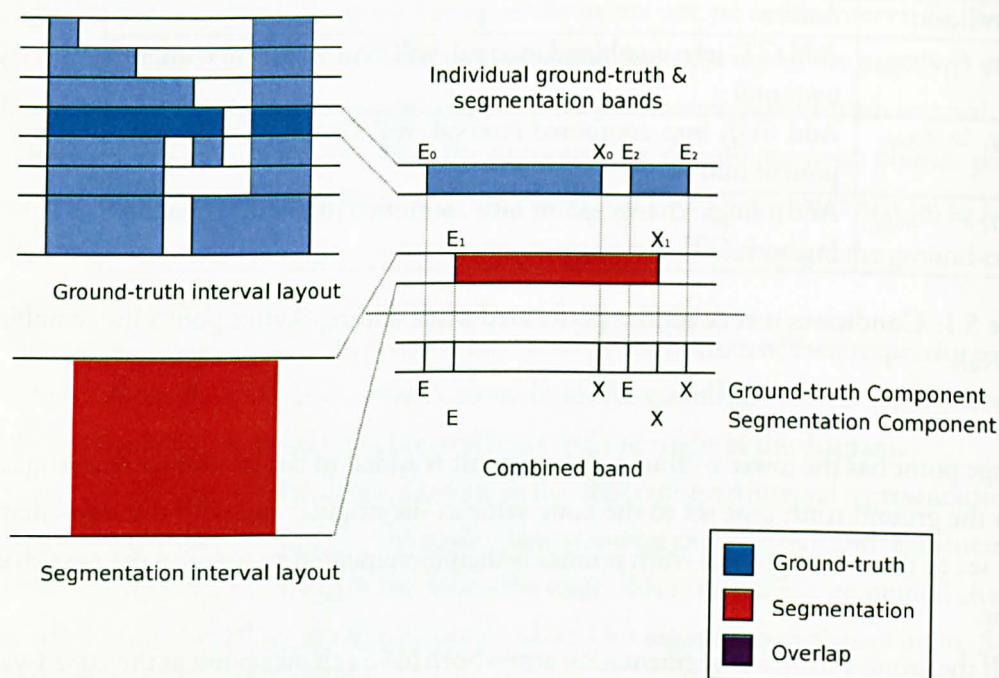


Figure 5.16: Overlapping ground-truth and segmentation bands from the example document layouts and the combined interval created from them.

or both. As we enter that region, the relevant pointer is updated to point to that region. So, for example, if a ground-truth entrance is reached, the pointer to the current ground-truth region is made to point to this region.

We then move on to the next change point. The type of this change point dictates the action to be performed next. For example, if we are now inside a ground-truth region and we encounter a ground-truth exit, then this marks the end of the current ground-truth region. So, the area from the entrance to this exit (inclusive) is added in to our geometric representation for that ground-truth region and the null segmentation region, signifying that this portion of the page was missed in the segmentation.

Alternatively, the next change point might be a segmentation entrance. In this case, the area beginning at the previous change point and ending *just before* the current change point is added into the geometric representation corresponding to that GT region and the null segmentation region. However, since we are already inside of a GT region and we are now entering a Segmentation, the area from now until the next change point will be added into a new geometric correspondence between the current GT region and the next Segmentation region.

The following will present a brief example of this process referring to Figure 5.17.

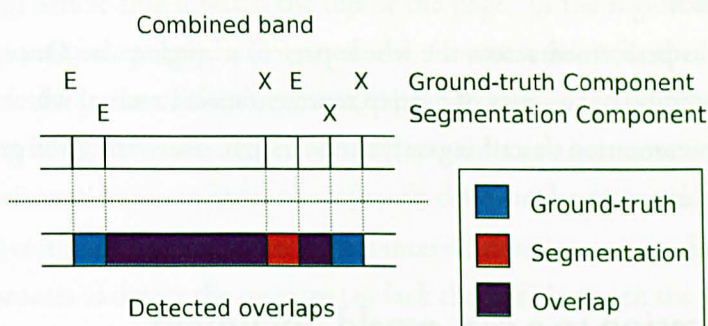


Figure 5.17: The excerpted combined interval from the previous diagram used to detect overlaps between the ground-truth & segmentation.

The diagram shows a series of six change points from the single interval excerpted from the example ground-truth and segmentation layouts. The method begins at the left of the page with the pointers to the current ground-truth and segmentation regions set to null, indicating that, to begin with, we are inside neither a ground-truth nor segmentation region. It begins moving across the page from left to right. The first change point encountered is a ground-truth entrance point. The pointer to the current ground-truth region is updated to the ground-truth region being entered and  $StartX$  is set to the current  $x$ -coordinate.

The next change point encountered is a segmentation entrance point. As we reach this point, the pointer to the current segmentation region is null. This means that the area from  $StartX$  to just before the current point was part of a ground-truth region and no segmentation region. A new overlap representation is created to describe the overlap between the given ground-truth region and no segmentation region and the area up to this point is added into it. Since we have reached a segmentation entrance point,  $StartX$  is updated to the current  $x$ -value and the pointer to the current segmentation region is updated to the new segmentation region.

Following this, the next change point encountered is a ground-truth exit point. The area from  $StartX$  up to *and including* the current  $x$ -value is added into the overlap between the given ground-truth and segmentation regions.  $StartX$  is updated to the *following*  $x$ -value and the current ground-truth region is again set to null.

Following this, another ground-truth entrance point is encountered marking the beginning of a new ground-truth region. The area from  $StartX$  is added as an overlap between the current segmentation and the null ground-truth region and the current ground-truth region is set to the newly-encountered ground-truth region. This continues until the end of the current interval is reached and then continues with the following intervals until the bottom of

the page is reached.

This process is performed across the whole page in a single pass. Once the process has completed, the method has a series of overlap representations, each of which comprises a region interval representation describing only the overlap between the given ground-truth and segmentation region.

## 5.6 Application to a real-world document

As described in section 5.2.1, the diagrams used to illustrate the method to this point were based on artificially-contrived example ground-truth and segmentation layouts which were designed to allow the key features of the method to be illustrated visually in relatively small spaces.

While the method would likely work with such examples, it was principally designed to be used with the type of complex document layouts which are likely to be encountered by researchers in Document Layout Analysis today. This section contains diagrams similar to those presented earlier in this chapter but, rather than featuring the artificial example layouts shown earlier, they depict the actual output from the method on a complex document selected from the PRImA Page Segmentation dataset, Magazine Page 42.

The output from the method on Magazine Page 42 of the PRImA Page Segmentation Dataset is shown in Figure 5.18. To the left of the diagram is the original (colour) document image. Two arrows lead from this to the manually-entered ground-truth data at the top (depicted here in blue) and, at the bottom of the diagram, to an example automatic segmentation (depicted here in red) created from the image using the White Tiles segmentation method discussed in section 2.4.3. Both the ground-truth data and segmentation are first shown in their original polygon format.

To the right of the ground-truth and segmentation polygon layouts is the actual result of using the method described earlier in this chapter to convert the polygon representations into region interval representation. One can notice firstly that the portions inside the region (marked in blue for the ground-truth and red for the segmentation) are identical to those inside the regions in the original polygon versions of the layout, as would be expected if the conversion process were functioning correctly.

In the region intervals diagrams, the horizontal black lines denote the boundaries of the horizontal bands of the region interval description. Referring specifically to the ground-truth region interval representation, it can be seen that the height of the horizontal bands varies dramatically across the page, depending upon the complexity of the page at that point. Take, for



example, the large article title towards the top of the page. In the region interval representation, this is represented entirely by just one band. Contrast this with the large, irregularly-shaped graphic at the centre of the document, especially the diagonal bottom-left corner of the graphic. It can be seen that the region interval representation here contains a large number of very short horizontal bands in order to accurately describe the diagonal line.

The ground-truth and segmentation region interval descriptions are then used in the final step of the process to detect the overlaps (or lack thereof) between the ground-truth layout and segmentation. This is displayed in the rightmost portion of the diagram. Here, the colours denote the overlaps, if any. Where a portion of the document is inside a ground-truth region but not a segmentation region, the diagram is coloured in blue, the colour used here for ground-truth regions. Where the reverse is true and a portion of the document is inside a segmentation region but not a ground-truth region, the segmentation region colour is used (red). Finally, where a portion of the page is inside both a ground-truth region and a segmentation region (the majority of the page in this instance), it is coloured in purple, a mixture of red and blue.

From this final part of the diagram, it is possible to see the differences between the ground-truth layout and the automatically-detected segmentation. Again, referring to the large title near the top of the page, it is possible to see that the title is correctly contained within one ground-truth region while the automatic segmentation has caused the title to be split into several smaller chunks corresponding to the words in the title. This is due to the use of global thresholds in the white tiles segmentation method, as discussed in section 2.4.3.

Looking towards the upper-left of the main body text, one can see that another mistake has been made in the opening drop-capital of the text. First, the two letters of the drop-capital (“J”), which are contained in a single region in the ground-truth, have been split in the segmentation into two separate regions. Secondly, to the right of the drop capital, there is an area of red which in this case shows that the letter J from the drop-capital has been merged with the first paragraph of body text.

## 5.7 Efficiency

Part of the reason for using region intervals as the foundation for the performance evaluation method was that they provide an extremely compact format for region representation which allows the comparison process to be performed extremely efficiently. This section examines the efficiency of the representation and the comparison method based upon it using real-world data.

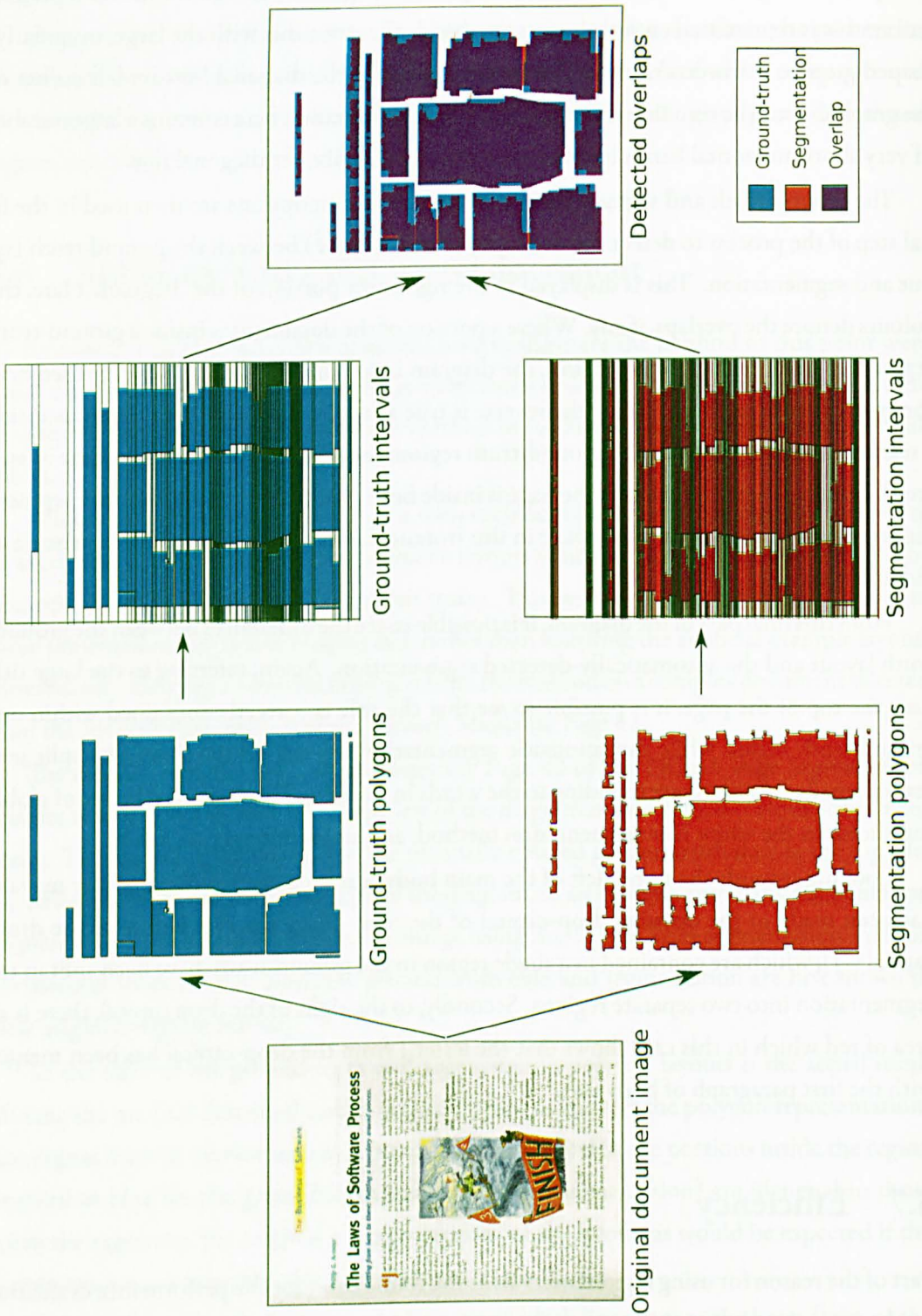


Figure 5.18: The operation of the method on a real-world document image, magazine page 42.

The main alternative region representation used in previous performance evaluation methods which still allows for the same descriptive capability as the region intervals described here is an image-based format. So, the examples described here will give the efficiency as compared with image-based performance evaluation methods. Bounding box-based methods are not considered here because they are unable to accurately describe complex document layouts.

Rather than base these comparisons on artificial data which may have little relevance in real-life evaluation systems, the comparisons presented here are based on the PRImA Page Segmentation dataset described in section 4.4.7 in order to see how the representation and comparison method perform on real data. The specific documents used will be the evaluation data used for the ICDAR 2005 Page Segmentation Competition described in section 3.4.3 and also used for the evaluation in section 7.3.4.

The discussion will be divided into space and time efficiency. A large part of the efficiency of the comparison method derives from the region interval representation used. As it will be seen from the subsection on space efficiency, the region interval representation allows the full layout of the document with full accuracy but in a much more compact form. In the section on time efficiency, it will be seen how the compact data representation greatly improves the efficiency of comparison.

### 5.7.1 Space-efficiency

With the region interval representation format, the entire page is divided into a number of horizontal bands. The location of regions in the page is simply denoted by a series of entrance and exit marks inside the horizontal bands. Table 5.2 shows all the data items required for the region interval representation.

Field	Size (bytes)	Description
NoBands	2	The number of horizontal bands
<b>For each band:</b>		
UpperY	2	The upper Y-coordinate of the band
LowerY	2	The lower Y-coordinate of the band
NoChangePts	2	The number of change points in the band
<b>For each change point:</b>		
Type	1	The type of the change point (Entrance, Exit or Combined)
XPos	2	The X-coordinate of the change point

Table 5.2: A description of the fields in the region interval representation along with their sizes.

Other methods use images as their region representation format. In this representation,

an image is allocated of the same dimensions as the original document image. Then, each region is allocated a unique ID (which may be thought of as a colour) and the pixels inside each of these regions are coloured with that colour. Table 5.3 gives a list of the data items required in the image representation.

Field	Size (bytes)	Description
Width	2	The width of the image
Height	2	The height of the image
<b>For each pixel:</b>		
PixelVal	2	The ID/colour representing the region of this pixel

Table 5.3: A description of the fields in the image representation along with their sizes.

In order to compare the relative space efficiency of the two representations, the amount of memory used is calculated for each of the two representations for each of the documents in the evaluation dataset for the ICDAR 2005 Page Segmentation Competition. The results may be found in table 5.4.

From the table, it can be seen that in every case, the region interval representation is dramatically smaller than the image-based equivalent. Over the entire set, the region intervals versions of the documents are approximately 99.98% smaller than the equivalent image-based representations.

This is relatively unsurprising since the image representation encodes, for every single pixel in the original document image, the region to which it belongs while the region interval representation records only the beginnings and ends of regions.

### 5.7.2 Time-efficiency

The interval comparison method presented in this thesis is also highly time efficient and this is largely due to the region representation upon which it is based.

Since image-based region representations have the size of a full image and each pixel's region membership is recorded individually, this means that in order to compare two document layouts, it is necessary to inspect the colour of each pixel in both the ground-truth and segmentation files.

With the region interval representation however, as only the entrances and exits to regions are recorded, it is only necessary to compare these entrances and exits rather than inspecting each individual pixel. This again dramatically reduces the number of individual comparisons which must be performed by the evaluation method.

Type	ID	Size		Relative size
		Image	Region Intervals	
Advertisement	1	16,968,998	629	0.004%
Magazine Page	1	15,191,514	5,756	0.038%
Magazine Page	4	15,157,132	14,624	0.096%
Magazine Page	9	14,826,724	3,008	0.020%
Magazine Page	11	15,092,724	926	0.006%
Magazine Page	21	14,738,940	1,802	0.012%
Magazine Page	27	14,915,100	1,256	0.008%
Magazine Page	31	14,703,080	1,700	0.012%
Magazine Page	39	14,615,374	1,079	0.007%
Magazine Page	42	15,185,830	7,331	0.048%
Magazine Page	86	14,728,192	1,079	0.007%
Magazine Page	90	14,778,390	749	0.005%
Magazine Page	137	17,194,756	2,135	0.012%
Magazine Page	138	16,973,884	1,241	0.007%
Magazine Page	139	17,065,066	2,768	0.016%
Magazine Page	160	14,683,924	941	0.006%
Magazine Page	161	14,986,792	2,336	0.016%
Magazine Page	177	14,952,868	2,066	0.014%
Technical Article	12	18,333,396	746	0.004%
Technical Article	16	17,875,706	1,151	0.006%
Technical Article	19	18,083,254	17,561	0.097%
Technical Article	20	17,283,842	686	0.004%
Technical Article	22	18,017,284	839	0.005%
Technical Article	23	18,567,876	824	0.004%
Technical Article	26	17,916,958	1,070	0.006%
Technical Article	27	17,113,008	1,859	0.011%
<b>Total</b>		<b>419,950,612</b>	<b>76,162</b>	<b>0.018%</b>

Table 5.4: The size of each document in the evaluation dataset of the ICDAR 2005 Page Segmentation Competition, using image representation and region interval representation.

The evaluation here will be performed again on the evaluation dataset of the ICDAR 2005 Page Segmentation Competition. Here it is necessary to have a set of sample segmentations to compare against the ground-truths. Here, the segmentations output by the BESUS method will be used as, of the four entrants to the ICDAR 2005 Competition, they were closest to average in layout complexity, the main influence on speed for this method.<sup>1</sup>

Table 5.5 contains details of the number of comparisons performed by the image-based comparison system as compared with the region interval-based one. It can be seen that, due to the significantly lower amount of data required by the region interval representation, the comparison method based on it is also significantly more efficient in terms of the number of comparisons performed, with 99.89% fewer comparisons.

It is interesting to note that, although the amount of data required to represent a complete document was reduced by 99.98%, the number of comparisons performed is only 99.89% fewer. This difference is due to the process described in section 5.5.5 — due to the necessary differences between ground-truth and segmentation, the combined interval representation will always have more horizontal bands than either the ground-truth and segmentation structures. The greater number of bands causes the number of comparisons made to increase proportionally.

Despite this, it can be seen that the more efficient storage enabled by the region interval representation allows the comparison between the ground-truth and segmentation to be performed more efficiently.

## 5.8 Discussion

This section has presented a novel method for comparing two polygon document layouts, producing a geometric description of all the overlapping pairs of ground-truth and segmentation regions. The process by which this is performed uses relatively simple transformations which are designed to retain all the detail of the original layouts, while being performed in an efficient manner. The following chapter describes how this geometric description is used as the basis of a powerful performance evaluation system.

---

<sup>1</sup>Conversely, image-based comparison methods are not influenced by layout complexity.

Type	ID	Comparisons		Relative Comparisons
		Image	Region Intervals	
Advertisement	1	8,484,497	11,774	0.139%
Magazine Page	9	7,413,360	10,708	0.144%
Magazine Page	11	7,546,360	8,484	0.112%
Magazine Page	21	7,369,468	14,451	0.196%
Magazine Page	31	7,351,538	11,132	0.151%
Magazine Page	39	7,307,685	3,836	0.052%
Magazine Page	86	7,364,094	15,810	0.215%
Magazine Page	90	7,389,193	7,560	0.102%
Magazine Page	137	8,597,376	7,001	0.081%
Magazine Page	138	8,486,940	8,949	0.105%
Magazine Page	139	8,532,531	6,966	0.082%
Magazine Page	160	7,341,960	12,114	0.165%
Technical Article	12	9,166,696	7,074	0.077%
Technical Article	16	8,937,851	5,374	0.060%
Technical Article	20	8,641,919	4,730	0.055%
Technical Article	22	9,008,640	12,565	0.139%
Technical Article	23	9,283,936	10,664	0.115%
Technical Article	26	8,958,477	5,508	0.061%
Technical Article	27	8,556,502	10,715	0.125%
<b>Total</b>		<b>155,739,023</b>	<b>175,415</b>	<b>0.113%</b>

Table 5.5: Using the evaluation set of the ICDAR 2005 Page Segmentation and BESUS segmentations, the number of comparisons performed by an image-based comparison method compared with the region interval comparison method.





# Chapter 6

## Evaluation System

### 6.1 Overview

The previous chapter described the method of comparing the polygonal ground-truth layout against the polygonal segmentation layout. The result of this is a geometric description of the overlaps between groundtruth and segmentation regions. This chapter uses this geometric description as the basis of a performance evaluation system fulfilling the requirements discussed in Chapter 3.

### 6.2 Identification of errors

#### 6.2.1 Error types

Following the processing described in the previous chapter, there is a set of region intervals for each overlap between a ground-truth region and a segmentation region, as well as cases where a ground-truth region overlaps the page background in the segmentation layout and vice-versa.

The geometric description contains, for each overlap, a geometric description of the area of the overlap and a link to the related ground-truth region (if any) and the related segmentation region (if any).

The next stage is to identify the errors which have occurred in the segmentation. The different types of errors detected in previous approaches fall broadly into the following categories:

- Merges
- Splits

- Missed regions
- Partially-missed regions
- Erroneously-detected regions

It should be noted that some approaches detect just these errors, while others divide them into sub-categories depending upon the focus of the particular evaluation method. The individual error types are described below and are related to the derived geometric description.

### **Merged regions**

Ground-truth regions are said to be merged in the segmentation when a single segmentation region overlaps (covers the same area of the page as) two or more regions in the ground-truth. The effect of this is that the contents of two distinct regions from the document would be merged in the segmentation being evaluated. In the geometric description, this may be identified by the presence of multiple overlaps between multiple ground-truth regions and a single segmentation region.

### **Split regions**

Splits are very similar to merges but with the region types reversed. That is, a ground-truth region is said to be split when it overlaps with more than one different segmentation region from the segmentation. The effect of this is that information from the document which belongs in the same region would, given the segmentation being evaluated, be divided into separate regions and no longer related as it should be. This may be identified by searching in the geometric description for multiple overlaps between a single ground-truth region and multiple segmentation regions.

### **Missed regions**

Missed regions are regions which are part of the ground-truth but do not have any overlapping regions in the segmentation. This is typically the most serious error which may be encountered in performance evaluation since it means that, if the segmentation is used, the contents of the given region will be completely omitted from further processing. In the geometric description, completely missed regions may be identified by finding ground-truth regions which have just one overlap and that overlap does not correspond to any segmentation region, but rather just the page background.

### **Partially-missed regions**

Similar to the missed regions described above, partially-missed regions are those ground-truth regions which are partially overlapped by some segmentation region but which also have areas which are not covered by any segmentation region. This error may not be as serious as fully missed regions since at least some of the information contained in the region has been detected as being part of a region. That part of the information will be passed on correctly to the recognition process. However, for the parts of the region which are missed, that information will be lost. Partially-missed regions may be identified by searching for ground-truth regions which are overlapped by more than one region, one of which corresponds to the background in the segmentation.

### **Erroneously-detected regions**

Erroneously detected regions are regions which occur in the segmentation but do not overlap any region in the ground-truth, or parts of segmentation regions which do not correspond to any part of a ground-truth region. Such errors may be detected using the processes described in the previous chapter by finding segmentation regions which overlap the page background in the ground-truth, either solely or as well as overlapping valid ground-truth regions. However, the system does not track such errors further as the inclusion of extra white space in regions rarely causes any significant problems to page segmentation.

## **6.2.2 Identification of region correspondences**

In order to identify all these types of errors, it is necessary to identify which regions overlap with other regions. In order to identify all the different types of errors, it is necessary to calculate these correspondences in both directions. In order to identify split, missed and partially-missed regions, it is necessary to find the segmentations overlapping a given ground-truth region. In order to identify a region merge, it is necessary to find the ground-truth regions covered by a given segmentation region.

This information is stored in two lookup tables, one for the ground-truth regions and one for the segmentation regions. For example, the ground-truth lookup table is a two-dimensional array with one row for each ground-truth region. The entries in each row correspond to the segmentation regions which overlap the given ground-truth region. Similarly, with the segmentation lookup table, the table contains a row for each segmentation region and the entries in each row correspond to the ground-truth regions which overlap that particular segmentation region.

The lookup tables can be populated very efficiently simply by making a single pass over the entire set of region overlaps. For each overlap, the corresponding ground-truth region and segmentation region is read. Then, one entry is made into each lookup table. In the ground-truth lookup table, the row corresponding to the current ground-truth region is found and then a pointer to the overlapped segmentation region is added into its list. Similarly, in the segmentation lookup table, the row corresponding to the current segmentation region is located and a pointer to the overlapped ground-truth region is added into the list.

Once this single pass over the region overlaps has been performed, the two lookup tables contain a complete description of the ground-truth regions overlapping each segmentation region and the segmentation regions overlapping each ground-truth region. An example ground-truth lookup table may be found in table 6.1. Please note that in the example table, for visual purposes, each region is identified simply by its region ID, a unique numeric identifier allocated to each region. In the system, each reference to a region is stored as a pointer to the region itself, allowing the region's meta-data to be accessed, thus allowing greater flexibility during later stages. Where a region in the ground-truth is partly missed in the segmentation, it is described in the table as overlapping the null region.

0	null	4		
1	null	4		
2	null	4		
3	null	6		
4	null	7	8	9
5	null	25	10	

Table 6.1: A ground-truth lookup table showing a row for each ground-truth region and a list of the overlapping segmentation regions for each.

Take, for example, the ground-truth lookup table depicted in table 6.1. This contains 6 rows (numbered 0–5), corresponding to six ground-truth regions. The entries in each row then depict the segmentation regions which overlap that ground-truth region. Row 0 contains the overlap information for ground-truth region 0. There are two segmentation regions listed as overlapping this region, segmentation region null and segmentation region 4. The null value corresponds to background in the segmentation layout, essentially meaning that part of this ground-truth region has been missed. Additionally, part of the region has been overlapped by segmentation region 4.

From these lookup tables, all of the errors listed above may be detected. The ground-truth regions which have been missed are those in the ground-truth lookup table which have only one region in the list, corresponding to the background. Similarly, ground-truth regions

which are partly missed are those which have more than one segmentation region listed, one of which corresponds to the background. Split regions are those which have more than one regular (non-background) segmentation region in their list. Similarly, Merged regions may be detected from the segmentation lookup table by looking for segmentation regions which have more than one regular ground-truth region listed.

### 6.2.3 Severity of errors

The initial description above has referred to only a simple set of errors. Indeed, many of the prior approaches have considered only these types of errors. However, it should be noted that different errors of the same type, e.g. misses, may not cause equally great problems for later stages of the recognition process.

In the PRImA Dataset upon which this performance evaluation method is based, areas of text are split into text regions on paragraph boundaries. Similarly, in some of the other datasets discussed in chapter 4, individual lines are described as text regions. However, segmenting the page along such lines may not be the aim of the individual layout analysis method. Furthermore, combining two consecutive parts of the same text column is unlikely to cause problems in subsequent stages of the recognition process.

Given a merge of two regions of different types, e.g. a text region and an image region, then this is much more likely to be contrary to what the developers of a layout analysis method intended, and what users are likely to expect from a correctly-functioning layout analysis method. Similarly, such mistakes are far more likely to cause problems in the later stages of the recognition process.

As mentioned previously, many prior performance evaluation systems simply output numbers of split and merged regions. Given the widely differing situations described in the previous two paragraphs, it is apparent that not all split and merged regions should be treated equally. Rather, the severity of those errors depends upon several factors. Given this, this performance evaluation method divides such errors into two distinct categories — severe and less severe — which allow the more problematic problems, such as merges between image and text regions, to be separated from the more technical errors which will cause few problems in real document analysis situations, such as merges of adjacent paragraphs of text in a single textual column. At this stage, severe and less severe are simply categories into which errors are divided. The quantitative effect on the results depends on the relative penalties allocated to each class in the given application scenario. This is discussed further in section 6.4.2

One of the chief advantages of using the PRImA Dataset is that it contains a large amount of region-level metadata which is invaluable when categorising errors as severe or less severe.

For merges, the type of each ground-truth region is checked to determine whether the regions are of the same or of different types. Where regions are of different types, this means that the recognisers to be used for each are different and merging regions of different types will have a detrimental effect on later stages of the recognition process. So, merges between regions of different types are immediately considered severe errors.

Where regions are of the same type, the categorisation is performed differently for each region type involved depending upon the reasons behind ground-truthing regions of the same type as separate regions. Image regions, for instance, always contain a whole single image. Where two different images are present on the page, then they are ground-truthed as separate image regions. Merging two separate regions is unlikely to be desirable so these are also categorised as severe errors.

For text regions, on the other hand, the situation is less clear-cut. As discussed earlier in this section, the merging of two consecutive text regions from the same column is unlikely to cause problems. However, finding a metric to measure this is more complicated. The PRImA XML format contains several pieces of metadata which may be useful in this case. Firstly, there is the polygon outline of the regions involved. The location of each region relative to the other may be used to find if, for instance, one region is directly beneath another. This may work in some cases but in others, it may work less well, for instance, when a page is divided horizontally into two articles. Columns from the second article may be vertically adjacent to columns from the first article but a merge between such regions would be considered a severe error.

The PRImA ground-truth format also contains a system for specifying the region order of the page by specifying links between each region and the region which follows it in the reading order. This could also be used in identifying severe or less severe errors. So, for instance, if a region follows another in the reading order, then that could be considered a less severe error. However, consider the case where adjacent columns in a text region are merged. The rightmost of the pair will typically follow the leftmost in the reading order. However, a merge between these two would be very undesirable since the later recognition stage may also merge the text lines between the two.

So, to take account of these problems, both of these features are considered when evaluating the severity of a merge or split. First, the text direction (left-to-right or top-to-bottom) and orientation of each of the regions is taken into account. If the text directions or orientations are different, the merge is considered a severe one. If they are the same, then the text direction is compared to the relative angle between the two regions. If the text is written left-to-right or right-to-left and the regions are horizontally adjacent, then this is a severe error.

If they are vertically adjacent, then the reading order is checked to see if they are adjacent in the reading order. Only if they are also adjacent in the reading order is the merge labelled less severe.

In this way, the errors previously detected as merges or splits are categorised into severe and less severe errors which allows them to be penalised accordingly later in the process.

### 6.3 Problems with the region-only approach

Up to this point, the comparison of the ground-truth and segmentation layout has been entirely based upon the region representations and no attempt has been made to access the original document image. From a performance perspective, this is a good thing. However, there is a problem with using this approach alone.

When creating ground-truths for documents using the standard bounding box method, there are rarely different ways of ground-truthing a given region. The correct bounding box will be the one which fits the region perfectly. The same may be said for the segmentation.

When a more detailed region representation is in use, such as polygons, then there are many more ways in which a given region may be ground-truthed. Of course, the goal in ground-truthing is to draw the region outline so that it includes the whole contents of the region inside, does not include any parts of other regions and includes as little as possible of the surrounding background space.

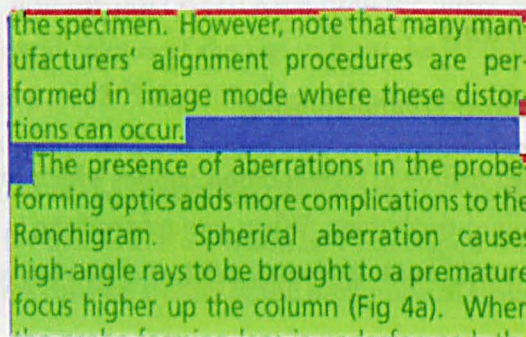


Figure 6.1: A portion of an image from the PRIMa Dataset with the ground-truth and a segmentation from the ICDAR 2005 Page Segmentation competition overlaid.

Take, for example, Figure 6.1. This shows an example document image which has the ground-truth and a segmentation from the ICDAR 2005 Page Segmentation competition overlaid on it. Portions which correspond to overlaps between the ground-truth and segmentation, the majority, are highlighted in green. Portions of the ground-truth which have been

missed by the segmentation are highlighted in red and portions of the segmentation which are not part of the ground-truth are highlighted in blue.

It can be seen that the evaluation method has detected some missed areas and some wrongly-detected areas. However, it can also be seen that all of the actual textual contents of the region are correctly detected since they appear in the areas highlighted in green. So, this highlights a problem. If all of the textual contents of the region have been correctly detected by the segmentation method, then should these errors be considered true errors?

Initially, it may seem that if the segmentation does not correspond correctly to the ground-truth, then the segmentation must contain errors. However, in situations such as that depicted in the figure, it can be seen that the segmentation conforms well to the rules discussed previously. It does contain the whole contents of the region while containing little of the surrounding page background and no portions of any neighbouring regions.

If the segmentation does not contain any error, then perhaps the ground-truth contains some error. However, the ground-truth also adheres well to those rules. It contains all of the contents of the respective regions while not containing any parts of other regions and containing little of the background region. So, the ground-truth cannot be said to be erroneous.

Instead, the problem is that in some instances, either the segmentation or ground-truth is fitting more closely to the data than the other. While neither can be said to be in error, the result is that the method using a pure region-based approach will penalise the segmentation for something which is not an error.

### 6.3.1 Possible solutions

#### Pixel content labelling during ground-truthing

This problem exhibits itself mainly in segmentation regions which fit to the region contents more closely than the ground-truth does. When ground-truthing is performed by humans, typically the region outline will closely surround each line of text but without fitting any more closely.

When a segmentation method is segmenting the page, however, the closeness of the fit depends largely upon the particular methods used. It is not uncommon for a segmentation method to produce outlines which wrap tightly around features of individual characters.

One possible solution to this problem would be to always make the ground-truth fit more closely to the region contents than any segmentation method could. However, there are several problems with this approach. One problem is it is difficult to decide how close a fit would be necessary to achieve this. Taken to the extreme, this would mean that the ground-truth



would have to include only pixels which are part of the actual contents of the region, in order to ensure that no segmentation method could fit more closely to the characters.

However, one problem with this approach is the cost involved in ground-truthing. The manual ground-truthing of documents is a resource-intensive process. If this required each pixel of a region to be specified rather than simply an outline, this would increase the cost of ground-truthing significantly.

### **Referring to the image during evaluation**

The problem discussed here is peculiar to region-based performance evaluation methods. In previous image-based performance evaluation methods, only black pixels are considered as region contents so only errors containing some black pixels are considered. Image-based methods were rejected because their assumption of black contents on a white background makes them unsuitable for dealing with modern colour documents and documents containing images. However, the idea of using the image to differentiate between the page background could be useful in situations such as this.

In the dataset used here, each region contains metadata, among which are the foreground and background colours of regions. So, when deciding which areas of the erroneous region are region contents, the pixels from the colour image can be compared against the known foreground and background colours of the region.

For the area of the error, the method begins by looping through each pixel in the area. For each pixel, the colour of the pixel is extracted from the image and compared against the foreground and background colours of the region from the ground-truth. The pixel is then allocated a score between 0 and 255 depending upon how close to the background (0) or foreground (255) the pixel's colour is. These values may be averaged over the whole area to obtain a metric of the proportion of the missed area which contains useful contents as opposed to background space. So, for instance, if a portion of a text region is detected as missed by the segmentation but the area consists wholly of the background colour, then no penalty will be assessed for the error.

One of the problems with image-based performance evaluation methods discussed in Chapter 3 was that the paradigm of dark foreground and light background does not apply well to pages containing images. Despite the use of colour here, this would cause a problem. Image regions may contain a variety of tones, all of which may be part of the useful contents of the image. So, the approach discussed in this subsection is applied only region types which fall into the paradigm of contents of one colour placed onto a background of a contrasting colour:

- Text
- Table
- Maths
- Separator
- Line Drawing

This alteration is an optional feature of the method but is enabled by default and recommended since it allows more accurate decisions to be made about partially missed regions.

It should be noted that the decision whether or not to refer to the image is based solely on the region type here. This is a good, but not perfect, indicator for the usefulness of referring to the image. Ideally, the ground-truth data would contain some metadata tag which would indicate whether or not each specific region is amendable to such analysis. Unfortunately, the dataset upon which this research is based does not yet contain such metadata.

## 6.4 Error quantification

The description so far has focussed on the correct identification of errors. Region merges, splits, misses and partial misses are detected. However, one of the principal goals of the method is, rather than providing a simple count of errors, to provide an evaluation which accurately reflects the problems which occur in the segmentation. Additionally, it is a goal to provide an evaluation which is tailored to the end-user's application area.

In quantifying the errors made, the method works from the viewpoint that the ground-truth is the hypothetical perfect segmentation of the page. If, for example, one were to compare the ground-truth against the ground-truth, then the match should be perfect with no errors made. When a regular segmentation is compared against the ground-truth, it will typically contain a number of errors and these errors should be quantified relative to their importance on the page.

In order to take account of the requirements of accuracy and flexibility, two complementary weighting schemes are proposed.

### 6.4.1 Area-weighting

One of the problems with methods which simply output counts of region merges or splits is that they do not take into account the severity of each error and the importance of the region(s) where the error occurs.

Take, for example, a split of a drop capital region and contrast it with a vertical split of a column of text. A method which simply outputs a count of errors will simply report that two splits have been detected. What is missing is some description of the relative importance of each, and an idea that one of the errors is much more costly than the other.

It is necessary, therefore, to allocate to each region some measure of the importance of that region. The authors of the Pink Panther method[29] suggested three different weighting schemes: weighting simply by the count of regions, weighting by the region heights and weighting by the area of the region. The first is unsuitable since it treats all regions as equal, regardless of their relative importance. The second is an improvement but the third provides the most true measure of the importance of a region on the page.

When document layouts are first designed, careful attention is often given to arranging the regions on the page in order to convey their relative importance. The chief method for doing this is by adjusting the relative sizes of different pieces of information on the page. Take, for example, an article from a newspaper. Typically, such articles are laid out in a hierarchical fashion. The most important piece of information, the quick summary of the story designed to attract attention, is called the headline and is printed at the top in a very large font compared to the rest of the page. Although it may be short in terms of the number of words, the importance is indicated by the size of the text. The final paragraphs of the article typically contain the less important details, perhaps more specific, and are usually placed towards the end and in a much smaller font size than the title. In between will be a range of different segments in different sizes. In order to measure accurately the importance of a given region, its area is used to weight any errors.

The weighting by area allows relatively small regions to be treated as less important than larger regions, *ceteris paribus*.

### 6.4.2 Application scenarios

Document Image Analysis methods, and Layout Analysis methods in particular, may be used in a wide variety of different application scenarios. Different users have different tasks on which methods may be used. As such, the particular Layout Analysis method which is best-suited to a particular application scenario may be different than that which is best suited for a different application. Given this, it is important for any performance evaluation system which will be used in evaluating Layout Analysis methods to evaluate based on the needs of the end-user.

The method has proceeded up to this point without regard to the application scenarios. All detected errors have been weighted according to the area of the regions involved. The

use of area-weighting allows errors to be weighted according to the effect on the recognition process. However, it does not take into account any measure of the importance to the user's application.

In particular application scenarios, it is typically the case that some types of region may be more important to the user than others. In an indexing application, body text will usually be ignored but regions containing metadata such as article titles and by-lines will be extremely important.

In order to take this into consideration, the method allows application scenarios to be defined by specifying weighting multipliers for each region type or sub-type. These multipliers are used as multipliers which may be applied to the areas of regions and the corresponding error deductions. This has the effect of increasing or decreasing the proportion of the document's final score which is allocated to the given region.

By default, all of these weighting multiples are set to 1.0. This means that each region is accounted for in the final evaluation in proportion to the region area. However, these multipliers may be adjusted by the user. For example, if a given type, or sub-type, of region is completely unimportant to a user, the multiplier for that region type may be set to 0.0. Thus, the region itself and any errors belonging to it will be allocated no weight in the final evaluation, causing other errors to appear relatively more important.

If, for example, a particular region type were considered more important than other regions, then it could be allocated a higher weighting. If the multiplier for a given region type were set to 2.0, then the region itself and any errors encountered in it would be weighted twice as highly as other region types, all other things remaining equal.

This allows the method to be adapted to different application scenarios by allowing the user to alter the relative weightings of the region types which are more important and less important in the given scenario.

Additionally, the application scenario may specify the penalties to be used for assessing the importance of split and merges. As described earlier, errors are categorised into severe and less severe and different penalties may be applied to each category. This is defined in the application scenario since the relative importance of errors may vary between application scenarios. In the two application scenarios described here, however, penalties of 40% are applied for severe merges & splits and 10% for less severe merges & splits. The ICDAR Page Segmentation Competitions applied overall penalties of 25% for all merges and splits since the capability did not exist to divide these errors into categories based on severity. The values of 40% and 10% were selected, therefore, to highlight the ability of the new method to separate errors of different types, to apply small penalties for trivial errors and larger penalties

for significant errors and so that the average penalty would be aligned with that used for the competitions to aid in comparing results.

### 6.4.3 Pre-defined application scenarios

The method allows weightings to be specified by the user. However, one of the goals of a performance evaluation method discussed in the introduction was that it should enable the published results of different layout analysis methods to be compared readily. Giving the ability for all users to adjust the parameters of the performance evaluation method could potentially work against this goal. Results which have been generated by the same performance evaluation method but using a different set of weights would not be comparable.

One solution to this problem is to define a small number of application scenarios which would allow developers in given areas to select the appropriate weighting scheme for their application area. This would allow evaluations to be tailored to a small number of scenarios, giving a more useful analysis, while at the same time ensuring that developers working in the same application area would be able to publish results which are comparable.

The sections below describe a number of such application scenarios. These have been implemented in the system. It should be noted that the aim here is to provide some useful scenarios and to highlight the flexibility of the system. It is not intended to be a comprehensive list of all applications in which the performance evaluation system may be used. See section 8.3.

#### General Document Recognition

In a general document recognition application, the goal is typically to obtain as full and correct a digital representation of the original document as is possible. No particular types of regions will be favoured over any others — the desire is to correctly detect whichever regions are present in the original document. Given this, the General Document Recognition application scenario does not weight any regions more heavily than others. All regions are weighted solely according to their frequency on the page and their area relative to others on the page.

#### Document Indexing application

In document indexing, the focus leans more towards identifying and locating the few regions which serve to summarise and locate the interesting regions. For instance, when indexing articles from a magazine or a technical journal, the key pieces of information which are required are article headings, authors and page numbers. Other regions such as regions of body text or

image regions are part of the contents themselves and so are uninteresting from an indexing perspective.

In order to implement an application scenario for Document Indexing, the relative weights for all region types are set to 0, except the following region sub-types, which are all allocated a relative weighting of 1.0:

- Headings
- Sub-headings
- Credits (*chiefly used for by-lines*)
- Page numbers

This particular scenario causes the evaluation method to disregard all regions which are not important from an indexing perspective, while weighting the remainder according to their area.

## 6.5 Presentation of results

Another of the key goals for the evaluation method was to provide a greater degree of descriptiveness than other methods. Prior methods have largely focussed on error quantification or benchmarking. This typically generates one single statistic or a small number of statistics which are used to represent the overall performance of the system being evaluated.

Such methods are undeniably useful for a number of applications. Where a user is aiming to select a Layout Analysis method which works well in a given application, then perhaps a single overall statistic will be more appropriate and will allow a simple comparison to be made against the results from other approaches.

Similarly, if a given Layout Analysis method contains a number of user-specifiable parameters and a users wants to find which combination of parameters is most likely to achieve the optimal result for the given application area, a single overall statistic will allow the user to select parameter values which maximise the overall metric.

In other situations, it will be more desirable to have more in-depth information available about the algorithm. This will be particularly true for those who are developing Layout Analysis methods. For a developer, a single statistic may be useful. For example, a single metric would allow a developer to ascertain whether or not a given change to the method had a positive or detrimental effect on the method overall.

However, when a developer is seeking to improve the capability of the system, the developer will desire to know the exact errors which are present in the detected segmentations. For example, knowing whether or not the method makes more splits or merges, the developer may be able to tune parameters which cause these problems to occur.

### 6.5.1 Implementation

To this point, the method has converted the geometric ground-truth and segmentation regions into a geometric description of the overlaps between the two. It has then derived from this a description of all of the errors present in the segmentation. These are described in the system with the following specific information:

- Ground-truth document in which the error occurs
- List of regions involved in the error
- Type of error — merge, split, partial miss or miss
- Significance of error — severe or non-severe
- Area involved in the error
- Deductions to be made for the error
- Weighting for the error

So, in essence, this is a comprehensive listing of all the errors made in the given segmentation and the regions to which they apply. While descriptive, the information has been greatly reduced from the original geometric descriptions. However, each region in the error description is described by a pointer to the original region. So, from this error description, all of the metadata of the regions involved in each error is still accessible.

This description may be sorted or excerpted to provide a variety of different outputs. The following describes the different types of output which are made and the methods used to obtain them.

#### Error type overview

From the perspective of a developer of Layout Analysis methods, one of the most important items to know is the frequency of different types of errors which are present in the segmentations produced by the method. This information may be desired at two different levels. At a

higher level, it would be desirable to know which basic error types — merges, splits — cause the greatest problems in the segmentation. At a lower level, it is desirable to know not only which error types but also which types of region the errors occur most often in.

In order to obtain the higher-level view, the above error description representation may simply be sorted by the error type. Once this is done, errors of the same type may be grouped together and the overall error deductions for each error of the given type may be summed to give a summary of the different types of errors, along with the effect of each upon the overall results.

The lower-level view, which is a categorisation of errors by error type and the region types involved. When merges are made, the regions involved may be of the same type or merges of regions of one or more different types are involved. These are categorised into single region types (i.e. text, image, etc.) or just merges involving multiple regions. The error list shown above is sorted using the standard C++ `qsort` function but with a custom comparison function which takes into account both region types and error type when returning. This sorts the array into the different categories. The error costs of each is then summed and output.

## 6.5.2 Detailed error description

The above functions allow statistics to be produced on categories of errors. This is useful from the point of view of developers but it would also be useful for developers to see the individual errors which contributed most to the poor performance of the method. This allows the circumstances involved in specific errors to be identified which could then identify potential improvements in the method. Listing these individual errors by their impact on the method's final score allows development effort to be focussed initially where it will have the greatest impact on the performance of the method.

This output is obtained by sorting the error list by the total cost of the error. Rather than then categorising as in the previous outputs, the errors are listed individually, along with details of the regions involved. There is the option to output a graphical representation of each individual error generated from the geometric description of the interval comparison, in order to allow these errors to be visualised. These are output as Scalable Vector Graphics, currently to individual per-error files. However, as part of future work, it is intended to develop a web interface which may expose this functionality. For example, if such a list of errors is displayed by the web interface, then clicking on the individual errors should display the associated diagram.



### 6.5.3 Detailed region description

The previous two sections have described ways in which the method outputs summaries and precise descriptions of errors are output by the system. As well as this, it is possible to output errors on a per-dataset, per-document and per-region. This listing contains a comprehensive list of document pages, regions and a listing of errors for each region, while giving recognition rate statistics at the region, page and dataset level.

This allows developers to obtain information at a variety of levels. By viewing at the dataset level, developers can obtain a summary of the method's performance for the whole dataset. This then may be broken down to the page level which allows developers to obtain a performance metric for each document page. This allows the developer to identify particular document pages on which the method performs particularly well and particularly poorly, again providing a means for focussing development effort on particularly problematic documents. Again, the document-level statistics may be broken down into individual regions, highlighting the individual regions which have the most errors.

This representation is obtained by iterating through the documents, then pages and then regions present in the dataset evaluated. For each individual region, the overall area (adjusted by the weighting multiplier) is obtained then a listing of individual errors in that region is listed. Statistics on the overall recognition rate for each region are output, given as percentages for each error and overall for the region. These are then summed and output both at the page level and at the dataset level.

## 6.6 Sample output

This section contains sample output from the evaluation system. In order to demonstrate the output of the system, an example document will be used. This example document is illustrated in Figure 6.2.

In the top-right of the figure, the correct ground-truth segmentation for the document is illustrated. To the bottom-left is an artificial segmentation of the image which is based on the ground-truth but which has several segmentation errors introduced. Firstly, the upper paragraphs of the first two columns (GT regions 1 & 3) have been merged horizontally, something which should be categorised as a serious merge. In addition, a small portion of GT region 3 has been missed. The next error is in the rightmost column where a paragraph of text has been divided into two horizontally, a serious merge. Also at the site of the split, a small portion of the region has been missed.

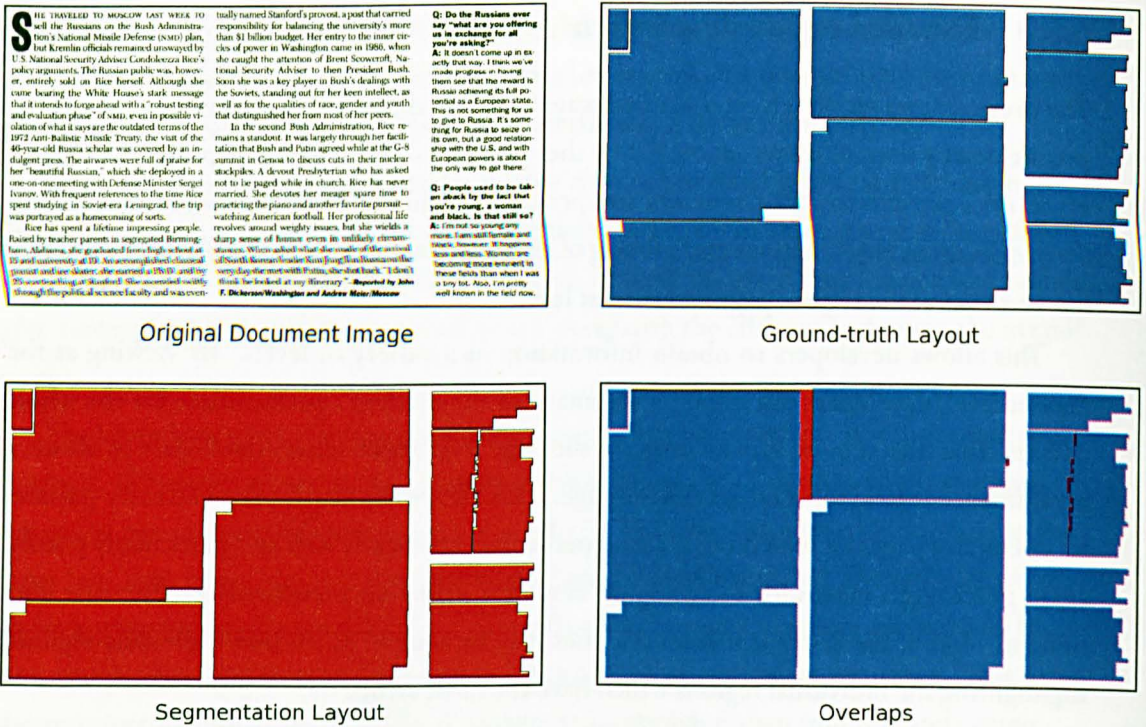


Figure 6.2: An artificial example document image, ground-truth layout, segmentation layout and the overlaps between them.

Figure 6.3 contains the listing of errors detected by the system when run on this example document. The first section of the output shows the "Overall Score" and this section is itemised by ground-truth region. For each GT region (0-8), the full area for each region is shown followed by a listing of errors found in that region and penalties assessed for each error. A score is given for how well each individual region is segmented. At the end of the listing, a listing of errors divided into region types is given followed by an overall score for the whole page.

In this example, it can be seen that regions 0, 2, 4, 5, 7 and 8 each have no errors listed and are given as they have been correctly segmented. As the top paragraphs of the first two columns (regions 1 & 3) have been merged, a merge error is listed under each of these regions. For each of these merge errors, a penalty of 40% of the region area has been applied which is the penalty specified in the main application scenario for a serious merge (see Section 6.4.3). In addition a portion of GT region 3 is listed as being missed with a penalty corresponding to the small portion of the region which was missed.

In the example segmentation, a paragraph in the rightmost column (GT region 6) was split into two. In the listing, it can be seen under the section for GT Region 6 that two split errors are listed, one for each Segmentation region into which the region has been split. Again,

the penalties listed for the splits add up to 40% of the region area involved. In addition, a small portion of the region is listed as missed, corresponding to the portions which fall outside both of the corresponding segmentation regions.

Towards the end of the listing can be seen a listing of the scores for each region type individually (note that this example document contains only text regions) and finally an overall score for the whole page, in this case 78.5% which is due to the page being generally well segmented but with some serious errors in important regions.

## 6.7 Discussion

This chapter has built upon the layout comparison method described in the previous chapter in order to provide a fully-fledged performance evaluation system which aims to meet the criteria discussed in the introduction.

The system uses a lookup table in order to identify parts of the segmentation which merge ground-truth regions, parts which split ground-truth regions and parts which correctly recognise ground-truth regions. Once these correspondences have been determined, the system allocates a weighted score to each error based upon the area of the region(s) involved, the user-specified penalties for the error and the relative weighting specified for the given region type in the application scenario. This gives the effect that errors are weighted according to their importance in the current document page and to the user's preferences.

Based on these quantified errors, a number of presentation schemes have been implemented which aim to display the detailed results of the system in a way which most benefits developers.

## 6.8 Summary

This chapter has presented a detailed description of a performance evaluation method based on the region comparison method described in the previous chapter. This method, being based on a polygon region comparison, referring to the image only where this is desirable, is very accurate while retaining efficiency. The error quantification system allows errors to be ranked with regard to the importance of the regions on the page and to the relative importance specified by the user. The results from the method are presented in several ways which are tailored for providing useful information to developers. The following chapter analyses the system against the goals set for it.

## Overall Score

-----

## Page 0:

## GT Region 0:

Full Area	12717	100.0%
-----------	-------	--------

-----

Score	12717	100.0%
-------	-------	--------

## GT Region 1:

Full Area	601654	100.0%
-----------	--------	--------

-----

Merged	240661	40.0%
--------	--------	-------

-----

Score	360993	60.0%
-------	--------	-------

## GT Region 2:

Full Area	225161	100.0%
-----------	--------	--------

-----

Score	225161	100.0%
-------	--------	--------

## GT Region 3:

Full Area	325949	100.0%
-----------	--------	--------

-----

Merged	130228	40.0%
--------	--------	-------

Missed	379	0.1%
--------	-----	------

-----

Score	195342	59.9%
-------	--------	-------

## GT Region 4:

Full Area	521804	100.0%
-----------	--------	--------

-----

Score	521804	100.0%
-------	--------	--------

GT Region 5:		
Full Area	47526	100.0%
-----		
Score	47526	100.0%
GT Region 6:		
Full Area	184386	100.0%
-----		
Split	32431	17.6%
Missed	4476	2.4%
Split	39532	21.4%
-----		
Score	107947	58.5%
GT Region 7:		
Full Area	52376	100.0%
-----		
Score	52376	100.0%
GT Region 8:		
Full Area	106265	100.0%
-----		
Score	106265	100.0%
Text =	1630131	/ 2077838 = 78.5%
Graphic =	0 / 0	= 0.0%
Line Art =	0 / 0	= 0.0%
Separator =	0 / 0	= 0.0%
Noise =	0 / 0	= 0.0%
-----		
Total =	1630131	/ 2077838 = 78.5%

Figure 6.3: A sample of the region description output of the system depicting a portion of the page containing a column of text which has been merged, detected by the system as a series of allowable merges.



# Chapter 7

## Evaluation

### 7.1 Overview

Chapter 6 contained a description of a performance evaluation method based on the region comparison method described in the chapter prior to that. This chapter focuses on evaluating this performance evaluation method. The chapter begins with a recap of the goals set out for the system in the introduction. Following from this, each of these goals are discussed in detail and compared with how well the system meets each of these goals. Examples are given from the system which highlight how each goal is met.

### 7.2 Introduction

Many previous performance evaluation methods have been published in the literature and the scores allocated to a particular segmentation by each of the performance evaluation methods is likely to be different. Given these differences, one question which arises is what causes these differences. If different performance evaluation methods give different results on the same data, then perhaps one of the methods contains some bug which causes it to operate differently to the developer's intentions. Or, if the results are as the developer intended, whether the different results offer a more useful or more accurate performance evaluation.

In the introduction, the following goals were set out for the system:

- Accuracy & Applicability
- Flexibility
- Descriptiveness

- Efficiency

The following sections discuss each of these goals individually, and how well the system meets the particular goal. The system is tested with synthetic and real-world data as is appropriate and results are given which illustrate the ability of the system to meet that goal.

### 7.3 Accuracy & Applicability

The first two goals of the system, accuracy and applicability stem from similar features of the system. The goal of accuracy is that the system be able to make evaluations which make full use of the input data. The goal of applicability means that the system must be based on a region representation which is capable of representing accurately modern, complex documents and that the method used by the system can evaluate such representations accurately.

As input, the system accepts both ground-truth and segmentation represented as arbitrary polygons. The most modern dataset for layout analysis, the PRImA Layout Analysis dataset, contains a large number of modern, complex documents. The region representation which is used for this dataset is the isothetic polygon, which is a special case of the arbitrary polygon. The system is designed to import such documents with complete accuracy. Similarly, of the modern layout analysis methods described in chapter 2, the most complex output polygon representations. The polygon representation was used as the basis for the ICDAR Page Segmentation Competitions. So, the system is capable of operating on data from the most modern layout analysis dataset and is capable of operating on the most complex layout analysis methods.

Given that the system is designed to import such data, it is necessary to show that the system can operate accurately on such data. The representation method upon which the system is based is the region interval. In terms of expressive capability, region intervals, in a discrete domain, are capable of representing all documents which may be represented by arbitrary polygons. The conversion between the initial polygon inputs and the region interval representation is performed in a lossless manner.

The following subsections contain a number of tests which are designed to test the accuracy of the system. Initially, a number of tests on small, synthetic data, which are designed to test the pixel-accuracy of the system, are described. Following this, manual verification is made that the geometric overlap representation computed from the ground-truth and segmentation inputs represents them both accurately.



### 7.3.1 Testing using artificial data

In order to answer these questions over any new performance evaluation method, the system will first be validated using artificial data. The reason for this is that the system is designed to operate normally on document layouts corresponding to images of around 2000 pixels wide and 3000 pixels tall. Given the size of the input data, the human effort in verifying operation directly even on small amounts of such data would be prohibitive. Additionally, using real-world data provides no assurance that all the possible boundary cases of the system will be tested correctly. Some may be tested more than once, redundantly, while others may not be tested at all.

So, in order to test the various different aspects of the system which could potentially highlight problems, a set of test cases has been designed which aims to test as many different aspects of the working of the system. Using small, well-defined test cases allows not only the error detection to be tested but also the scores allocated to them.

### 7.3.2 Testing pixel-accuracy and simple errors

#### Testing accuracy and detection of partial or complete misses

For the first phase of testing, it was desired to test the accuracy of the system and the detection of simple errors. The system was designed to operate with pixel-accuracy so the output of the system should be as expected. In order to test this, the first simple test involves a small 3 pixel square ground-truth and segmentation region. The segmentation is moved by 1 pixel across the ground-truth region and the output from the system is measured against the expected result. These simple tests served to highlight one logic error in the system which was corrected in order to provide the correct output. Table 7.1 contains a listing of these test cases along with a diagram showing the relative position of the ground-truth and segmentation region in each case, the expected errors detected by the system and the correctness of the actual output.

A similar test with a further nine test-cases was performed in the vertical direction in order to check the correct operation of the system on this boundary. The results are identical to those described in Table 7.1 so they are omitted for brevity.

#### Detection of other types of errors

These initial 18 test cases have verified that the system correctly and accurately detects simple errors on simple input data. To this point, only the detection of misses and partial misses has been verified.










No.	Diagram	Expected result	Correct output
01		100% Missed	Yes
02		100% Missed	Yes
03		67% Missed	Yes
04		33% Missed	Yes
05		0% Missed	Yes
06		33% Missed	Yes
07		67% Missed	Yes
08		100% Missed	Yes
09		100% Missed	Yes

Table 7.1: Nine test cases intended to test the accuracy and simple matching of regions. For each test case is a diagram with the ground-truth region in blue and the segmentation region in red.

Other types of errors which the system is designed to recognise are merges and splits. The artificial test cases in Table 7.2 contain simple tests for the presence of splits and merges. For splits, the ground-truth contains a single region and the segmentation contains two regions which divide the ground-truth vertically into halves. For detecting merges, the same data are used but the ground-truth and segmentation are swapped.



No.	Diagram	Expected result	Correct output
19		100% Split	Yes
20		100% Merged	Yes

Table 7.2: A test case intended to test the correct detection of the different types of errors. For each test case is a diagram with the ground-truth region in blue and the segmentation region in red.

### Non-rectangular regions

The test cases presented so far have verified that the method is capable of evaluating accurately small, simple rectangular regions. It is necessary to confirm that the method operates on more complex polygonal regions as it was designed to. In order to do this, the method was tested again using artificial test cases so that the expected results could be calculated manually for comparison against the observed results of the method.

These test cases replicate the initial set of tests but use parallelograms rather than squares but of similar sizes to the squares used in the previous example, such that the results should be *identical*. *Each parallelogram is three pixels high and consists of a row of three pixels, with the second and third rows offset to the right by one and two pixels, respectively*. *For each test, the ground-truth and segmentation parallelogram are moved relative to each other by one pixel as in the previous test*. Given the construction of the parallelograms, the results should be the same as those from the previous test, but in this case the additional complexity will serve to verify the pixel-accuracy of the method in the vertical direction.

Diagrams of each test case, along with the expected result and how this matches the observed result, may be found in Table 7.3.

No.	Diagram	Expected result	Correct output
21		100% Missed	Yes
22		100% Missed	Yes
23		67% Missed	Yes
24		33% Missed	Yes
25		0% Missed	Yes
26		33% Missed	Yes
27		67% Missed	Yes
28		100% Missed	Yes
29		100% Missed	Yes

Table 7.3: Nine test cases intended to test the matching on simple non-rectangular regions. For each test case, a diagram is shown with the ground-truth region in blue and the segmentation region in red.

### 7.3.3 Manual verification using real-world data

The tests so far have used small, artificial tests which have been designed specifically to test individual features of the system. In measuring the accuracy of the system, as well as running test cases like those above, it is necessary to test the system with real-world data. Real-world data is significantly larger and more complex than what has been presented so far. This section presents a selection of real-world documents which are presented as the polygon representations from the dataset or layout analysis method from which they came, as well as the geometric description generated by the performance evaluation method.

Due to the size and complexity of the data, it is not possible to calculate the exact areas of the data in order to verify that the detected errors are correct. However, it is possible to

compare the two different descriptions visually to confirm that the derived geometric representation is the exact equivalent to the ground-truth and segmentation regions involved. See figures 7.1 and 7.2 for a sample ground-truth, segmentation and geometric representation.

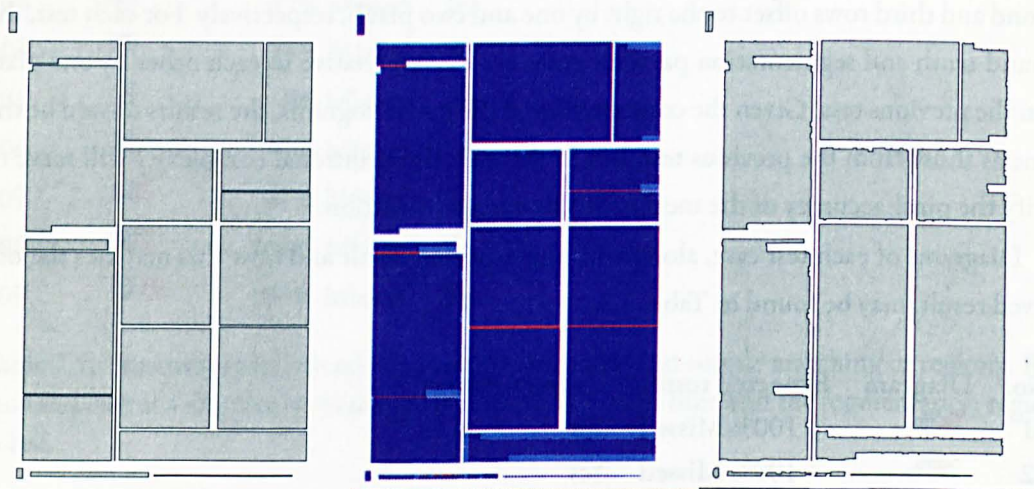


Figure 7.1: a) An example ground-truth, b) the geometric description derived from it and c) a sample segmentation of the image.

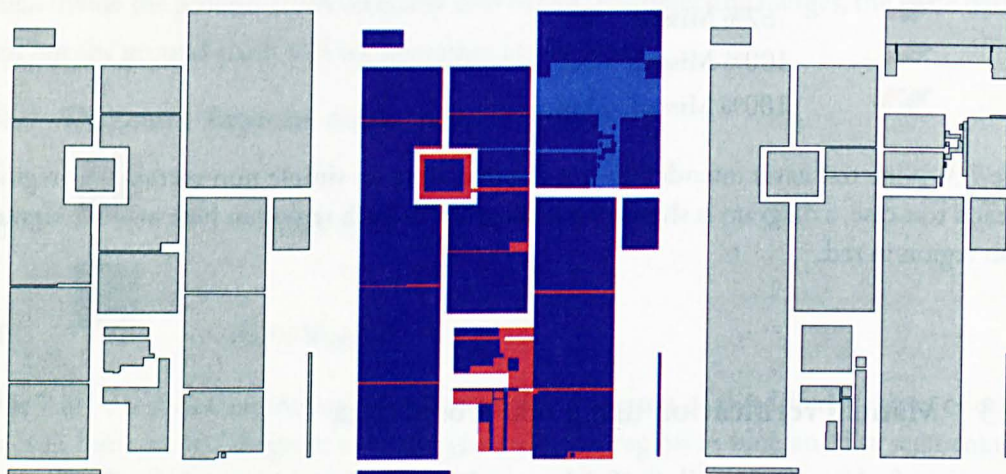


Figure 7.2: From the ICDAR 2007 Competition dataset, a) an example ground-truth, b) the geometric description derived from it and c) a sample segmentation of the image.

From the diagram displayed here, it is possible to see that the geometric description of the ground-truth and segmentation constructed by the evaluation method is an extremely accurate representation of the overlaps between the two. Indeed, this holds for the other documents on which the system has been tested, including all of the entries for the ICDAR 2005 and 2007 Page Segmentation Competitions.

### 7.3.4 Evaluation against competition results

Previous evaluations have been either on small, artificial test cases or necessarily restricted datasets. These have shown that the algorithm works correctly to single pixel accuracy on small non-rectangular features. A manual evaluation of outputs using a small selection of real-world data has shown that the method can correctly identify errors in real-world data.

Given that the errors detected have been detected correctly, it is necessary to check that the overall evaluation scores calculated by the system are representative of the true performance of the layout analysis methods it is being used to analyse. In order to do this, the system will be compared against another performance evaluation method.

The performance evaluation method which has been used most widely recently in benchmarking Layout Analysis has been the one used for the ICDAR Page Segmentation competitions which itself was based on the Pink Panther method.

In order to perform this comparison, the results from the ICDAR 2005 Page Segmentation Competition have been selected to maximise the amount of data to be used. For the 2005 competition, a total of four methods were submitted, while only three entered the 2007 competition.

For the competition, 30 images with accompanying ground-truths were selected for the test set. Here, the new method has been run on all of the data for each of the four entrants. For the evaluation, penalties for errors in the new system have been set to be the same as those used for the competitions to maximise the comparability of the results.

It should be noted that since the metrics used for the two systems are different, the magnitude of the results are not directly comparable. However, the relative results should be broadly comparable, bearing in mind the significant changes introduced in the new system.

The results of the ICDAR 2005 Page Segmentation Competition and the results of the new method on the same data may be found in Figure 7.3.

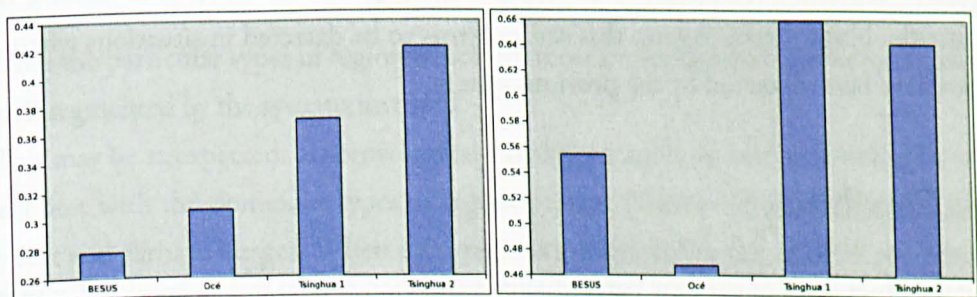


Figure 7.3: a) The published results of the ICDAR 2005 Page Segmentation Competition, and b) results on the same data with the new system.

### Explanation of differences & similarities

In the results from the new method, there are some significant differences. As before, the results are split roughly into two classes. The two entrants from Tsinghua University gain excellent scores while the other two entrants, from Océ and BESUS, achieve somewhat lower results.

However, within these two classes, there are significant differences in results, meaning that the order within the classes changed. The second Tsinghua method which previously held a 5% lead over the first Tsinghua method now leads by 1.8%. The Océ method which originally led by 3% now trails by 8.7%.

Given the new results, what are the chief differences between the old performance evaluation metric and the new which cause the results to be so significantly different?

One of the chief differences between the old method and the new is the weighting system used. The previous method works primarily on a region basis. Single regions are categorised as being missed, accurately recognised or merged or split and then the segmentation metric describes basically a percentage of regions which are correctly recognised. With the new system, all errors are weighted by the region area, meaning that errors in larger regions are considered more important than those in smaller regions. With the old system, all were treated equally.

The new system also improves on the accuracy of the matching. The old system used for the competitions incorporates some thresholds so that errors smaller than a given percentage of the region size are ignored, something which is necessary due to the reliance on region-level matching. However, using thresholds in this way effectively allows errors to be made without penalty, potentially causing loss of data without this being reflected in the segmentation metric. The new system describes errors in the magnitude in which they occur. If 5% of a region is missed, then an error of this size will be recorded.

One of the most significant differences in the new system compared to the old is that evaluations are detected, measured and weighted using the full contents of regions, rather than just the black pixels. Again, this causes errors to be detected in situations where they may not have been detected by the previous system.

## 7.4 Flexibility

One of the chief goals of the new performance evaluation system was to provide flexibility in the system. This recognises that layout analysis methods may be used for a number of different applications. The particular features which are most important in different applications are different, meaning that the layout analysis method most suited for a given application may be

different to that most suited for another application.

In order to accommodate the needs of different end-users, the system incorporates flexibility which allows different region types or sub-types to be weighted differently in the evaluation. In the previous chapter, several pre-defined application scenarios are defined which consist of a set of weights which are designed to weight the results towards the types of regions which are most important in the given scenario.

The effect of these scenarios may be observed by evaluating the results of the ICDAR 2005 Page Segmentation Competition twice using the evaluation system separately using the different scenarios. See figure 7.4 for a comparison of the results using the general layout analysis scenario and the document indexing scenario.

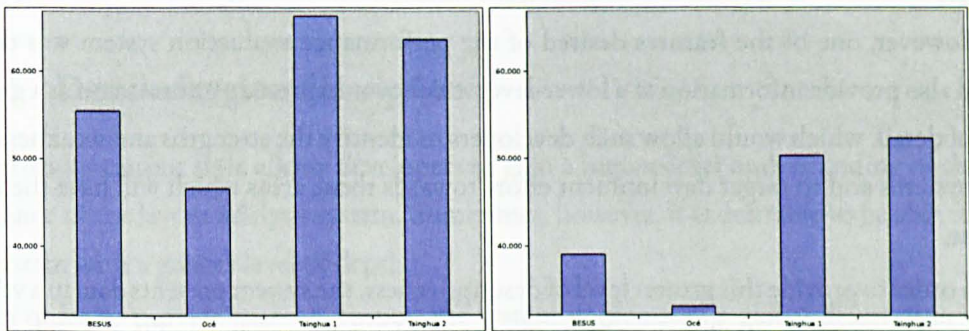


Figure 7.4: The results from ICDAR 2005 evaluated using a) the general layout analysis scenario, and b) the indexing scenario.

The general layout analysis scenario weights each region type equally, allowing them to be weighted solely by their area on the page. The indexing scenario excludes (allocates zero weight to) all region types which are not useful in a document indexing scenario. Only those which are useful, headings, by-lines and page numbers are retained.

The first observation which stems from the two graphs is that the overall recognition results are significantly lower for the methods using the document indexing scenario. This implies that the particular types of region which are most important in document indexing are less well segmented by the systems involved.

This may be as expected. It seems logical that layout analysis methods would be trained to deal best with the dominant types of region present in everyday documents, regions of body text and perhaps images. When these regions are ignored in the analysis, the remaining regions are less well recognised.

The next observation from the graphs is that the best-performing method for the indexing scenario, Tsinghua 2, was different from that in the general layout analysis scenario, Tsinghua 1.

The differences between the performance of the four methods between the two different scenarios, and the different optimal methods for each scenario, serve to highlight the need for goal-oriented evaluations as implemented in the system presented here.

## 7.5 Descriptiveness

As has been demonstrated earlier in this section, the method is capable of outputting single statistics which summarise the performance of a layout analysis method on a given document or even on a dataset. This is ideal when the system is being used for benchmarking or for tuning parameters of systems or comparing the results of different systems.

However, one of the features desired of the performance evaluation system was that it would also provide information at a lower-level which would provide information at a greater level of detail, which would allow such developers to identify the strengths and weaknesses of their systems and to target development effort towards those areas which will have the most benefit.

In order to provide this greater level of descriptiveness, the system presents data in a variety of ways. This section will illustrate these ways and evaluate how well they fill the goals.

### 7.5.1 Error type overview

From the viewpoint of a developer, it is desirable to have a broad overview of the problems with the system. Given this, one of the output methods implemented in the system aims to give developers a breakdown of all the errors detected by the system by the type of the errors involved. This allows developers to have a broad idea of where poor performance is coming from and how it can be rectified.

In order to view these error statistics, the system was run over all the documents and all the submitted segmentations from the ICDAR 2005 Page Segmentation Competition. The results are depicted in Figure 7.5.

The graphs show the percentage of the errors produced by each method which fall into the specific categories. From this side-by-side comparison, it is possible to see the strengths and weaknesses of the particular methods. For instance, both of the Tsinghua methods, which were ranked first and second in the competition, have most of their errors in wrongly-detected regions. The BESUS and Océ methods both have significantly larger proportions of errors in misses while having much fewer wrongly-detected regions.



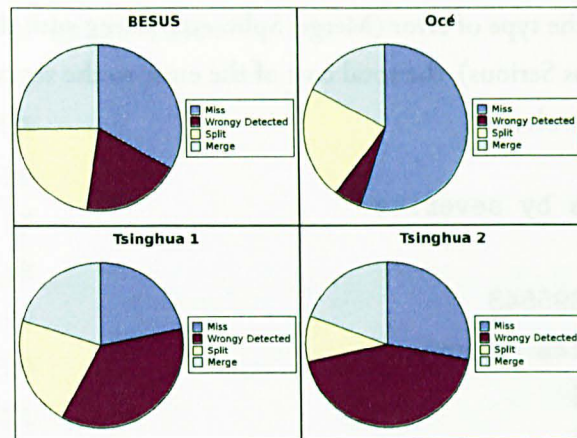


Figure 7.5: The categories of errors which contributed most to each method's score.

## 7.5.2 Detailed region description

The previous output style allows developers to gain a higher-level understanding of the performance of the layout analysis system. Sometimes, however, it is desirable to be able to view the system with a greater level of depth.

In order to provide such an output, the system implements a global description which contains a comprehensive description of the whole output of the system in a hierarchy. At the top level is the dataset. An overall recognition metric is given for the performance of the system at the dataset level. This is then divided to the page level which gives a recognition metric for each individual page evaluated, allowing the developer to see if the system performs particularly well or particularly badly on any specific page. From the page level, the output is further sub-divided into individual regions which have their own recognition metrics and for each individual region, a listing is given of all the errors which involve that region.

This multi-level description allows developers to pinpoint individual errors, the regions in which they occur, which documents in the dataset are most problematic for the system and which individual regions have the most problems. An listing of this style of output is given in 6.3

## 7.5.3 Detailed error description

As well as listings of errors by category, the system can output listings of errors with details of the error, details of the regions involved and the listing is sorted by the severity of the error. This allows developers to observe the most serious specific errors which have been caused by their system and allows individual problems to be identified and fixed.

The listing shows the type of error (Merge, Split, etc.) along with the severity where appropriate (Serious, Less Serious), the total cost of the error to the segmentation and gives a listing of the regions involved.

Listing of errors by severity:

Serious Merged: 395543

Seg Region 14 has merged:

GT Region 15

GT Region 13

Serious Merged: 392009

Seg Region 13 has merged:

GT Region 14

GT Region 12

Missed: 315146

GT region 17

Serious Split: 90825

GT Region 11 is split into:

Seg Region 1

Seg Region 11

Seg Region 12

Serious Merged: 77508

Seg Region 0 has merged:

GT Region 20

GT Region 19

Serious Merged: 67838

Seg Region 5 has merged:

GT Region 17

GT Region 6

Serious Split: 67838

GT Region 17 is split into:

Seg Region 5

Seg Region 23

Seg Region 16

Seg Region 17

Seg Region 18

Seg Region 19

Seg Region 20

Seg Region 21

...

## 7.6 Efficiency

One of the goals of the system was to ensure that the system was efficient in that it could compare ground-truths and segmentations derived from complex document in relatively small amounts of time. This may seem contrary to the goal of having accurate evaluations.

However, the region representation selected for the algorithm, region intervals, was deliberately selected because of its dual properties of accuracy and efficiency. As has already been mentioned, the accuracy of region intervals is equivalent to the polygon representations of the inputs. However, the deconstruction of complex regions into smaller rectangular intervals means that the comparison of otherwise complex regions becomes similar in complexity to bounding box comparisons.

In order to measure the efficiency of the system, it is necessary to test how long the system takes to run over a large set of data. For this, the results from the ICDAR 2005 Competition were used. The competition test dataset had a total of 26 documents. For the competition, a total of four methods were submitted. The evaluation method was run for each of these methods on each image from the test dataset and the process was timed. The average time taken by the method was 2.05 seconds.

The time given here includes the full time taken by the method including XML parsing on the input documents and output. That the method takes on average such a short time to operate means that the method is efficient. The short run time allows developers to obtain results in short periods of time and opens the possibility of using the method on much larger datasets which could make the output from the information more useful as it incorporates a greater variety of documents.

## 7.7 Analysis of competition results

Throughout this chapter, the results of the ICDAR 2005 Page Segmentation Competition have been used to evaluate different parts of the system. This section presents a summary and analysis of the competition results gained through evaluation with the new system.

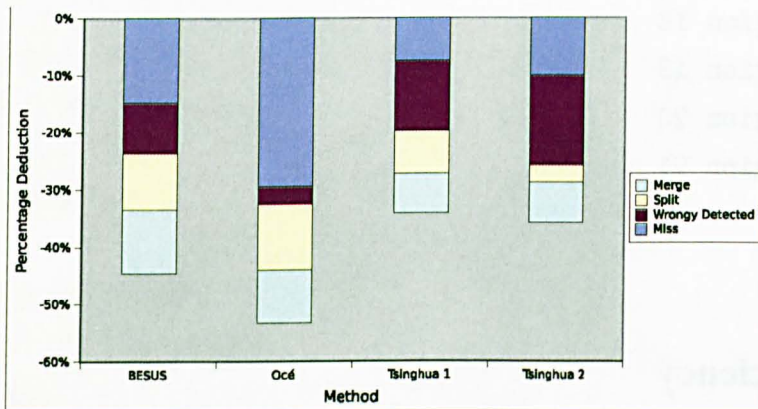


Figure 7.6: The overall errors detected for each method and divided into each of the categories of error measured.

The results of the ICDAR 2005 Page Segmentation Competition when evaluated with the new evaluation method presented here are given in table 7.4 and shown visually in figure 7.6.

Figure 7.6 contains the same overall results as seen previously in figure 7.4 but is presented from an alternative viewpoint. The previous figure presented the overall segmentation score while this figure shows the overall error rate (i.e. 100% - segmentation score).

The error rates here are divided into each of the different types of mistake. This shows visually the contribution of each different type of error to the each method's evaluation result.

From the graph, it may be seen that the worst performing method here, Océ, has a particularly significant problem with missing regions, which contributed 29.4% to the overall errors for that method, significantly worse than the next best method in that category, BESUS with

Method	Total Mistakes	Missed	Wrongly Detected	Split	Merge
BESUS	44.6%	14.8%	8.7%	9.9%	11.2%
Océ	53.2%	29.4%	2.9%	11.7%	9.2%
Tsinghua 1	34.0%	7.5%	12.3%	7.4%	6.9%
Tsinghua 2	35.9%	10.1%	15.6%	3.2%	7.0%

Table 7.4: The overall errors detected for each method and divided into each of the categories of error measured.

14.8%. Conversely, the Océ method had far fewer wrongly-detected regions than any other method, contributing only 2.9% to its error rate.

Wrongly-detected and Missed regions are in some sense opposites and the particular prevalence of missed regions with very few wrongly-detected regions may provide evidence that the method is using a conservative threshold when deciding whether or not a part of the image belongs to a region. It may be that this constitutes an area for improvement in the Océ method.

The two Tsinghua methods together perform significantly better overall than the other two methods evaluated. In fact, the two methods perform better than the other methods in three of the four error categories, with very few merged, split and missed regions. The one category of error in which the Tsinghua methods perform poorly is in wrongly-detected regions. Given the otherwise excellent performance of these methods, focussing on improving their performance in this one area could potentially improve the methods' results dramatically.

The BESUS method was the second worst performing method in the test. Unlike the other methods which demonstrate single problems which contribute significantly to their error rates, the errors of the BESUS method are more evenly distributed over the different error types. Missed regions are the single worst category for the BESUS method, reducing its results by 14.8% but the difference between that and other error types is small, given the sample size.

The poor performance in each error category may be an indication that this particular method is still at an early stage of development or that the method has not previously been widely tested with the types and complexities of documents present in the competition evaluation set.

Overall, the results from evaluating the competition entrants using the new evaluation method show that the areas in which each individual method performs best and worst are different and likely are due to the particular approach of each individual method. Similarly, where one method performs particularly poorly, there is usually another method which performs well in that error category but poorly in a different error category.

This suggests that there is no particular area which ought to be the focus of development of layout analysis in general but rather that each individual method has specific issues which, if improved, would improve performance. It may also be that, where one method performs well in one area and another method performs well in another, the methods could be improved by adopting common techniques or, alternatively, that some method of combining the results of different layout analysis methods would give results better than any individual method.

## 7.8 Summary

This chapter has discussed the goals laid out for the new system in the introduction and has compared the new system against these goals. The following chapter concludes the thesis by summarising the features of the new system and discussing avenues for future work.

# Chapter 8

## Conclusion

### 8.1 Overview

The previous chapter contained a validation and evaluation of the performance evaluation method presented in this thesis. This, the concluding chapter, will return to the goals specified for the method in chapter 1 and discuss how well each of these goals is met by the completed system. Finally, there is a discussion of ways in which the system may be expanded upon in the future.

### 8.2 Review of goals

In the introduction, a list of goals for the evaluation system is given. This section will review each of these goals in light of the completed system and assess how well each of these goals are met.

#### 8.2.1 Accuracy

One of the key characteristics desired in performance evaluation systems is that the evaluations be accurate. This stems from several different aspects. First, the dataset on which the evaluation is based must use a representation scheme which allows complex features of documents to be described accurately. Secondly, the evaluation system itself must operate with a degree of accuracy that allows for even complex documents to be evaluated accurately, preserving information where possible.

The dataset upon which this system is based uses an isothetic polygon representation which is equivalent in representative capability to standard polygons. The evaluation system

is capable of operating on arbitrary polygons and converts these into a region interval representation which is capable of representing any document described by arbitrary polygons to a pixel-accurate degree. During the comparison of these interval representations, all of the data involved is maintained throughout the working of the system and contributes to the resulting evaluation in proportion to the area of the region and user-specified weights.

Some previous performance evaluation systems which relied solely on counts of errors encountered were forced to use thresholds to remove very small errors which otherwise would be treated equally to much more serious errors, were they not removed in the processing. The system described here maintains a full description of all errors encountered and, rather than discard small errors, weights them accordingly, allowing the attention of users to be drawn to the most serious errors.

### 8.2.2 Applicability

Another desirable characteristic of performance evaluation systems is that they be applicable to as broad a range of documents as possible. As described in the previous section, the method relies on a region interval representation which allows the accurate representation of the vast majority of modern documents and the segmentations detected from them by modern layout analysis methods.

Most previous region-based performance evaluation methods have been based upon the bounding-box representation which has significant problems with many modern documents. Similarly, previous pixel-based performance evaluation methods relied on using black pixels to identify the useful contents of a document. However, for documents which contain images, it may not be the case that the contents are described in black pixels alone. For documents which contain more than two levels of colour, such methods also have significant problems.

### 8.2.3 Flexibility

It was desired that the final system allow for a degree of flexibility which would allow the system to be used by those who have different requirements of Layout Analysis methods. The system allows for the user to specify region type- or sub-type-specific relative weightings which allow information regarding the user's application scenario to be incorporated into the final evaluation, allowing specific region types to be ignored completely or weighted more or less relative to other region types.

The system incorporates a series of pre-defined application scenarios which are designed to give a set of weights which may be used in specific applications, allowing users in similar



areas to perform evaluations which are tailored to their application area but enabling results evaluated using the same application scenario to be compared.

### 8.2.4 Descriptiveness

Another key feature desired in the system was for the output to contain a level of detail which would aid users requiring more detailed output. Some previous systems focussed simply on providing a number of key statistics to users. For some uses, this is adequate and the current system matches this capability.

However, users working on developing Layout Analysis methods will require a higher level of detail in results which will enable them to identify particular weaknesses in their methods and to improve them. In order to accommodate this, the system provides detailed output at a number of levels in terms of error groupings, individual error descriptions and region, page and dataset-level statistics. Rather than output mere listings of errors, more detailed information is output and more serious errors are weighted more heavily, allowing development effort to be prioritised.

### 8.2.5 Efficiency

A final characteristic which was desired of the final system is that the evaluation be performed efficiently. This may seem to be negatively affected by the goal of accuracy. However, the region representation on which the system is based, region intervals, was selected because it provides a balance between accuracy and efficiency. The comparison between a ground-truth and segmentation region interval representation is similarly efficient.

Given this, the evaluation takes 2-3 seconds on a typical document, including parsing of the input XML documents and all output. This short runtime allows the system to be used on large datasets in a relatively short period of time.

## 8.3 Future work

### 8.3.1 Web interface

One of the problems of performance evaluation which was discussed in the introduction is that the lack of a common evaluation system means the results published by developers of layout analysis methods are often not comparable which, in turn, makes it difficult to form a reliable opinion of the effectiveness of individual methods and the overall maturity of the area. This thesis presents a performance evaluation method which meets the criteria laid out

in the introduction. However, there is still significant work necessary to make it a common evaluation system for the future.

One way of promoting the adoption of a common performance evaluation system is by making it available as widely as possible to researchers. One method of making the system available to as many researchers as possible, who may be using disparate systems, is to make the system available over the internet as a companion to the PRImA Document Dataset discussed in Chapter 4.4.7.

The availability over the internet of a large, general document dataset dedicated to Layout Analysis combined with the internet availability of the performance evaluation system would give rise to some interesting possibilities.

Previous Page Segmentation competitions have run in an off-line fashion whereby potential entrants are invited to download a training dataset several months before the competition begins. They may use this to train their algorithms prior to the release of the testing dataset. However, with the availability of a performance evaluation method, this could enable a feedback loop where developers could test their methods using the evaluation system used for the competition and receive immediate feedback which would allow methods to be refined, thus improving the general quality of results.

### **8.3.2 Adaptation to other document types**

One of the key focal points of this thesis has been to improve upon previous performance evaluation systems to allow more modern, complex documents to be evaluated. Indeed, many of the examples used in this thesis are of modern documents containing complex features selected from the PRImA Layout Analysis Dataset.

A trend in Image Analysis in recent years has been an increase in the amount of research performed on degraded, historical documents. Such documents, due to their age and fragility, often have significantly greater problems with noise and other artefacts than is typical with modern documents.

It would be interesting to see how the system developed for this thesis could be adapted to dealing with the unique challenges present in historical documents.

### **8.3.3 Definition of further application scenarios**

The method presented here has been developed specifically with flexibility in mind. Given this, in section 6.4.3, a number of application scenarios were described which would tailor the results from the performance evaluation to particular application areas.

The list of application scenarios presented is intended to highlight the flexibility of the method and provide a number of application scenarios which would be immediately useful. However, the area of Image Analysis is a diverse one. Given this, the list is not exhaustive at the moment but may be expanded to introduce new application scenarios to deal with existing problems. Indeed, new problems may arise in Image Analysis which may require new application scenarios to be developed in order to provide adequate evaluations.

## 8.4 Summary

This chapter has provided a review of the original goals for the performance evaluation system and a description of how each of these goals has been met in the new system. It concluded with a discussion of avenues for further development of research in the area.



# Bibliography

- [1] A. Antonacopoulos, "Page segmentation using the description of the background," *Computer Vision and Image Understanding*, vol. 7, no. 3, p. 350–369, June 1998.
- [2] A. Antonacopoulos and D. Bridson, "Performance analysis framework for layout analysis methods," in *Proceedings of the 9th International Conference on Document Analysis and Recognition*, Curitiba, Brazil, September 2007, p. 1258–1262.
- [3] A. Antonacopoulos and A. Brough, "*Methodology for flexible and efficient analysis of the performance of page segmentation algorithms*," in *Proceedings of the 5th International Conference on Document Analysis and Recognition*, Bangalore, India, September 1999, p. 451–454.
- [4] A. Antonacopoulos, B. Gatos, and D. Bridson, "ICDAR 2005 page segmentation competition," in *Proceedings of the 8th International Conference on Document Analysis and Recognition*, Seoul, South Korea, August 2005, p. 75–79.
- [5] -----, "ICDAR 2007 page segmentation competition," in *Proceedings of the 9th International Conference on Document Analysis and Recognition*, Curitiba, Brazil, September 2007, p. 1279–1283.
- [6] A. Antonacopoulos, B. Gatos, and D. Karatzas, "ICDAR 2003 page segmentation competition," in *Proceedings of the 7th International Conference on Document Analysis and Recognition*, Edinburgh, United Kingdom, August 2003, p. 688–692.
- [7] A. Antonacopoulos, D. Karatzas, and D. Bridson, "Ground truth for layout analysis performance evaluation," in *Proceedings of the 7th IAPR Workshop on Document Analysis Systems*, Nelson, New Zealand, February 2006, p. 302–311.
- [8] B. Gatos, S. L. Mantzaris, and A. Antonacopoulos, "First international newspaper segmentation contest," in *Proceedings of the 6th International Conference on Document Analysis and Recognition*, Seattle, United States, September 2001, p. 1190–1194.

- [9] M. Hart, "The history and philosophy of project gutenber," Project Gutenberg, August 1992. [Online]. Available: [http://www.gutenberg.org/wiki/Gutenberg:The\\_History\\_and\\_Philosophy\\_of\\_Project\\_Gutenberg\\_by\\_Michael\\_Hart](http://www.gutenberg.org/wiki/Gutenberg:The_History_and_Philosophy_of_Project_Gutenberg_by_Michael_Hart)
- [10] A. K. Jain and S. Bhattacharjee, "Text segmentation using gabor filters for automatic document processing," *International Journal of Imaging Systems and Technology*, vol. 5, no. 3, p. 169–184, June 1992.
- [11] J. Kanai, S. V. Rice, T. A. Nartker, and G. Nagy, "Automated evaluation of ocr zoning," *IEEE Transactions on Pattern Analysis and Machine Understanding*, vol. 17, no. 1, p. 86–90, January 1995.
- [12] K. Kise, A. Sato, and M. Iwata, "Segmentation of page images using the area voronoi diagram," *Computer Vision and Image Understanding*, vol. 70, no. 3, p. 370–382, June 1998.
- [13] J. Liang, I. T. Phillips, and R. M. Haralick, "Performance evaluation of document layout analysis algorithms on the uw dataset," *Proceedings of the SPIE*, vol. 3027, no. 3, p. 149–160, March 1997.
- [14] S. M. Lucas, "Icdar 2005 text locating competition results," in *Proceedings of the 8th International Conference on Document Analysis and Recognition*, Seoul, South Korea, August 2005, p. 134–137.
- [15] S. M. Lucas, A. Panaretos, and L. Sosa, "Icdar 2003 robust reading competitions," *International Journal of Document Analysis and Recognition*, vol. 7, no. 2–3, p. 105–122, July 2005.
- [16] G. Nagy, S. Seth, and M. Viswanathan, "A prototype document image analysis system for technical journals," *Computer*, vol. 25, no. 7, p. 10–22, July 1992.
- [17] T. A. Nartker, S. V. Rice, and S. E. Lumos, "Software tools and test data for research and testing of page-reading ocr systems," in *Proceedings of the 12th International Conference on Document Recognition and Retrieval*, San Jose, United States, January 2005, p. 37–47.
- [18] L. O'Gorman, "The document spectrum for page layout analysis," *IEEE Transactions on Pattern Analysis and Machine Understanding*, vol. 15, no. 11, p. 1162–1172, November 1993.

- [19] L. Peng, M. Chen, C. Liu, X. Ding, and J. Zheng, "An automatic performance evaluation method for document page segmentation," in *Proceedings of the 6th International Conference on Document Analysis and Recognition*, Seattle, United States, September 2001, p. 134–137.
- [20] I. T. Phillips, S. Chen, J. Ha, and R. M. Haralick, "English document database design and implementation methodology," in *Proceedings of the 18th International Conference on Pattern Recognition*. Hong Kong, Hong Kong: IEEE Computer Society, August 2006, p. 872–875.
- [21] R. Reddy and G. StClair, "The million book digital library project," Carnegie Mellon University, December 2001. [Online]. Available: <http://www.rr.cs.cmu.edu/mbdl.doc>
- [22] J. Sauvola and H. Kauniskangas, "Mediateam document database ii, a cd-rom collection of document images."
- [23] F. Shafait, D. Keysers, and T. M. Breuel, "Pixel-accurate representation and evaluation of page segmentation in document images," in *Proceedings of the 18th International Conference on Pattern Recognition*. Hong Kong, Hong Kong: IEEE Computer Society, August 2006, p. 872–875.
- [24] M. Suzuki, S. Uchida, and A. Nomura, "A ground-truthed mathematical character and symbol image database," in *Proceedings of the 8th International Conference on Document Analysis and Recognition*, Seoul, South Korea, August 2005, p. 675–679.
- [25] M. Thulke, V. Märgner, and A. Dengel, "A general approach to quality evaluation of document segmentation results," in *Proceedings of the 3rd IAPR Workshop on Document Analysis Systems*. Nagano, Japan: Springer, November 1998, p. 43–57.
- [26] L. Todoran, M. Worring, and A. W. Smeulders, "The UvA color document dataset," *International Journal of Document Analysis and Recognition*, vol. 7, no. 4, p. 228–240, September 2005.
- [27] United States National Library of Medicine, "Medical article records groundtruth," 2003. [Online]. Available: <http://marg.nlm.nih.gov/>
- [28] L. Vincent, "Google book search: Document understanding on a massive scale," in *Proceedings of the 9th International Conference on Document Analysis and Recognition*. Curitiba, Brazil: IEEE Computer Society, September 2007, p. 819–823.

- [29] B. A. Yanikoglu and L. Vincent, "Pink panther: a complete environment for ground-truthing and benchmarking document page segmentation," *Pattern Recognition*, vol. 31, no. 2, p. 1191–1204, February 1998.



# **Appendix A**

## **Published paper on the PRImA Dataset**

The following pages contain a paper published in the proceedings of the International Workshop on Document Analysis Systems describing the PRImA Document Layout XML Format used as the basis for this research, the Document Dataset based on it, as well as the issues met and tools used in their creation.[7].

## Ground Truth for Layout Analysis Performance Evaluation\*

A. Antonacopoulos<sup>1</sup>, D. Karatzas<sup>2</sup>, and D. Bridson<sup>1</sup>

<sup>1</sup> Pattern Recognition and Image Analysis (PRImA) Research Lab,  
School of Computing, Science and Engineering,  
University of Salford, Manchester, M5 4WT, United Kingdom  
<http://www.primaresearch.org>

<sup>2</sup> School of Electronics and Computer Science,  
University of Southampton, Southampton, SO16 1BJ, United Kingdom  
<http://www.ecs.soton.ac.uk/~dk3>

**Abstract.** Over the past two decades a significant number of layout analysis (page segmentation and region classification) approaches have been proposed in the literature. Each approach has been devised for and/or evaluated using (usually small) application-specific datasets. While the need for objective performance evaluation of layout analysis algorithms is evident, there does not exist a suitable dataset with ground truth that reflects the realities of everyday documents (widely varying layouts, complex entities, colour, noise etc.). The most significant impediment is the creation of accurate and flexible (in representation) ground truth, a task that is costly and must be carefully designed. This paper discusses the issues related to the design, representation and creation of ground truth in the context of a realistic dataset developed by the authors. The effectiveness of the ground truth discussed in this paper has been successfully shown in its use for two international page segmentation competitions (ICDAR2003 and ICDAR2005).

### 1 Introduction

Layout analysis is a very important step in document analysis. Errors made at this stage will propagate in the subsequent OCR and document understanding stages and can adversely impact on the success of the application as a whole.

Over the past two decades a significant number of layout analysis (mostly page segmentation and region classification) approaches have been proposed in the literature. Each approach has been devised for and/or evaluated using relatively narrow-focused application-specific datasets, which more often than not do not reflect the real-world occurrence of documents. As a result, it is difficult to evaluate the practical value of each method and to make a direct comparison between the different approaches.

Whilst the need for objective performance evaluation of layout analysis algorithms is evident, there does not exist a suitable dataset with ground truth that reflects the

---

\* This work was supported by GCHQ (UK Government Communications Headquarters) and the EPSRC (UK Engineering and Physical Sciences Research Council).

realities of everyday documents (widely varying layouts, complex entities, colour, noise etc.). A number of layout analysis approaches in the literature have reported evaluation results based on the University of Washington dataset [1] which mostly contains (relatively stylised) technical article images, a large number of which are synthetic (created by the dataset authors using LaTeX and output as images). It is the view of the authors that such a database can be useful but does not reflect the complexities of the majority of widely available documents.

This lack of a representative and practical (in terms of use) dataset can be attributed mostly to the need to subtly balance wide-ranging issues involved in its design as well as to the effort required in its realisation.

While the design of the dataset architecture is of central importance in terms of its usefulness and usability, the crucial (and most influential) element is the design of the *ground truth*. It should be mentioned, for completeness, that ground truth is defined as a representation of the agreed correct result of the ideal layout analysis method (i.e. the result of the method that, if existed, would put an end to the research problem). The ground truth forms the basis for all comparisons with the output of any layout analysis method to be evaluated.

A significant clarification must be made at this point between *performance evaluation* and *benchmarking*. The former involves in-depth analysis of results and is aimed at providing feedback to developers, the latter usually outputs a single value that is used to compare between approaches. Clearly, for in-depth performance evaluation, a more thorough specification and design is required for the dataset in general and for the ground truth in particular.

This paper presents and discusses the issues related to the design, representation and creation of ground truth in the context of the layout analysis performance evaluation dataset developed by the authors. In contrast to previous approaches (the most prominent of which is [1]), the proposed dataset is not only realistic in the selection of documents but it has significant flexibility in the description and use of ground truth. A more accurate region representation scheme is used in favour of using rectangles (unable to describe complex-shaped regions) but without sacrificing ease of use or performance. The additional information describing the physical and logical characteristics of regions ensures the applicability of the ground-truth to a wide range of evaluation scenarios and anticipated future needs (as evidenced by current developments).

The remainder of the paper starts with a brief description of the context within which the ground truth needs to be designed, created and used. In this respect, Section 2 describes the performance evaluation framework while Section 3 presents aspects of the dataset. The main considerations for the design of successful ground truth are discussed in Section 4. The specification of the ground truth and its XML representation are introduced in Section 4.1. An overview of a software tool designed by the authors to support the ground truth creation is given next (Section 4.3). Section 5 concludes the paper.

## 2 Performance Evaluation Framework

One of the important issues to address and one of the advantages of the ground truth representation described in this paper is the flexibility of its use within different

304 A. Antonacopoulos, D. Karatzas, and D. Bridson

performance evaluation contexts. These can range from simple listings of regions missed/detected etc. to sophisticated evaluation of scenarios (e.g. the detection of headlines and separators) with configurable penalties etc.

A brief description of this wider perspective, in the form of the framework being developed by the authors, is given here to highlight the needs that ground truth has to fulfil within a wider, more-demanding application. The most important objective of the framework is to provide the (layout analysis) algorithm developer with an in-depth analysis of the performance of the method being evaluated. Detailed statistical information is given on the ability of a method in terms of correctly detected, merged, split, partially or wholly missed regions (along with combinations of these conditions as well as the incorrect detection of noise as valid regions) [2]. *Goal-oriented* performance evaluation is enabled through the creation of scenarios (application of sets of weights on the detected errors). An example of this can be when an OCR developer is interested in not missing any text regions and in not merging text regions across columns etc. (to preserve the reading order), while they may not assign high value to the accurate detection of graphic regions.

At a higher semantic level, a scenario may involve the evaluation of logical as well as physical layout characteristics. For instance, in an indexing application the developer may be interested in correctly locating figure captions (for indexing photographs), or article titles and dates (for indexing newspaper articles).

Moreover, the framework is able to summarise the performance of a method by providing scores (based on scenarios) at different levels as required. For instance, a developer who needs to assess the resulting improvement of a newly introduced modification may customise the framework to provide them with both an overall scenario evaluation score but with detailed scores for the tasks that are most affected by the given modification.

It is therefore important that the ground truth must hold information that supports these evaluation tasks.

### 3 Dataset

In its most crude form, a performance evaluation dataset comprises a set of images and associated ground truth (for each image). The dataset on which layout analysis methods are evaluated has an obvious bearing on the relevance of the evaluation results. This section briefly presents the dataset developed by the authors with two key objectives in mind. First, to give the reader a broader understanding of the contextual issues for ground-truth design in terms of the choice of documents (page images) it needs to describe. Second, to provide an understanding of the overall architecture of which ground truth is part (and within which it is used).

The choice of documents to include in a dataset has to fulfil two major requirements. First, the types (categories) of documents have to be representative of everyday occurrences. Second, the proportion of documents (population in the dataset) between categories should reflect realistic usage and at the same time the documents in each category must be sufficiently varied and numerous to enable meaningful evaluation for specific applications.

To that effect, the authors have established a detailed taxonomy of existing documents (text carriers), based on physical and logical layout characteristics (about 21 document types and 80 subtypes). Document types range from official documents (e.g., certificates) to various drawings and maps, to forms, books, tickets and text in natural scenes, to name but a few. However, certain types of document are more widely distributed and are more targeted by application developers. These are documents that contain information that a wide variety of users need to extract. Examples are office documents, magazine pages, advertisements and technical articles. The dataset created by the authors reflects this situation by containing more instances of these types of document.

It should be noted that the layouts of these types of document vary considerably. Office documents and technical articles have more structured layouts that usually follow simple formatting rules. On the other hand, magazine pages have more complex layouts and advertisements even more so. As it will be seen in the next section, the complexity of layout regions is one of the deciding factors in ground truth design.

The dataset is organised in two broad layers of functionality. The outer layer is a database holding certain physical and administration attributes for each document page in the dataset. Physical attributes include dimensions, the presence (or absence) of colour, whether or not the document is single or multi-columnned, the (main)

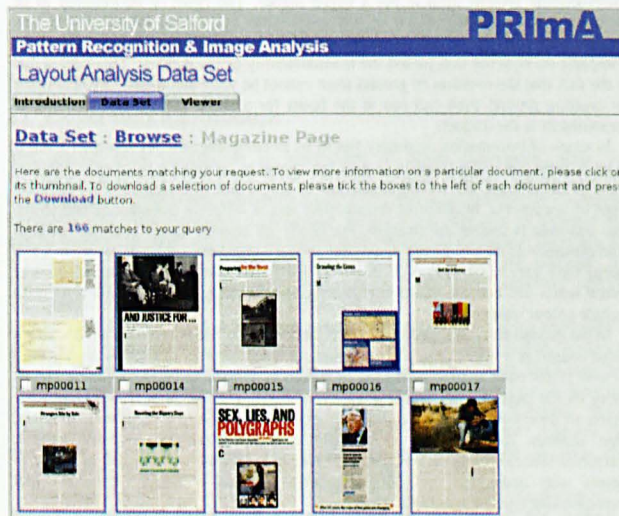


Fig. 1. The Web interface for browsing a specific document category

306 A. Antonacopoulos, D. Karatzas, and D. Bridson

language of the text, the reading direction, the resolution of the image as well as a characterisation of the complexity of the layout. All these attributes are deemed to be interesting for searching and selecting sets of documents for evaluation (they represent major factors that influence layout analysis methods). Administration attributes are mostly used by the dataset keepers and include authorship and source information, copyright information etc. A web front-end enables both searching of the dataset based on the above attributes as well as browsing of the dataset according to document types (as defined in the established taxonomy). A screenshot of the web interface (browsing magazine pages) can be seen in Fig. 1.

The inner layer of functionality comprises the image-ground truth pair. An image file (or two, as in the case of colour/greyscale documents both the original and a bilevel copy are kept) and its corresponding ground truth description file are linked to each record in the database. The design issues and characteristics of the ground truth description are discussed in the next section.

The current dataset (to be made freely available to researchers) can be found in:  
<http://www.prima.cse.salford.ac.uk/dataset/>.

#### 4 Ground Truth

It can be appreciated that, given a document image, the description of the corresponding ground truth is not a trivial matter. The *types of information* to be included and the *representation* of this information are crucial for successful use. Another important underlying factor is the significant cost of creating ground truth, as it impacts on both the design and the maintainability of the dataset. This cost is due to the fact that the creation of ground truth cannot be fully automated. Typical times for creating ground truth can run in the hours for a single page (from scanning to commitment in the dataset).

In terms of information, it simply has to be as comprehensive as possible. Even if some information is not filled-in or may not appear to be directly relevant to familiar types of documents, the infrastructure has to be present in anticipation of different types of documents, in different scripts, with text in different orientations and so on. One example is colour information. Practically all current layout analysis methods (and certainly all the prominent ones) deal almost exclusively with bilevel or (in a few cases) with grey scale images. It is almost inevitable, however, that the analysis of colour scans will become increasingly necessary and therefore the ground truth must include colour information.

In the ground truth described here, information is recorded regarding the document (page image) as a whole (e.g. physical characteristics, number of regions present etc.) as well as for each individual region. A region is defined to be the smallest logical entity on the page. For the purpose of layout analysis methods, it is sufficient for a region to represent a single paragraph in terms of text (body text, header, footnote, page number, caption etc.), or a graphic region (half-tone, line-art, images, horizontal/vertical ruling etc.). Composite elements of a document, such as tables or figures with embedded text, are considered each as a single region (of that corresponding type such as table, chart etc.).

The region-representation scheme plays a critical role in the efficiency and accuracy of the performance analysis strategy. For the comparison between regions (a ground truth region against a region resulting from a method to be evaluated), bounding rectangles are the most efficient representation. However, complex-shaped regions cannot be accurately represented by bounding rectangles. The proposed scheme describes regions using isothetic (having only horizontal and vertical edges) polygons [3]. This representation of regions is very accurate and flexible since each region can have any size, shape and orientation. Furthermore, a region, whose contour is an isothetic polygon, can be represented by a number of rectangular horizontal intervals whose height is determined by the corners of its contour polygon (effectively achieving decomposition into rectangles). This interval structure makes checking for inclusion and overlaps, and calculation of area, possible with very few operations, thus approximating the efficiency of rectangles [4].

In general, ground truth must fulfil the following objectives:

- *Accuracy*, both in terms of absence of human errors and in the inherent ability to represent complex information.
- *Richness of information*, to enable various evaluation scenarios.
- *Efficiency of comparison*, to enable evaluation using large datasets.
- *Ease of understanding*, in terms of representation organisation to facilitate maintenance and use.
- *Ease of creation*, in terms of the ability to achieve the above objectives with the use of a specially designed ground-truthing tool (see below).
- *Anticipation of future requirements*, in terms of extensibility to avoid obsolescence.

#### 4.1 Ground Truth Representation

The ground truth information is represented in XML (addressing, thus, the representation-related goals listed in the previous section). Figure 2 shows a ground truth example of a document containing a single text region (simplified for illustration purposes). The main element is a *Document*, which is the only type of element that can be found in an XML file after the header lines. Inside the Document (between the <document> and </document> tags) two types of element are allowed: the *Document Summary* and a number of *Pages*. The document summary section specifies how many pages there are in the document.

Each page is represented as a separate element, and information about each page is given between the <page> and </page> tags. The image filename attribute is used to indicate the name of the image file on which the ground truth is based. Each page can be decomposed into a number of regions. In the current ground truth version, there are ten distinct types of regions defined: *Text*, *Image*, *Line Drawing*, *Graphic*, *Table*, *Chart*, *Separator*, *Maths*, *Noise* and *Frame*. The "page summary" contains the number of occurrences of each type of region in the page, while the page size attributes define the width and height (in pixels) of the page.

Each region must contain a unique ID number to identify it within the document. A number of attributes (their occurrence depending on the type of the region) is

308 A. Antonacopoulos, D. Karatzas, and D. Bridson

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE document SYSTEM
'http://www.prima.cse.salford.ac.uk/dataset/documentlayout.dtd>

<document>
  <document_summary no_pages="1"/>
  <page page_id="1" image_filename="mp00088bw.tif">
    <page_summary no_text_regions="22"
      no_image_regions="0" no_line_drawing_regions="0"
      no_graphic_regions="0" no_table_regions="0"
      no_chart_regions="0" no_separator_regions="0"
      no_maths_regions="0" no_frame_regions="0"
      no_noise_regions="0"/>
    <page_pixel_size width="2340" height="3135"/>
    <text_region id="1" txt_orientation="0"
      txt_reading_direction="Left_To_Right"
      txt_leading="" txt_kerning=""
      txt_font_size="12" txt_type="Paragraph"
      txt_colour="Black" txt_reverse_video="No"
      txt_indented="No" txt_primary_lang="English"
      txt_secondary_lang="None"
      txt_primary_script="Latin"
      txt_secondary_script="None" txt_bgcolour="White"
      txt_reading_orientation="0">
      <coords no_coords="4">
        <point x="10" y="10"/>
        <point x="20" y="10"/>
        <point x="20" y="20"/>
        <point x="10" y="20"/>
      </coords>
    </text_region>
  </page>
</document>

```

Fig. 2. Example of ground truth representation

optional. These attributes describe as many characteristics of the region as possible. Various attributes relevant to text regions are shown in the example of Fig. 2. It is mandatory that each region contains coordinate sets that define its outline (isothetic polygon).

The full Document Type Definition (DTD) file which defines the XML representation of ground-truth information can be found at:

<http://www.prima.cse.salford.ac.uk/dataset/documentlayout.dtd>.

#### 4.2 Ground Truth Creation

To enable the creation of detailed and flexible ground truth, a semi-automated tool has been designed by the authors. When designing this tool the decision was made to provide full flexibility and the focus was placed on the *creation* of ground truth, rather than the *correction* of the results of a first-pass segmentation process. This is a pragmatic approach to the problem, stemming from previous experience of the authors with ground-truthing [5]. The crucial observation was that the time spent in correcting the errors of segmentation is more often than not significantly longer than following a bottom-up approach to build ground truth information and fewer errors are made (users tend to miss errors made by the first-pass segmentation process).



It is worth mentioning at this point that there are other approaches to "ground truth" creation in the literature (e.g., [6]). In these cases though, the tools are meant to be used in the final stages of an automated process to ensure the validity of the outcome of the conversion process of a paper document into electronic form, while the "ground truth" information sought is also application specific and lacks the depth and breadth needed for performance evaluation.

The ground-truthing tool "*Aletheia*" (from the Greek word for "truth") operates on the bilevel version of the document images and comprises functionality to perform connected component analysis and, subsequently, combine the resulting components into regions (as required by the ground truth specification). In addition, it provides the necessary interface to label the regions identified and specify an appropriate set of attributes for each, customized according to its type. Finally, the software can export the ground truth as an XML file, which fully conforms to the ground truth specification.

Before any editing operations become available, the software performs a connected component analysis of the document image. A fast one-pass algorithm is employed for that purpose. The connected components identified in the image are the base units for the construction of regions. Each target region will comprise a list of components, and will be described by a boundary which will enclose only the specified components, and possibly some white space.

There are four supported methods to group together connected components into a region that affect the way the boundary of the region is derived offering different levels of flexibility. At the lower level the user can select the components of a region one by one. The boundary of the region is then defined as the minimum bounding rectangle which encloses the selected components. A higher-level approach is to use a drag-and-resize operation to specify a rectangle and select all the components inside it. The system then allows the user to either adopt the specified rectangle as the boundary of the region, or shrink the specified rectangle in order to produce the minimum rectangle in the same manner as before. Finally, in order to address cases where complicated region shapes are necessary, the software offers the option to use a freehand drawing method to select components. In this case the user defines a polygon by successively selecting its corner points. The isothetic rectangle boundary in this case is calculated based on the initial polygon, which is reduced in such a way so that most of the white space is removed.

*Aletheia* also offers more advanced region-editing functions, for instance to combine regions, or to combine existing regions with individual components, while regions can always be dissolved into their constituent components. Following the bottom up approach described above, a higher (region) level segmentation of a document can be obtained in a few minutes.

Subsequent to geometrically defining the regions of the document page the user has to label the resulted regions and define the associated attributes. According to the ground truth specification, *Aletheia* allows each region to be of any of the ten region types defined. By right-clicking a region, the user is presented with a dialog box, which lists the type and associated attributes of each region. The user can then select the type of the region from a drop down list, and specify the values for all attributes associated with the region type. The only attribute the user cannot control is the region ID number, which is assigned and managed automatically by the software. Figure 3 shows the attributes dialog for a text region, and a line drawing region.

310 A. Antonacopoulos, D. Karatzas, and D. Bridson

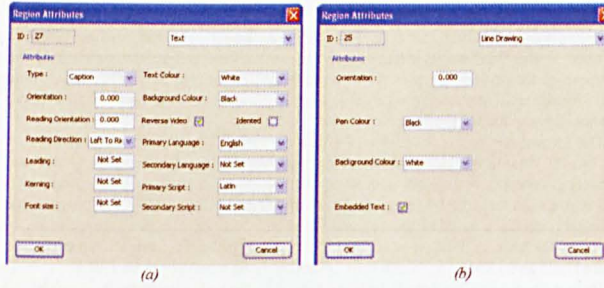


Fig. 3. The attribute dialogs for (a) a Text region and (b) a Line Drawing region

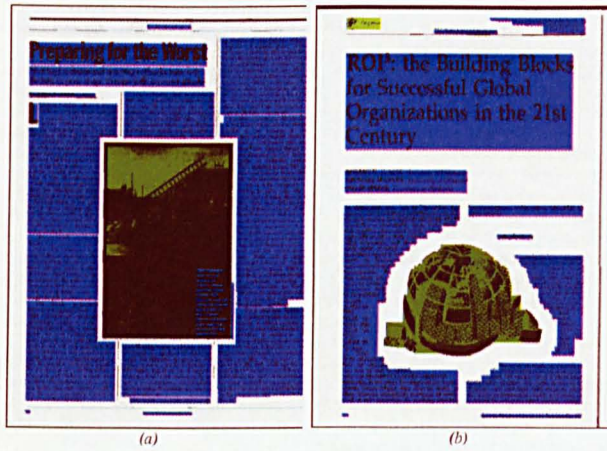


Fig. 4. Examples of the final ground-truth for (a) a Magazine page and (b) a Technical Article page

Figure 4 shows two instances of ground-truth regions created with *Aletheia*. The software visualises the ground-truth information by assigning different colours to regions depending on their type. This facilitates the process of labelling the regions, since the user can easily identify any unprocessed regions. Any regions or components that have not been labelled are automatically marked as noise regions.

Finally, *Aletheia* offers two options for storing the final ground truth description. The first is to export it as an XML file (a series of individual regions, along with their boundaries and detailed attributes) which fully conforms to the ground truth specification as described above. The second option is to save the ground truth representation in the software's own format, which has the advantage of preserving the actual components in addition to the higher-level information, thus facilitating more powerful editing at a later time.

## 5 Concluding Remarks

This paper has introduced and discussed a number of important issues surrounding ground truth for the evaluation of the performance of layout analysis methods. The focus was on the design, representation and creation stages in the context of a new dataset developed by the authors. The resulting ground truth is the product of the authors' effort over the past few years and reflects their experience with performance evaluation. The ground truth created has been successfully used as the basis for two international competitions, held under the auspices of the International Conference on Document Analysis and Recognition in 2003 [7] (in an earlier version) and 2005 [8].

## References

1. Philips, I.T., Chen, S., Ha, J., and Haralick, R.M. English Document Database Design and Implementation Methodology. In *Proceeding of the 2nd Annual Symposium on Document Analysis and Retrieval* (UNLV, USA, 1993), 65–104.
2. Antonacopoulos, A. and Brough, B. Methodology for Flexible and Efficient Analysis of the Performance of Page Segmentation Algorithms. In *Proceedings of the 5th International Conference on Document Analysis and Recognition (ICDAR'99)*, (Bangalore, India, 1999), IEEE-CS Press, 451–454.
3. Antonacopoulos, A. Page Segmentation Using the Description of the Background. *Computer Vision and Image Understanding*, Vol. 70, No. 3 (1998), 350–369.
4. Antonacopoulos, A. and Ritchings, R.T. Representation and Classification of Complex-Shaped Printed Regions Using White Tiles. In *Proceedings of the 3rd International Conference on Document Analysis and Recognition (ICDAR'95)* (Montreal, Canada, 1995), IEEE-CS Press, 1132–1135.
5. Antonacopoulos, A., and Meng, H. A Ground-Truthing Tool for Layout Analysis Performance Evaluation. In *Document Analysis Systems V*, D. Lopresti, J. Hu and R. Kashi (Eds.), Springer Lecture Notes in Computer Science, LNCS 2423, 2002, 236–244.
6. Simske, S.J., and Sturgill, M.. A Ground-Truthing Engine for Proofsetting, Publishing, Repurposing and Quality Assurance. In *Proceedings of the 2003 ACM Symposium on Document Engineering (DocEng'03)* (Grenoble, France, 2003), ACM Press, 150–152.
7. Antonacopoulos, A., Gatos, B., and Karatzas, D. ICDAR2003 Page Segmentation Competition. In *Proceedings of the 7th International Conference on Document Analysis and Recognition (ICDAR2003)* (Edinburgh, UK, August 2003), IEEE-CS Press, 688–692.
8. Antonacopoulos, A., Gatos, B., and Bridson, D. ICDAR2005 Page Segmentation Competition. In *Proceedings of the 8th International Conference on Document Analysis and Recognition (ICDAR2005)* (Seoul, South Korea, August 2005), IEEE-CS Press, pp. 75–79.



## **Appendix B**

### **The PRImA Document Layout XML Format**

The following pages contain the Document Type Definition for the PRImA XML Document Layout Format. Greater detail on the design decisions made for the format may be found in the published paper in Appendix A. The author was one of many contributors to the design of the format, alongside Dr. Apostolos Antonacopoulos, Dr. Dimosthenis Karatzas, Mark Ellis and John Spafford.

```

<! v4 DTD >

<! Element declarations >

<!ELEMENT document (document_summary, page+)>

<!ELEMENT document_summary EMPTY>
<!ELEMENT page_summary EMPTY>
<!ELEMENT point EMPTY>
<!ELEMENT page_pixel_size EMPTY>

<!ELEMENT page (page_summary, page_pixel_size, (text_region*,
  image_region*, line_drawing_region*, graphic_region*,
  table_region*, chart_region*, separator_region*,
  maths_region*, frame_region*, noise_region*))>
<!ELEMENT text_region (coords)>
<!ELEMENT image_region (coords)>
<!ELEMENT line_drawing_region (coords)>
<!ELEMENT graphic_region (coords)>
<!ELEMENT table_region (coords)>
<!ELEMENT chart_region (coords)>
<!ELEMENT separator_region (coords)>
<!ELEMENT maths_region (coords)>
<!ELEMENT noise_region (coords)>
<!ELEMENT frame_region (coords, (text_region*, image_region*,
  line_drawing_region*, graphic_region*, table_region*,
  chart_region*, separator_region*, maths_region*,
  noise_region*, frame_region*))>
<!ELEMENT coords (point*)>

<! attribute declarations >

<!ATTLIST
  coords no_coords CDATA #REQUIRED
>

<!ATTLIST

```

```
document_summary no_pages CDATA "1"
>
<!ATTLIST page_summary
  no_text_regions          CDATA "0"
  no_image_regions        CDATA "0"
  no_line_drawing_regions CDATA "0"
  no_graphic_regions      CDATA "0"
  no_table_regions        CDATA "0"
  no_chart_regions        CDATA "0"
  no_separator_regions    CDATA "0"
  no_maths_regions        CDATA "0"
  no_frame_regions        CDATA "0"
  no_noise_regions        CDATA "0"
>
<!ATTLIST point
  x CDATA #REQUIRED
  y CDATA #REQUIRED
>
<!ATTLIST page_pixel_size
  width  CDATA #REQUIRED
  height CDATA #REQUIRED
>
<!ATTLIST page
  page_id          CDATA #REQUIRED
  image_filename  CDATA #IMPLIED
>
<!ATTLIST frame_region
  id CDATA #REQUIRED
>
<!ATTLIST noise_region
  id CDATA #REQUIRED
>
```

<code>&lt;!ATTLIST text_region</code>	
<code>id</code>	<code>CDATA #REQUIRED</code>
<code>txt_orientation</code>	<code>CDATA "0"</code>
<code>txt_reading_direction</code>	<code>( Left_To_Right   Right_To_Left   Top_To_Bottom   Bottom_To_Top ) "Left_To_Right"</code>
<code>txt_leading</code>	<code>CDATA #IMPLIED</code>
<code>txt_kerning</code>	<code>CDATA #IMPLIED</code>
<code>txt_font_size</code>	<code>CDATA "12"</code>
<code>txt_type</code>	<code>( Paragraph   Heading   Sub_Heading   Sentence   Caption   Header   Footer   Page_Number   Quote   Drop_Capital   Credit ) "Paragraph"</code>
<code>txt_colour</code>	<code>( Black   Red   White   Green   Blue   Yellow   Orange   Pink   Grey   Turquoise   Indigo   Violet   Cyan   Magenta ) "Black"</code>
<code>txt_reverse_video</code>	<code>( Yes   No ) "No"</code>
<code>txt_indented</code>	<code>( Yes   No ) "No"</code>
<code>txt_primary_language</code>	<code>( Afrikaans   Albanian   Amharic   Arabic   basque   Bengali   Bulgarian   Cambodian   Cantonese   Chinese   Czech   Danish   Dutch   English   Estonian   Finnish   French   German   Greek   Gujarati   Hebrew   Hindi   Hungarian   Icelandic   Gaelic   Italian   Japanese   Korean   Latvian   Malay   Norwegian   Polish   Portuguese   Punjabi   Russian   Spanish   Swedish   Thai   Turkish   Urdu   Welsh   None ) "English"</code>
<code>txt_secondary_language</code>	<code>( Afrikaans   Albanian   Amharic   Arabic   basque   Bengali   Bulgarian   Cambodian   Cantonese   Chinese   Czech   Danish   Dutch   English   Estonian   Finnish   French   German   Greek   Gujarati   Hebrew   Hindi   Hungarian   Icelandic   Gaelic   Italian   Japanese   Korean   Latvian   Malay   Norwegian   Polish   Portuguese   Punjabi   Russian   Spanish   Swedish   Thai   Turkish  </code>



```

    Urdu | Welsh | None) "None"
txt_primary_script (Arabic | Bengali | Cyrillic | Devangari |
                  Ethiopic | Greek | Gujarati | Gurmukhi |
                  Hebrew | Latin | Simplified_Chinese | Thai |
                  Traditional_Chinese | None) "Latin"
txt_secondary_script (Arabic | Bengali | Cyrillic | Devangari |
                    Ethiopic | Greek | Gujarati | Gurmukhi |
                    Hebrew | Latin | Simplified_Chinese | Thai |
                    Traditional_Chinese | None) "None"
txt_bgcolour (Black | Red | White | Green | Blue | Yellow |
             Orange | Pink | Grey | Turquoise | Indigo |
             Violet | Cyan | Magenta) "White"
txt_reading_orientation CDATA "0"
>

<!ATTLIST image_region
  id CDATA #REQUIRED
  img_colour_type (Black_And_White | 4_Bit_Greyscale |
                 8_Bit_Greyscale | 4_Bit_Colour | 8_Bit_Colour |
                 16_Bit_Colour | 24_Bit_Colour | 32_Bit_Colour)
                 "Black_And_White"
  img_orientation CDATA "0"
  img_emb_text (Yes | No) "No"
  img_bgcolour (Black | Red | White | Green | Blue | Yellow | Orange |
              Pink | Grey | Turquoise | Indigo | Violet | Cyan |
              Magenta) "White"
>

<!ATTLIST line_drawing_region
  id CDATA #REQUIRED
  drwg_emb_text (Yes | No) "No"
  drwg_orientation CDATA "0"
  drwg_pen_colour (Black | Red | White | Green | Blue | Yellow | Orange |
                 Pink | Grey | Turquoise | Indigo | Violet | Cyan |
                 Magenta) "Black"
  drwg_bgcolour (Black | Red | White | Green | Blue | Yellow | Orange |
               Pink | Grey | Turquoise | Indigo | Violet | Cyan |
               Magenta) "White"

```

```

>
<!--ATTLIST graphic_region
  id          CDATA      #REQUIRED
  gfx_type    (Logo | Letterhead | Handwritten_Annotation |
              Stamp | Signature | Paper_Grow | Punch_Hole | Other)
              #IMPLIED
  gfx_emb_text (Yes|No) "No"
  gfx_orientation CDATA    "0"
  gfx_no_colours CDATA    "0"
>

<!--ATTLIST table_region
  id          CDATA      #REQUIRED
  tbl_rows    CDATA      #IMPLIED
  tbl_columns CDATA      #IMPLIED
  tbl_line_colour (Black | Red | White | Green | Blue | Yellow |
                 Orange | Pink | grey | Turquoise | Indigo | Violet |
                 Cyan | Magenta) "Black"
  tbl_orientation CDATA    "0"
  tbl_line_separators (Yes|No) "Yes"
  tbl_bgcolour (Black | Red | White | Green | Blue | Yellow |
              Orange | Pink | grey | Turquoise | Indigo | Violet |
              Cyan | Magenta) "White"
  tbl_emb_text (Yes|No) "Yes"
>

<!--ATTLIST chart_region
  id          CDATA      #REQUIRED
  chart_emb_text (Yes|No) "Yes"
  chart_orientation CDATA    "0"
  chart_no_colours CDATA    "0"
  chart_type    (Pie | Line | Other) #IMPLIED
  chart_bgcolour (Black | Red | White | Green | Blue | Yellow | Orange |
                Pink | Grey | Turquoise | Indigo | Violet | Cyan |
                Magenta) "White"
>

```

```
<!ATTLIST separator_region
  id          CDATA #REQUIRED
  sep_orientation CDATA "0"
  sep_colour   ( Black | Red | White | Green | Blue | Yellow | Orange |
                Pink | Grey | Turquoise | Indigo | Violet | Cyan |
                Magenta ) "Black"
  sep_bgcolour ( Black | Red | White | Green | Blue | Yellow | Orange |
                Pink | Grey | Turquoise | Indigo | Violet | Cyan |
                Magenta ) "White"
>

<!ATTLIST maths_region
  id          CDATA #REQUIRED
  maths_bgcolour ( Black | Red | White | Green | Blue | Yellow | Orange |
                  Pink | Grey | Turquoise | Indigo | Violet | Cyan |
                  Magenta ) "White"
  maths_orientation CDATA "0"
>
```



## **Appendix C**

### **Published paper on the ICDAR 2005 Competition**

The following pages contain a paper published in the proceedings of the International Conference on Document Analysis and Recognition describing the running and results of the ICDAR 2005 Page Segmentation Competition, based on the dataset described in appendix A.[4].

## ICDAR2005 Page Segmentation Competition

A. Antonacopoulos<sup>1</sup>, B. Gatos<sup>2</sup> and D. Bridson<sup>1</sup>

<sup>1</sup>Pattern Recognition and Image Analysis (PRIMA) Research Lab  
School of Computing, Science and Engineering, University of Salford, Manchester, M5 4WT, United Kingdom  
<http://www.primaresearch.org>

<sup>2</sup>Computational Intelligence Laboratory, Institute of Informatics and Telecommunications,  
National Center for Scientific Research "Demokritos", GR-153 10 Agia Paraskevi, Athens, Greece  
<http://www.iit.demokritos.gr/cil>

### Abstract

*There is an established need for objective evaluation of layout analysis methods, in realistic circumstances. This paper describes the Page Segmentation Competition (modus operandi, dataset and evaluation criteria) held in the context of ICDAR2005 and presents the results of the evaluation of four candidate methods. The main objective of the competition was to compare the performance of such methods using scanned documents from commonly-occurring publications. The results indicate that although methods seem to be maturing, there is still a considerable need to develop robust methods that deal with everyday documents.*

### 1 Introduction

Layout analysis methods—page segmentation in particular—continue to be reported in the literature on a frequent basis, despite this being one of the most mature sub fields of Document Image Analysis. It is not difficult to see that the reason for this is that the problem is far from being solved. Successful methods have certainly been reported but, frequently, those are devised with a specific application in mind and are fine-tuned to the test image data set used by its authors. The wider gamut of documents encountered in real-life situations is far wider than the target applications of most methods.

There is no doubt that, for a given application, or for a generic selection of real-life documents, it would be desirable to obtain an objective evaluation of the performance of different layout analysis methods. Such a direct comparison between algorithms is not straightforward as it requires both the creation of suitable ground truth (a relatively laborious and precise task) as well as the definition of a set of objective evaluation criteria (and a method to analyse them).

This competition focuses on the evaluation of page segmentation and region classification subsystems. To the best of the Authors' knowledge, this is only the second instance of an international generic layout analysis competition (the first being the ICDAR2003 Page

Segmentation Competition [1]). It should be mentioned that a relatively close previous instance, focussing on a specific application domain, was the First International Newspaper Page Segmentation Contest [2] held by the Authors in the context of ICDAR2001. Prior to that, an evaluation of page segmentation (as part of OCR systems) was performed at UNLV [3], based on the results of OCR. That approach, however, cannot not be strictly considered to evaluate layout analysis methods since the OCR-based evaluation does not give sufficient information on the performance of page segmentation and region classification and is only applicable to regions of text (or text-only documents).

The motivation for this competition was the evaluation of page segmentation and region classification methods in realistic circumstances. By realistic it is meant that the participating methods are applied to scanned documents from a variety of sources, occurring in real life. This is in contrast to the majority of datasets and reports of results using mostly structured documents (e.g., technical articles).

The competition and its modus operandi is described next. In Section 3, an overview of the dataset and the ground-truthing process is given. The performance evaluation method and metrics are described in Section 4, while each of the participating methods is summarised in Section 5. Finally, the results of the competition are presented and the paper is concluded in Sections 6 and 7, respectively.

### 2 The competition

The objective of the competition was to evaluate layout analysis (page segmentation and region classification) methods using scanned documents from commonly-occurring publications. While there is a comparative assessment element involved, the real advantage is an initial look in the performance of different classes of methods (e.g., connected component analysis, morphological processing, analysis of background etc.) in identifying different types of regions in a variety of documents.

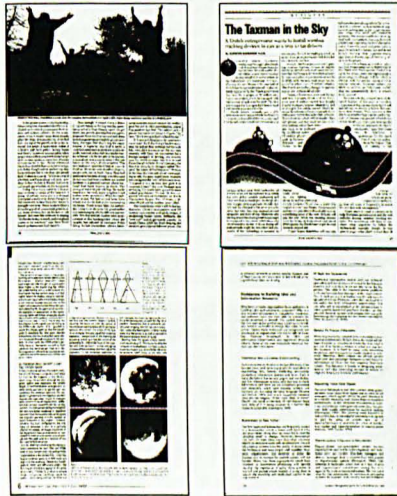


Figure 1. Sample page images from the training dataset.

The competition run in an off-line mode. The authors of candidate methods registered their interest in the competition and downloaded the *training* dataset (document *images* and associated *groundtruth*). One week before the competition closing date, registered authors of candidate methods were able to download the document *images* of the *evaluation* dataset. At the closing date, the organisers received the results of the candidate methods, submitted by their authors in a pre-defined format. The organisers then evaluated the submitted results.

It should be noted that the off-line mode is based on trust that the results submitted by the methods' authors are genuine. This trust is even more necessary if the evaluation system is publicly available. In this case, the evaluation system was not published (only the principles) and above all, the organisers have faith in the authors' scientific integrity.

### 3 The dataset

For any performance evaluation approach, the Achilles' heel is the availability of realistic and accurate ground truth. As ground-truthing cannot (by definition) be fully automated, it remains a laborious and, therefore, expensive process. One approach would be to use synthetic data [4]. It is the authors' opinion, however, that

for the realistic evaluation of layout analysis methods, 'real' scanned documents give a better insight.

It should be noted that ground truth there is scarce availability of ground truth for the evaluation of methods analysing complex layouts (e.g., having non-rectangular regions). Such a dataset was created for the ICDAR2003 competition [1]. However, the current competition was based on a subset of a significantly updated dataset. This dataset, which will shortly be released by the PRIMA research lab, contains richer ground truth (in a correspondingly updated XML format) that provides a very wide range of information on region attributes (physical and logical).

Although the dataset contains instances (images and ground truth) of an exhaustive list of document types it does focus, however, (for meaningful evaluation purposes) on the most heavily used (in terms of information content and need to analyse) types of documents, such as office documents, magazine pages, advertisements and technical articles.

For the competition, a subset of documents was selected that reflected both realism in their frequent occurrence and, at the same time, the existence of sufficiently general interest to analyse them.



Figure 2. Sample page image from the training dataset showing superimposed description of region contours.

Furthermore, a balance had to be achieved between logistics (a manageable number of document images) and tractability for current methods. The decision was, therefore, made to focus on a cross section of 26 page images, comprising 30% technical articles (not necessarily with Manhattan layouts) and 70% magazine

pages. It should be noted that also for reasons of tractability, the competition images were bilevel (in the general dataset the original images are in colour). A sample of page images given as part of the training dataset can be seen in Fig. 1.

The ground-truth of each page image is an XML file (defined as part of the general dataset) that contains image and layout-specific information as well as the description of the regions in terms of isothetic (having only horizontal and vertical edges) polygons. The ground-truth for the competition was produced using a semi-automated tool developed by the authors. An XML viewer was developed for examining the images and the corresponding ground-truth XML, and was distributed to the competition participants. Another sample page image with the corresponding description of regions superimposed as isothetic polygons can be seen in Fig. 2.

The types of regions defined for the competition (simplified from the total number of different types in the general dataset) are:

- text,
- graphics,
- line-art,
- separator, and
- noise.

#### 4 Performance evaluation

The performance evaluation method used is based on counting the number of matches between the entities detected by the algorithm and the entities in the ground truth [5-7]. We use a global MatchScore table for all entities whose values are calculated according to the intersection of the ON pixel sets of the result and the ground truth (a similar technique is used at [8]).

Let  $I$  be the set of all image points,  $G_j$  the set of all points inside the  $j$  ground truth region,  $R_i$  the set of all points inside the  $i$  result region,  $g_j$  the entity of  $j$  ground truth,  $r_i$  the entity of  $i$  result,  $T(s)$  a function that counts the elements of set  $s$ . Table MatchScore( $i,j$ ) represents the matching results of the  $j$  ground truth region and the  $i$  result region. Based on a pixel based approach of [5], and using a global MatchScore table for all entities, we can define that:

$$\text{MatchScore}(i,j) = \alpha \frac{T(G_j \cap R_i \cap I)}{T((G_j \cup R_i) \cap I)}, \text{ where } \alpha = \begin{cases} 1, & \text{if } g_j = r_i \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

If  $N_i$  is the count of ground-truth elements belonging to entity  $i$ ,  $M_i$  is the count of result elements belonging to entity  $i$ , and  $w_1, w_2, w_3, w_4, w_5, w_6$  are pre-determined weights, we can calculate the detection rate and recognition accuracy for  $i$  entity as follows:

$$\text{DetectRate}_i = w_1 \frac{\text{one2one}_i}{N_i} + w_2 \frac{\text{g\_one2many}_i}{N_i} + w_3 \frac{\text{g\_many2one}_i}{N_i} \quad (2)$$

$$\text{RecognAccuracy}_i = w_4 \frac{\text{one2one}_i}{M_i} + w_5 \frac{\text{d\_one2many}_i}{M_i} + w_6 \frac{\text{d\_many2one}_i}{M_i} \quad (3)$$

where the entities one2one<sub>*i*</sub>, g\_one2many<sub>*i*</sub>, g\_many2one<sub>*i*</sub>, d\_one2many<sub>*i*</sub>, and d\_many2one<sub>*i*</sub>, are calculated from MatchScore table (1) following the steps of [5] for every entity  $i$ .

A performance metric for detecting each entity can be extracted if we combine the values of the entity's detection rate and recognition accuracy. We can define the following Entity Detection Metric (EDM<sub>*i*</sub>):

$$\text{EDM}_i = \frac{2\text{DetectRate}_i \text{RecognAccuracy}_i}{\text{DetectRate}_i + \text{RecognAccuracy}_i} \quad (4)$$

A global performance metric for detecting all entities can be extracted if we combine all values of detection rate and recognition accuracy. If  $I$  is the total number of entities and  $N_i$  is the count of ground-truth elements belonging to entity  $i$ , then by using the weighted average for all EDM<sub>*i*</sub> values we can define the following Segmentation Metric (SM):

$$\text{SM} = \frac{\sum_i N_i \text{EDM}_i}{\sum_i N_i} \quad (5)$$

#### 5 Participating methods

Brief descriptions of the methods whose results were submitted to the competition are given next. Each account has been provided by the method's authors and edited (summarised) by the competition organisers. The descriptions vary in length according to the level of detail in the source information provided.

##### 5.1 The BESUS method

This method—BESUS stands for Bengal Engineering and Science University, Shibpur (India)—was submitted by S.P. Chowdhury, S. Mandal and A.K. Das (of that university) in association with B. Chanda of the Indian Statistical Institute (ISI) in Calcutta. Similarly to the method submitted by the authors to the ICDAR2003 Page Segmentation Competition [1], this is a system constructed using a number of morphology-based modules.

In a pre-processing step that information is gathered and skew is corrected. Horizontal and vertical separators are extracted next by opening the bilevel image with a



horizontal or vertical (respectively) structuring element and connected component analysis [9]. Text is segmented based on the spatial relationship between pairs of textlines (identified based on the similarity and distribution of connected components) [10]. Graphics regions are extracted from a greyscale image (created from the original bilevel one) based on the analysis of a co-occurrence matrix in relation to the result of opening and closing operations on the whole image [11]. Line art regions (components) are identified based on topological features and a density ratio. Remaining regions are classified as noise.

### 5.2 The Océ method

This method was submitted by M. Bilderbeck, Z. Goey and R. Audenaerde of Océ Technologies B.V. in the Netherlands. It is a variant of the winning method of the ICDAR2003 Page Segmentation Competition [1]. Its working principles are as follows.

Connected components are identified in the image (after removing a 25-pixel wide border) and classified into small character, normal character, large character, photograph, graphic, vertical line, horizontal line or noise (in terms of the region types used in the competition, photographs are graphics, lines are separators and graphics are line-art) using a manually constructed decision tree based on features such as width, height, number of pixels etc. Using the result of this classification four images are split off:

- (a) an image containing photos and noise,
- (b) an image containing graphics,
- (c) an image containing lines, and
- (d) an image containing text.

In the last case, those blocks, in which the majority of connected components are classified as large characters are split off to a separate image. Thus, the image containing text is divided into two images:

- (d1) an image containing normal/small text, and
- (d2) an image containing headers.

Next, the components in the normal/small text image (d1), in the photo/noise image (a) and in the graphics image (b) are joined into blocks using a run length smearing procedure.

The resulting blocks are then classified by a voting algorithm that takes the connected component class statistics as its input. In the line image (c), each line is considered as a separate block with class label 'separator'. The blocks in the header image (d2) are identified by applying a connected component grouping algorithm, which also applies a post-classification step to assure that the blocks really contain text.

A boundary tracking algorithm [12] is used to trace the outer contours of all blocks (originally represented as

rectangles) in the smeared images and represent them as polygons. Finally, a cleaning step removes all polygons that are contained within others, (re)labels all very small polygons as noise and merges polygons that overlap to a certain extent.

### 5.3 The Tsinghua methods

Di Wen and Ming Chen, of Tsinghua University (State Key Laboratory of Intelligent Technology and Systems), in China submitted two different methods.

The first one (referred to as "Tsinghua method 1" here) is a bottom-up approach that works by progressively merging primitives at different levels (starting from connected components and resulting in text paragraphs etc.) based on the calculation of a quantitative measure (the Multi-Level Confidence - MLC value). This method has been reported in [13] and is adapted to English layouts for this competition. The output of this method is bounding rectangles only (a region may appear split as a result or bounding rectangles may overlap for different regions)

The second method ("Tsinghua method 2") is devised to deal better with irregular regions. It starts with the output of method 1 and text regions are separated from non-text ones. Text regions are identified as isothetic polygons based on a background analysis algorithm similar to [14] but working with connected components. Other types of regions are output as rectangles exactly as in method 1.

## 6 Results

We evaluated the performance of the 4 segmentation algorithms using equations (1)-(5) for all 26 test images with parameters  $w_1 = 1$ ,  $w_2 = 0.75$ ,  $w_3 = 0.75$ ,  $w_4 = 1$ ,  $w_5 = 0.75$  and  $w_6 = 0.75$ . All evaluation results for all entities are shown in Fig. 3 where the EDM<sub>i</sub> values averaged over all images are depicted. Fig. 4 presents the Segmentation Metric (SM) values for all segmentation algorithms averaged over all images. Fig. 4 shows that the second approach of Tsinghua has an overall advantage.

Concerning text region segmentation, the second approach of Tsinghua achieved the highest averaged EDM rate value (53.22%) while the first approach Tsinghua, the Océ method and the BESUS method achieved an averaged EDM rate value of 46.64%, 31.16% and 29.62% respectively. For graphics, the second approach of Tsinghua achieved the highest averaged EDM rate value (42.38%). For line-art and noise entities, the BESUS method achieved the highest averaged EDM rate values (80% and 20.24% respectively) while for

separator detection, the Océ method achieved the highest averaged EDM rate value (51,13%). The Tsinghua methods achieved zero EDM rate values for line-art, separator and noise entity segmentation.

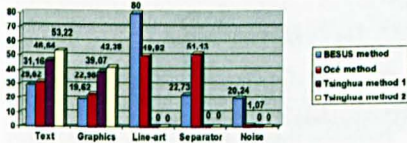


Figure 3. Evaluation results for all entities (EDM<sub>i</sub> values averaged over all images).

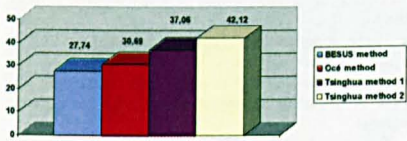


Figure 4. Averaged Segmentation Metric (SM) values.

7 Conclusions

The motivation of the ICDAR2005 Page Segmentation Competition was to evaluate existing approaches for page segmentation and region classification using a realistic dataset and an objective performance analysis system. The image dataset used comprised scanned technical articles and (mostly) magazine pages. The performance evaluation method used is based on counting the number of matches between the entities detected by the algorithm and the entities in the ground truth. The competition run in an off-line mode and evaluated the performance of four segmentation algorithms. The evaluation results show that the second Tsinghua method has an overall advantage (and gives better results for text and graphics). The Océ method is third overall with good consistency (and the best performance on separators). The BESUS method achieved the highest rates for line-art and noise entity segmentation.

References

[1] A. Antonopoulos, B. Gatos and D. Karatzas, "ICDAR2003 Page Segmentation Competition", *Proceedings of the 7th International Conference on Document Analysis and Recognition (ICDAR2003)*, Edinburgh, UK, August 2003, pp. 688-692.  
 [2] B. Gatos, S.L. Mantsaris and A. Antonopoulos, "First International Newspaper Contest", *Proceedings of the 6th*

*International Conference on Document Analysis and Recognition (ICDAR2001)*, Seattle, USA, September 2001, pp. 1190-1194.  
 [3] J. Kanai, S.V. Rice, T.A. Nariker and G. Nagy, "Automated Evaluation of OCR Zoning", *IEEE Transactions on Pattern Recognition and Machine Intelligence*, Vol. 17, No. 1, January, 1995, pp. 86-90.  
 [4] I.T. Philips, S. Chen and R.M. Haralick, "CD-ROM Document Database Standard", *Proceedings of 2nd International Conference on Document Analysis and Recognition (ICDAR '93)*, Tsukuba, Japan, 1993, pp. 478-483.  
 [5] I. Phillips and A. Chhabra, "Empirical Performance Evaluation of Graphics Recognition Systems," *IEEE Transaction of Pattern Analysis and Machine Intelligence*, Vol. 21, No. 9, pp. 849-870, September 1999.  
 [6] A. Chhabra and I. Phillips, "The Second International Graphics Recognition Contest - Raster to Vector Conversion: A Report," in *Graphics Recognition: Algorithms and Systems, Lecture Notes in Computer Science*, volume 1389, pp. 390-410, Springer, 1998.  
 [7] I. Phillips, J. Liang, A. Chhabra and R. Haralick, "A Performance Evaluation Protocol for Graphics Recognition Systems" in *Graphics Recognition: Algorithms and Systems, Lecture Notes in Computer Science*, volume 1389, pp. 372-389, Springer, 1998.  
 [8] B.A. Yanikoglu, and L Vincent, "Pink Panther: a complete environment for ground-truthing and benchmarking document page segmentation", *Pattern Recognition*, volume 31, number 9, pp. 1191-1204, 1994.  
 [9] S. Mandal, S.P. Chowdhuri, A.K. Das and B. Chanda, "Automated Detection and Segmentation of Form Document", *Proceedings of the 5th International Conference on Advances in Pattern Recognition (ICAPR2003)*, December 2003, Calcutta, India, pp. 284-288.  
 [10] A.K. Das and B. Chanda, "Segmentation of Text and Graphics in Document Image: A Morphological Approach", *Proceedings of the International Conference on Computational Linguistics, Speech and Document Processing (ICCLSDP '98)*, Calcutta, India, December 1998, pp. A50-A56.  
 [11] A.K. Das, S.P. Chowdhuri and B. Chanda, "A Complete System for Document Image Segmentation", *Proceedings of national Workshop on Computer Vision, Graphics and Image Processing (WVGIP2002)*, Madurai, India, February 2002, pp. 9-16.  
 [12] F. Chang, C.J. Chen, and C.J. Lu, "A Linear-Time Component-Labeling Algorithm Using Contour Tracing Technique", *Computer Vision and Image Understanding*, vol. 93, no. 2, 2004, pp. 206-220.  
 [13] M. Chen, X. Ding, et al. "Analysis, Understanding and Representation of Chinese newspaper with complex layout". *Proceedings of 7th IEEE International Conference on Image Processing*, 10-13 Sept. 2000, Vancouver, BC, Canada, IEEE.  
 [14] A. Antonopoulos, "Page Segmentation Using the Description of the Background" *Computer Vision and Image Understanding*, vol. 70, no. 3, 1998, pp. 350-369.

## **Appendix D**

### **Published paper on the ICDAR 2007 Competition**

The following pages contain a paper published in the proceedings of the International Conference on Document Analysis and Recognition describing the running and results of the ICDAR 2007 Page Segmentation Competition, based on the dataset described in appendix A.[5].

## ICDAR2007 Page Segmentation Competition

A. Antonacopoulos<sup>1</sup>, B. Gatos<sup>2</sup> and D. Bridson<sup>1</sup>

<sup>1</sup>Pattern Recognition and Image Analysis (PRIMA) Research Lab  
School of Computing, Science and Engineering, University of Salford, Manchester, M5 4WT, United Kingdom  
<http://www.primaresearch.org>

<sup>2</sup>Computational Intelligence Laboratory, Institute of Informatics and Telecommunications,  
National Center for Scientific Research "Demokritos", GR-153 10 Agia Paraskevi, Athens, Greece  
<http://www.iit.demokritos.gr/~bgat/>, [bgat@iit.demokritos.gr](mailto:bgat@iit.demokritos.gr)

### Abstract

*This paper continues the authors' attempt to address the need for objective comparative evaluation of layout analysis methods in realistic circumstances. It describes the Page Segmentation Competition (modus operandi, dataset and evaluation criteria) held in the context of ICDAR2007 and presents the results of the evaluation of three candidate methods. The main objective of the competition was to compare the performance of such methods using scanned documents from commonly-occurring publications. The results indicate that although methods continue to mature, there is still a considerable need to develop robust methods that deal with everyday documents.*

### 1 Introduction

Layout analysis methods—page segmentation in particular—continue to be reported in the literature on a frequent basis, despite this being one of the most researched sub-fields of Document Image Analysis. It is not difficult to see that the reason for this is that the problem is far from being solved. Successful methods have certainly been reported but, frequently, those are devised with a specific application in mind and are finetuned to the test image dataset used by their authors. The variety of documents encountered in real-life situations is far wider than the target applications of most methods.

There is no doubt that, for a given application or for a generic selection of real-life documents, it would be desirable to obtain an objective evaluation of the performance of different layout analysis methods. However, such a direct comparison between algorithms is not straightforward as it requires both the creation of suitable ground truth (a relatively laborious and precise task) as well as the definition of a set of objective evaluation criteria (and a method to analyse them).

This competition focuses on the evaluation of page segmentation and region classification subsystems. To the best of the authors' knowledge, this is only the third instance of an international generic layout analysis

competition (the previous two being the ICDAR2003 and ICDAR2005 Page Segmentation Competitions [1–2]). It should be mentioned that a relatively close previous instance, focusing on a specific application domain, was the First International Newspaper Page Segmentation Contest [3] held by the authors in the context of ICDAR2001. Prior to that, an evaluation of page segmentation (as part of OCR systems) was performed at UNLV [4], based on the results of OCR. That approach, however, cannot not be strictly considered to evaluate layout analysis methods since the OCR-based evaluation does not give sufficient information on the performance of page segmentation and region classification and is only applicable to regions of text (or text-only documents).

The motivation for this competition was the evaluation of page segmentation and region classification methods in realistic circumstances. By realistic it is meant that the participating methods are applied to scanned documents from a variety of sources, occurring in real life. This is in contrast to the majority of existing datasets and reports of method results using mostly structured documents (e.g., technical articles).

The competition is described next. In Section 3, an overview of the dataset and the ground-truthing process is given. The performance evaluation method and metrics are described in Section 4, while each of the participating methods is summarised in Section 5. Finally, the results of the competition are presented and the paper is concluded in Sections 6 and 7, respectively.

### 2 The competition

The objective of the competition was to evaluate layout analysis (page segmentation and region classification) methods using scanned documents from commonly-occurring publications. In addition to the comparative assessment, another objective was to obtain a broad look at the performance of different classes of methods (e.g., connected component analysis, morphological processing, analysis of background etc. as submitted for evaluation) in identifying different types of regions in a variety of documents.

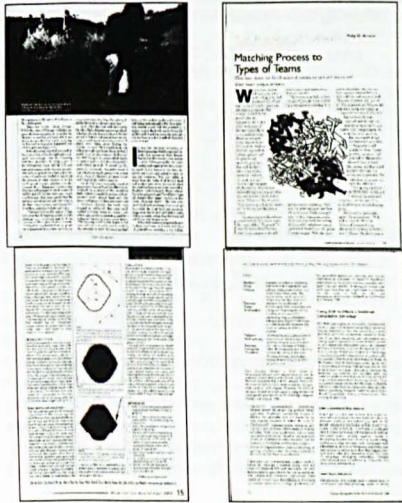


Figure 1. Sample page images from the training dataset.

The competition ran in an off-line mode. The authors of candidate methods registered their interest in the competition and downloaded the *training* dataset (document *images* and associated *ground truth*). One week before the competition closing date, registered authors of candidate methods were able to download the document *images* of the *evaluation* dataset. At the closing date, the organisers received the results of the candidate methods, submitted by their authors in a pre-defined format. The organisers then evaluated the submitted results.

It should be noted that the off-line mode is based on trust that the results submitted by the methods' authors are genuine. This trust is even more necessary if the evaluation system is publicly available. In this case, the evaluation system was not made available (only the principles were publicised) and above all, the organisers have faith in the authors' scientific integrity.

### 3 The dataset

It should be noted that there has been scarce availability of ground truth for the evaluation of methods analysing complex layouts (e.g., having non-rectangular regions). Such a dataset was created for the ICDAR2003 and ICDAR2005 competitions [1–2]. However, the current competition was based on a subset of a

significantly updated dataset. This dataset, which will shortly be released by the PRImA research lab, contains richer ground truth (in a correspondingly updated XML format) that provides a very wide range of information on region attributes (physical and logical).

Although the dataset contains instances of an exhaustive list of document types, the competition subset focuses (for meaningful evaluation purposes) on the most heavily used (in terms of information content and need to analyse) types of documents, such as magazine pages and technical articles.

It should be noted that, as the competition is on page segmentation, the images in the dataset have been processed to remove skew and other artefacts that would affect pre-processing and therefore implicitly also evaluate the pre-processing capabilities of the candidate methods.



Figure 2. Sample page image from the training dataset showing the superimposed description of region contours.

A balance had to be achieved between logistics (a manageable number of document images) and tractability for current methods. The decision was, therefore, made to focus on a cross section of 32 page images, comprising 47% technical articles (not necessarily with Manhattan layouts) and 53% magazine pages. It should be noted that also for reasons of tractability, the competition images were bi-level (in the general dataset the original images are in colour). A sample of page images given as part of the *training* dataset can be seen in Fig. 1.

The ground truth of each page image is an XML file (defined as part of the general dataset) that contains image and layout-specific information as well as the description

of the regions in terms of isothetic (having only horizontal and vertical edges) polygons. The ground truth for the competition was produced using a semi-automated tool developed by the authors. An XML viewer was developed for examining the images and the corresponding ground-truth XML, and was distributed to the competition participants. Another sample page image with the corresponding description of regions superimposed as isothetic polygons can be seen in Fig. 2.

The types of regions defined for the competition (simplified from the total number of different types in the general dataset) are: (i) *text*, (ii) *graphics*, (iii) *line art*, (iv) *separator*—graphical line segments between regions, and (v) *noise*.

#### 4 Performance evaluation

The performance evaluation method used is based on counting the number of matches between the entities detected by the algorithm and the entities in the ground truth [5–7]. We use a global MatchScore table for all entities whose values are calculated according to the intersection of the ON pixel sets of the result and the ground truth (a similar technique is used in [8]).

Let  $I$  be the set of all the ON image points,  $G_j$  the set of all points inside the  $j$  ground truth region,  $R_i$  the set of all points inside the  $i$  result region,  $g_j$  the entity of  $j$  ground truth,  $r_i$  the entity of  $i$  result,  $T(s)$  a function that counts the elements of set  $s$ . Table MatchScore( $i, j$ ) represents the matching results of the  $j$  ground truth region and the  $i$  result region. Based on a pixel-based approach [5], and using a global MatchScore table for all entities, we can define that:

$$\text{MatchScore}(i, j) = \alpha \frac{T(G_j \cap R_i \cap I)}{T((G_j \cup R_i) \cap I)}, \text{ where } \alpha = \begin{cases} 1, & \text{if } g_j = r_i \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

If  $N_i$  is the count of ground-truth elements belonging to entity  $i$ ,  $M_i$  is the count of result elements belonging to entity  $i$ , and  $w_1, w_2, w_3, w_4, w_5, w_6$  are pre-determined weights, we can calculate the detection rate and recognition accuracy for  $i$  entity as follows:

$$\text{DetectRate}_i = w_1 \frac{\text{one2one}_i}{N_i} + w_2 \frac{g\_one2many_i}{N_i} + w_3 \frac{g\_many2one_i}{N_i} \quad (2)$$

$$\text{RecognAccuracy}_i = w_4 \frac{\text{one2one}_i}{M_i} + w_5 \frac{d\_one2many_i}{M_i} + w_6 \frac{d\_many2one_i}{M_i} \quad (3)$$

where the entities  $\text{one2one}_i, g\_one2many_i, g\_many2one_i, d\_one2many_i$ , and  $d\_many2one_i$  are calculated from MatchScore table (1) following the steps of [5] for every entity  $i$ .

A performance metric for detecting each entity can be extracted if we combine the values of the entity's

detection rate and recognition accuracy. We can define the following Entity Detection Metric (EDM <sub>$i$</sub> ):

$$\text{EDM}_i = \frac{2\text{DetectRate}_i \text{RecognAccuracy}_i}{\text{DetectRate}_i + \text{RecognAccuracy}_i} \quad (4)$$

A global performance metric for detecting all entities can be extracted if we combine all values of detection rate and recognition accuracy. If  $I$  is the total number of entities and  $N_i$  is the count of ground-truth elements belonging to entity  $i$ , then by using the weighted average for all EDM <sub>$i$</sub>  values we can define the following Segmentation Metric (SM):

$$\text{SM} = \frac{\sum_i N_i \text{EDM}_i}{\sum_i N_i} \quad (5)$$

#### 5 Participating methods

Brief descriptions of the methods whose results were submitted to the competition are given next. Each account has been provided by the method's authors and edited (summarised) by the competition organisers.

##### 5.1 The Tsinghua methods

D. Wen and X. Ding, of Tsinghua University (State Key Laboratory of Intelligent Technology and Systems), in Beijing, China submitted two methods they developed as part of their effort to build a multi-language page segmentation method. Both methods are improved versions of the methods submitted to the ICDAR2005 competition [2].

Both methods are based on the same kernel, which is called the *Text Line Extraction (TLE)* module. The TLE is designed to solve the (common to both approaches) problem of extracting text lines in various types of document, whether magazines or newspapers, with regular or irregular layouts, English or Chinese (or any other language). It is a bottom-up aggregating method, which starts from connected components and merges them incrementally to obtain hierarchical layout structures. The first step of TLE is *Candidate Line Merging*, where connected components are merged according to their 4-direction Nearest Neighbour Connecting Strength [9]. Then in the second step, *Text Line Fitting*, candidate line segments are further merged into integrated text lines by comprehensive consultation of three factors: background separators, single line consistency and neighbouring lines consistency. That is, each pair of neighbouring candidate lines is merged when: 1) there is no background column separator between them; 2) the merged line has good consistency in

character sizes, alignments and spacing, 3) at least one of their common neighbouring lines in the vertical direction suggests them to be merged.

It is based on the results from TLE that different regions are formed. In this subsequent step, the first Tsinghua method (TH1) is different from the second (TH2) with respect to the region shape it supports. TH1 only supports rectangular regions. That is, each region is only represented by its bounding rectangle. For the non-rectangular (isothetic) textual regions, it tends to split them into several rectangular sub-regions. As for irregular graphics and image regions, it will output their bounding boxes only, even if they may overlap with other regions.

On the other hand, TH2 can support irregular regions. It takes the results from TH1 in terms of foreground information and uses a background analysis method to trace the contours of textual regions [10]. Neighbouring textual regions are glued and output as isothetic polygonal regions. However, for the graphics and image regions, the process is still inherited from TH1 so they are still output as bounding boxes

## 5.2 The BESUS method

This method—BESUS stands for Bengal Engineering and Science University, Shibpur (India)—was submitted by S.P. Chowdhury, S. Mandal and A.K. Das (of that university) in association with B. Chanda of the Indian Statistical Institute (ISI) in Calcutta. Similarly to the earlier versions of the method submitted by the authors to the ICDAR2003 and ICDAR2005 competitions [1–2], this is a system constructed using a number of morphology-based modules [11]. The segmentation procedure is applicable to both Manhattan and non-Manhattan layouts and it can detect text in any orientation.

The segmentation is carried out through the following phases:

1. **Pre-processing.** Skew correction is performed (not necessary in the competition dataset). The information zone is also found out of the whole document by omitting boundary noise.

2. **Graphics segmentation.** A pseudo-greyscale image is first created (the method works in greyscale whereas the test images were bi-level) using a low-pass adaptive filter based on the size of objects and on the frequency of their occurrence. Morphological open and close operations are then used to generate a unique feature known as OCF matrix [12] which is examined to estimate and remove the graphics regions from the image.

3. **Line art segmentation.** At this stage the page images contain mainly line art and text. The idea is to remove line art regions using the fact that they do not exhibit regular band structures as text lines do. An

extended mask region is computed on all components to form groups and the similarity of the components is examined. Line art regions exhibit different characteristics to text and are identified and removed from the image [13].

4. **Text segmentation.** Text mostly remains in the image at this point, exhibiting a regular structure of textlines and gaps between them. A vertical window of size  $2(\text{Text}_w + \text{Gap}_w)$  is created adaptively based on the statistical estimation of the height of the text band ( $\text{Text}_w$ ) and the line gap ( $\text{Gap}_w$ ) in between two text lines. Using this window a rough estimation of text lines is obtained. Further refinement is achieved through the use of additional features such as pen width [14].

## 6 Results

The performance of the 3 segmentation algorithms (BESUS, TH1 and TH2) was evaluated using equations (1)–(5) for all 32 test images with parameters  $w_1 = 1$ ,  $w_2 = 0.75$ ,  $w_3 = 0.75$ ,  $w_4 = 1$ ,  $w_5 = 0.75$  and  $w_6 = 0.75$ . These parameters are set to give maximum score to one-to-one matches and rather generous scores to other (partial) matches. Evaluation results for all types of entities are shown in Fig. 3 where the EDM, values averaged over all images are depicted (“noise” regions are omitted as their number was not significant enough). Fig. 4 presents the Segmentation Metric (SM) values for all segmentation algorithms averaged over all images. The BESUS method has a slight overall advantage over TH2 and TH1 with SM results of 55.75%, 55.46% and 51.75% respectively.

In more detail, concerning text region segmentation, the BESUS method achieved the highest averaged EDM, rate value (68.29%) while TH1 and TH2 achieved an averaged EDM, rate value of 53.82% and 58.56%, respectively. For graphics, TH1 achieved the highest averaged EDM, rate value (17.32%). For line-art entities, the BESUS method achieved the highest averaged EDM, rate value (14.52%) while for separator detection, TH1 and TH2 both achieved the highest averaged EDM, rate value (64.38%). Both Tsinghua methods achieved zero EDM, rate values for line-art segmentation.

## 7 Conclusions

The motivation for the ICDAR2007 Page Segmentation Competition was to evaluate existing approaches for page segmentation and region classification using a realistic dataset and an objective performance analysis system. The image dataset used comprised both scanned technical articles and (mostly) magazine pages. The performance evaluation method used is based on counting the number of matches between the entities detected by the algorithm

and the entities in the ground truth. The competition ran in an off-line mode and evaluated the performance of three segmentation algorithms. The evaluation results show that the BESUS method has an overall advantage (and gives better results for text and line-art). TH1 and TH2 performed better at segmenting separator regions, while the TH1 method performed best on graphics regions.

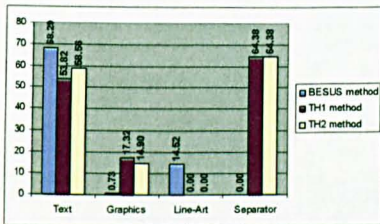


Figure 3. Evaluation results for all entities (EDM<sub>i</sub> values averaged over all images).

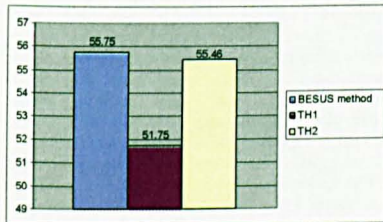


Figure 4. Averaged Segmentation Metric (SM) values.

**Acknowledgement**

The authors gratefully acknowledge the support of Google in creating the dataset used in this competition.

**References**

[1] A. Antonacopoulos, B. Gatos and D. Karatzas, "ICDAR2003 Page Segmentation Competition", *Proceedings of the 7<sup>th</sup> International Conference on Document Analysis and Recognition (ICDAR2003)*, Edinburgh, UK, August 2003, pp. 688-692.  
 [2] A. Antonacopoulos, B. Gatos and D. Bridson, "ICDAR2005 Page Segmentation Competition", *Proceedings of the 8<sup>th</sup> International Conference on Document Analysis and Recognition (ICDAR2005)*, Seoul, South Korea, August 2005, pp. 75-79.

[3] B. Gatos, S.L. Mantzaris and A. Antonacopoulos, "First International Newspaper Segmentation Contest", *Proceedings of the 6<sup>th</sup> International Conference on Document Analysis and Recognition (ICDAR2001)*, Seattle, USA, September 2001, pp. 1190-1194.  
 [4] J. Kanai, S.V. Rice, T.A. Nartker and G. Nagy, "Automated Evaluation of OCR Zoning", *IEEE Transactions on Pattern Recognition and Machine Intelligence*, Vol. 17, No. 1, January, 1995, pp. 86-90.  
 [5] I. Phillips and A. Chhabra, "Empirical Performance Evaluation of Graphics Recognition Systems," *IEEE Transaction of Pattern Analysis and Machine Intelligence*, Vol. 21, No. 9, pp. 849-870, September 1999.  
 [6] A. Chhabra and I. Phillips, "The Second International Graphics Recognition Contest - Raster to Vector Conversion: A Report," in *Graphics Recognition: Algorithms and Systems, Lecture Notes in Computer Science*, volume 1389, pp. 390-410, Springer, 1998.  
 [7] I. Phillips, J. Liang, A. Chhabra and R. Haralick, "A Performance Evaluation Protocol for Graphics Recognition Systems" in *Graphics Recognition: Algorithms and Systems, Lecture Notes in Computer Science*, volume 1389, pp. 372-389, Springer, 1998.  
 [8] B.A. Yanikoglu, and L. Vincent, "Pink Panther: a complete environment for ground-truthing and benchmarking document page segmentation", *Pattern Recognition*, volume 31, number 9, pp. 1191-1204, 1994.  
 [9] M. Chen, X. Ding, et al. "Analysis, Understanding and Representation of Chinese newspaper with complex layout". *Proceedings of 7<sup>th</sup> IEEE International Conference on Image Processing*, 10-13 Sept. 2000, Vancouver, BC, Canada, IEEE.  
 [10] A. Antonacopoulos, "Page Segmentation Using the Description of the Background" *Computer Vision and Image Understanding*, vol. 70, no. 3, 1998, pp. 350-369.  
 [11] A.K. Das and B. Chanda, "Segmentation of Text and Graphics in Document Image: A Morphological Approach", *Proceedings of the International Conference on Computational Linguistics, Speech and Document Processing (ICCLSDP'98)*, Calcutta, India, December 1998, pp. A50-A56.  
 [12] A. K. Das and B. Chanda, "Extraction of half-tones from document images: A morphological approach", *Proceedings of the International Conference on Advances in Computing*, Calicut, India, April 6-8, 1998, pp. 15-19.  
 [13] S. P. Chowdhury, S. Mandal, A. K. Das, and B. Chandha, "An efficient method for graphics segmentation from document images," *Proceedings of the 6<sup>th</sup> International Conference on Advances in Pattern Recognition*, Kolkata, India, Jan. 2-4, 2007, pp. 107-111.  
 [14] S. P. Chowdhury, S. Mandal, A. K. Das, and B. Chandha, "Segmentation of text and graphics from document images," *Proceedings of the 9<sup>th</sup> International Conference on Document Analysis and Recognition (ICDAR 2007)*, 23-26 Sept, 2007, Curitiba, Brazil.



# Appendix E

## Published paper on the evaluation method

The following pages contain a paper published in the proceedings of the International Conference on Document Analysis and Recognition describing the research presented in this thesis.[2].

## Performance Analysis Framework for Layout Analysis Methods

A. Antonacopoulos and D. Bridson

*PRIMA Lab, School of Computing, Science and Engineering, University of Salford,  
Greater Manchester, M5 4WT, United Kingdom  
<http://www.primaresearch.org>*

### Abstract

*This paper presents a new framework for in-depth analysis of the performance of layout analysis methods. Contrary to existing approaches aimed at evaluation or benchmarking, the proposed framework provides detailed information at various levels that can be used by method developers to identify specific problems and improve their work. Complex layouts are supported as well as the flexible configuration of goal-oriented performance analysis scenarios. The comparison of segmentation results against the ground truth is performed in a very efficient way based on a decomposition of any region shape into an interval-based description. The framework has been validated using the dataset and method results of the ICDAR2005 Page Segmentation Competition.*

### 1 Introduction

Layout Analysis is central to most Document Image Analysis systems and applications. It comprises Page Segmentation (identification of regions of interest), Region Classification (identification of the type of content of each region) and further processes such as Logical Labelling (labelling of regions in terms of their function) and reading order determination.

A considerable amount of effort has been devoted over the past two decades to develop various layout analysis methods (page segmentation, in particular) and new methods continue to be reported in the literature. Most methods were primarily aimed at specific applications and consequently were based on specific assumptions about their target document classes (e.g. text blocks are expected to be rectangular). Typically, each method was evaluated on relatively narrowly-focused application-specific datasets, which more often than not do not reflect the real-world occurrence of documents.

The need for objective and realistic *evaluation* of layout analysis methods is more pressing than ever, as evidenced by the various evaluation approaches proposed so far and the inception of ICDAR competitions in the area [1][2][3].

Past approaches have focused on calculating various error metrics in order to *quantify* the performance of page

segmentation methods, mostly for benchmarking or comparative evaluation. Early approaches [4] considered the recognised text inside each region and the corresponding number of edit operations necessary to correct errors. However, such a metric cannot give an accurate indication of page segmentation performance since a number of errors in the text are also due to OCR processes [5][6].

Later approaches focus on calculating discrepancies between ground truth and segmentation *region* characteristics. Such methods can be divided in two main categories: those that examine *geometric* correspondences of regions and those that perform *pixel* comparisons between regions. In almost all methods in the former category [6][7][8], regions (characters, textlines or paragraphs) are described by bounding boxes. Comparisons are efficient and corresponding ground truth straightforward to produce. However, a significant disadvantage is that documents with complex-shaped regions cannot be handled by such approaches although some early ideas of addressing this issue were explored [5][9].

Pixel-based region comparison approaches [10][1][2][3][11] on the other hand are very accurate and can work with complex-shaped regions. However, ground truth creation for such approaches can be more cumbersome [12] and it takes up a lot more storage. Furthermore, pixel-based comparison is much less efficient than geometric comparison.

In addition to the benchmarking goals of past approaches, there is also need for detailed *performance analysis* for each method. Such analysis extends beyond a set of simple scores for each method based on cumulative errors over a whole dataset. While evaluation and benchmarking are useful for a performance overview and direct comparison of methods they do not provide sufficient information for researchers and developers. For them, it is necessary to provide both a more detailed quantitative and a qualitative account of errors. As errors have different significance in different contexts, it is necessary to take this into account during evaluation so that developers may receive the in-depth information necessary to improve their methods.

The proposed framework is designed to provide in-depth information at various levels (dataset/page/region) to assist with method development in addition to goal-oriented performance evaluation and characterisation based on different user-defined scenarios. The correspondences between ground truth and segmentation regions are identified through geometric comparisons of regions represented as polygons achieving, thus, both accuracy in dealing with complex-shaped regions and efficiency (similar to bounding box comparison).

The framework is briefly described in the next section. An overview of ground truth requirements and related issues is given in Section 3. In Section 4, the performance analysis method is presented, with region representation, region correspondence determination and error qualification/quantification explained in separate subsections. The presentation of the analysis results is described in Section 5, while Section 6 discusses the proposed approach and concludes the paper.

## 2 Framework overview

The proposed performance analysis framework comprises two main components. First, a user interface through which batches of ground truth and segmentation results are selected, evaluation scenarios defined and interactive presentation of performance analysis results takes place.

Second, the performance analysis system itself which performs the following steps:

1. *Region representation*: Ground truth and segmentation regions are transformed into an interval-based representation.
2. *Region correspondence determination*: Using the interval-based representation, correspondence between parts of ground truth, segmentation and background regions is established.
3. *Error qualification and quantification*: Errors in correspondence between ground truth and segmentation regions are examined in the context of application scenario and their significance is established.

## 3 Ground truth

To take advantage of the full power of the framework there must be suitable ground truth with enough information about the regions and a sufficiently flexible description of the region outlines.

In developing the method, we have used the dataset which was also used for the ICDAR2005 Page Segmentation Competition [3]. Its ground truth contains rich information about the content and function of each region as well as about the corresponding page and

document [13]. Regions are described in terms of isothetic polygons (polygons having horizontal and vertical edges only).

## 4 Performance analysis

This is the most important framework component both in terms of technical issues and in terms of achieving the resulting information richness and accuracy.

The key challenge is the effective and efficient analysis and identification of correspondence of polygons instead of bounding boxes or pixel representations of regions.

Each of the steps in the process is described below.

### 4.1 Region representation

Region representation is key to both efficiency and accuracy of performance analysis. The proposed approach accepts both segmentation results and ground truth regions having practically any shape. However, it should be noted that, as printed regions on documents are mostly polygonal in shape with many of their edges being horizontal or vertical, it is naturally more efficient to represent them as isothetic polygons wherever possible.

Given a set of region contours (segmentation or ground truth), the first step is to create a representation of them in terms of *intervals*. An interval is defined as a maximal rectangle that can be fitted horizontally inside a region (starting at a given point on a vertical edge), spanning the whole width of the region [14]. This process can be thought of as a decomposition of a shape into a set of vertically adjacent horizontally-oriented rectangles. A simple decomposition of a region along these lines is illustrated in Fig 1(a).

The polygons of less complex regions will, more often than not, be decomposed into a set of taller intervals than more complex-shaped regions. In the representation of more complex shapes, certain intervals may be collapsed to horizontal lines. In the simple case of regions represented by bounding boxes (in Manhattan layouts, for instance) a single region will consist of a single interval.

Given a whole document page, the interval representation takes into account the existence of more than one region in the horizontal direction. Intervals are therefore fitted across regions as shown in the simplified (for clarity) example of Fig. 1(b).

For each document page in a dataset, the interval representation of the ground truth regions can be created in advance. The corresponding segmentation result regions are then also represented in a similar interval structure. The two interval structures are subsequently merged to form a *combined interval representation*. It is that representation which is used to determine the

correspondence between ground truth and segmentation regions. A simplified example of this representation is given in Fig. 2.

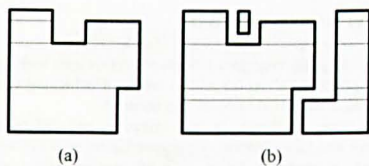


Figure 1. Interval representation of (a) a single region and (b) multiple regions.

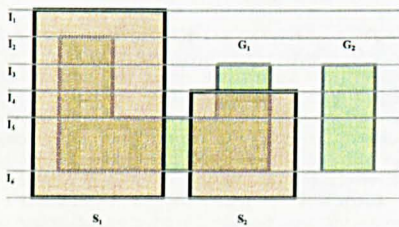


Figure 2. Combined (segmentation and ground truth) interval representation.

#### 4.2 Region correspondence determination

Within the combined interval representation, each interval line is examined in turn and overlaps are detected between:

- Segmentation interval and nothing (see interval in  $I_1$  line in Fig. 2)
- Segmentation interval and ground truth interval (see interval in  $I_2$  line in Fig. 2)
- Ground truth interval and nothing (see last two intervals in  $I_3$  line in Fig. 2)

Keeping track of the overlaps detected (as above) for all intervals of a given region it is straightforward to identify the following conditions for each region:

- A segmentation region that has no overlap with any ground truth region (wrongly detected region)
- A ground truth region that has been completely overlapped by a segmentation region (correctly detected region)

- A ground truth region that has been overlapped – completely or partially – by more than one segmentation region (split region)
- More than one ground truth region has been overlapped – completely or partially – by a single segmentation region (merged regions)
- A ground truth region that has not been completely overlapped by any number of segmentation regions (partially missed region)
- A ground truth region that has not been overlapped by any segmentation region (completely missed region)

The actual area of the overlap between individual intervals is calculated when overlaps are detected. Therefore, for each region the total area of overlap with other region(s) is recorded.

#### 4.3 Error qualification and quantification

The degree of success of a layout analysis method directly depends on the *type* as well as on the *quantity* of errors it makes. In terms of page segmentation, the five types of error (as listed above) have different significance depending on

- context (within the document)
- application scenario (user defined)

Error significance according to context is in most cases independent of the type of document. Examples include:

- A merger between two adjacent paragraphs within a single column of text is insignificant
- A merger between a paragraph of body text and a figure caption is a significant error
- A merger between two paragraphs across different columns is a significant error
- A merger between a text paragraph and a graphical region is a significant error

Error significance according to application scenario supplements the above, allowing a user to further tailor the performance analysis process. Examples of situations include:

- A merger between two graphical regions may not be significant in an OCR application.
- A merger between a section heading and a body text paragraph may not be significant in a general text processing application but may be significant if a table of contents needs to be constructed using section headings.

The significance of both context and application scenario is expressed by corresponding weights.

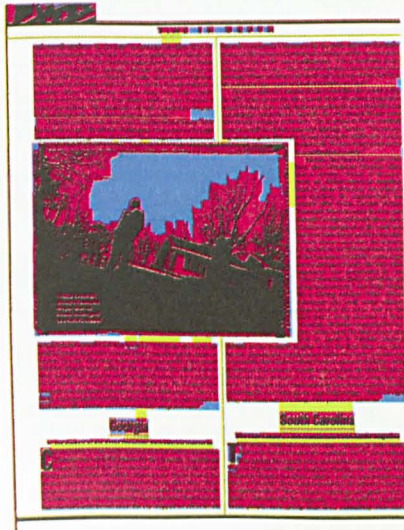


Figure 3. An example of visual presentation of results at the page level. Ground truth is in medium-dark (blue) colour while segmentation regions are in lighter (light green) colour. Overlapping regions are in darker (red) colour. Split and merged regions can be seen at a glance.

The proposed approach records each individual error, its context and the general application scenario. Based on this information, it also uses the information on the area of overlap between regions to assess and quantify the severity of the error.

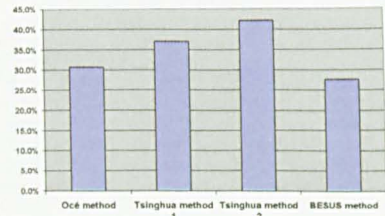
### 5 Presentation of analysis results

The above performance analysis gives rise to a considerable amount of information from overall task performance down to details of individual errors.

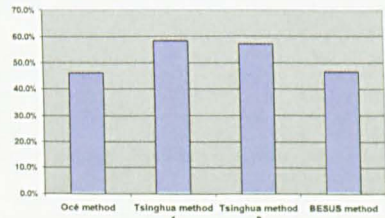
Information is available at dataset, page and region levels. Information is available by region type or error type.

A developer, for instance, can order results by error significance and individual errors can be displayed superimposed on the original page image (see Fig. 3).

A system integrator looking to choose between methods can specify a suitable scenario and a set of scores can be produced to provide a summary of the performance of each method for direct comparison.



(a)



(b)

Figure 4. (a) Results from the ICDAR2005 Page Segmentation Competition, and (b) from the proposed approach.

### 6 Discussion and conclusions

The new performance analysis method was compared against the published evaluation process of the ICDAR2005 Page Segmentation Competition [3]. Both the competition dataset and the results reported by the individual segmentation methods that took part were used in evaluating the system.

A graph of the overall competition results of the four different segmentation methods is shown in Fig 4(a). The corresponding graph using the proposed approach is shown in Fig 4(b).

Overall, the results broadly agree. Detailed results on different types of regions (not shown here) indicate that the main difference between the second and the third candidate methods (variant methods from the same research group) is due to slightly different weighting in application scenario (the ICDAR2005 seems to have been

heavily weighted towards the detection of text). The use of the proposed framework has provided detailed information in order to better understand this situation and to suggest a more balanced scenario for future competitions.

In addition to the page segmentation results discussed above, it is of course straightforward to also analyse the performance of region classification and logical layout analysis. As long as suitable information (region type and functional labels) exists in the ground truth it can be utilised. In fact, as evident from above, such information is necessary in order to take full advantage of the error qualification and quantification process of the framework.

Concluding, a new performance evaluation framework has been presented. Its novelty lies in two main directions. First, it provides considerably more in-depth information which is useful for developers (as opposed to evaluation or benchmarking only). It also enables goal-oriented performance analysis through a detailed error qualification and quantification scheme. Second, it is efficient and accurate using an interval-based region representation to establish correspondence between ground truth and segmentation regions. This representation closely approaches the efficiency of rectangular representation schemes but with the advantage that it supports the accurate handling of layouts with complex-shaped regions.

Further work continues towards building an on-line system (web service) which will enable researchers to use the framework as a web service.

## References

- [1] B. Gatos, S.L. Mantzaris and A. Antonacopoulos, "First International Newspaper Page Segmentation Competition", *Proceedings of the 6<sup>th</sup> International Conference on Document Analysis and Recognition (ICDAR2001)*, Seattle, USA, September 10-13, 2001, pp. 1190-1194.
- [2] A. Antonacopoulos, B. Gatos and D. Karatzas, "ICDAR2003 Page Segmentation Competition", *Proceedings of the 7<sup>th</sup> International Conference on Document Analysis and Recognition (ICDAR2003)*, Edinburgh, UK, August 3-6, 2003, pp. 688-692.
- [3] A. Antonacopoulos, B. Gatos and D. Bridson, "ICDAR2005 Page Segmentation Competition", *Proceedings of the 8<sup>th</sup> International Conference on Document Analysis and Recognition (ICDAR2005)*, Seoul, South Korea, August 29-September 1, 2005, pp. 75-79.
- [4] J. Kanai, S.V. Rice, T.A. Nartker and G. Nagy, "Automated evaluation of OCR zoning" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17 (1995), pp. 86-90.
- [5] A. Antonacopoulos and A. Brough, "Methodology for Flexible and Efficient Analysis of the Performance of Page Segmentation Algorithms", *Proceedings of the 5<sup>th</sup> International Conference on Document Analysis and Recognition (ICDAR1999)*, Bangalore, India, September 20-22, 1999, pp. 451-454.
- [6] M. Thulke, V. Märgner and A. Dengel, "A General Approach to Quality Evaluation of Document Segmentation Results", *Proceedings of the 3rd IAPR Workshop on Document Analysis Systems (DAS98)*, Nagano, Japan, November 4-6, 1998, Springer LNCS (1655), pp. 43-57.
- [7] S. Mao and T. Kanungo, "Software Architecture of PSET: A Page Segmentation Evaluation Toolkit" *International Journal of Document Analysis and Recognition*, Vol. 4 (2002), pp. 205-217.
- [8] A.K. Das, S.K. Saha and B. Chanda, "An empirical measure of the performance of a document image segmentation algorithm" *International Journal of Document Analysis and Recognition*, Vol. 4 (2002), pp. 183-190.
- [9] A. Antonacopoulos, F. Coenen, "Region Description and Comparative Analysis using a Tesseral Representation", *Proceedings of the 5<sup>th</sup> International Conference on Document Analysis and Recognition (ICDAR1999)*, Bangalore, India, September 20-22, 1999, pp. 193-196.
- [10] B. Yanikoglu and L. Vincent, "Pink panther: a complete environment for ground-truthing and benchmarking document page segmentation" *Pattern Recognition*, Vol. 31 (1998), pp. 1191-1204.
- [11] F. Shafait, D. Keyzers and T.M. Breuel, "Pixel-Accurate Representation and Evaluation of Page Segmentation in Document Images", *Proceedings of the 18<sup>th</sup> International Conference on Pattern Recognition (ICPR2006)*, Hong Kong, China, August 20-24, 2006, pp. 872-875.
- [12] J. Kanai, "Automated Performance Evaluation of Document Image-Analysis Systems: Issues and Practice" *International Journal of Imaging Systems and Technology*, Vol. 7 (1996), pp. 363-369.
- [13] A. Antonacopoulos, D. Karatzas and D. Bridson, "Ground Truth for Layout Analysis Performance Evaluation", *Proceedings of the 7<sup>th</sup> IAPR Workshop on Document Analysis Systems (DAS2006)*, Nelson, New Zealand, February 13-15, 2006, Springer LNCS (3872), pp. 302-311.
- [14] A. Antonacopoulos and R.T. Ritchings, "Representation and Classification of Complex-Shaped Printed Regions Using White Tiles", *Proceedings of the 3rd International Conference on Document Analysis and Recognition (ICDAR95)*, Montreal, Canada, August 14-15, 1995, pp. 1132-1135.