An Intelligent Document Understanding & Reproduction System

M.Sharpe, N.Ahmed and G.Sutcliffe
Department of Computer Science,
James Cook University of North Queensland,
Townsville, Australia.

Abstract

This paper presents the design and implementation of an intelligent document understanding and reproduction system. The system analyses a printed document to create an electronic document. The electronic document is written in a standard typesetting language (LaTeX), and is thus human usable. The output from processing the electronic document closely resembles the original printed document. A new document understanding technique, which utilises the Definite Clause Grammar formalism, has been developed.

1 Introduction

Documents in electronic form have a number of desirable features. They are easily modified, can be flexibly included in other documents, can be quickly searched for keywords, are inexpensive to store and can be easily distributed. Despite the growing popularity of electronic documents, the majority of documents are currently still in printed form. A system that can produce an electronic document from a printed document is obviously a useful tool. Previous approaches to this task have been able to produce good hard copy reproductions, but not useful electronic documents. This paper describes an intelligent document understanding and reproduction (the IDUR) system. Input to the IDUR system is the scanned image of a document. Output from the IDUR system is a LaTeX file, whose processed and printed form closely resembles the original document. LaTeX is one of the most popular document preparation packages, but our approach is general enough for most electronic document formats.

At first glance document analysis and reproduction appears to be predominantly an application of optical character recognition (OCR), the objective of which is to extract text characters from a document image. See [MOR92], [NAG92] and [KAH87] for a review of OCR research and techniques. However, OCR serves only as a subordinate task within a suite of cognitive functions [SCH92]. The functions determine firstly the spatial structure, and subsequently the logical structure, of the document. The spatial structure is extracted by applying image processing operations to the document image. The operations include identifying text, graphics and image blocks, performing OCR on text blocks, determining font types and sizes, establishing positional relationships between blocks, and determining the reading order of the document. This phase, therefore, extracts syntactic information from the document. On the other hand, extracting the logical structure captures the semantic information in the document. This phase of logical analysis includes identifying headings, paragraphs, captions, etc. The transformation of a spatial structure into a logical structure can be seen as document understanding and the reverse transformation as document reproduction. The reverse transformation may not be unique, because a logical structure may correspond to a variety of spatial structures [TSU90].

A considerable amount of research has been devoted to the subareas of spatial and logical analysis [GOR93], [WAN89], [NAG92], [BAI90] and [TSU90]. However, objectives of previous document analysis have been diverse, depending upon the application. Few researchers have tried to combine document analysis with document reproduction [BOK92]. Our effort in document analysis is driven by its reproduction capabilities, using the standard typesetting package LaTeX.

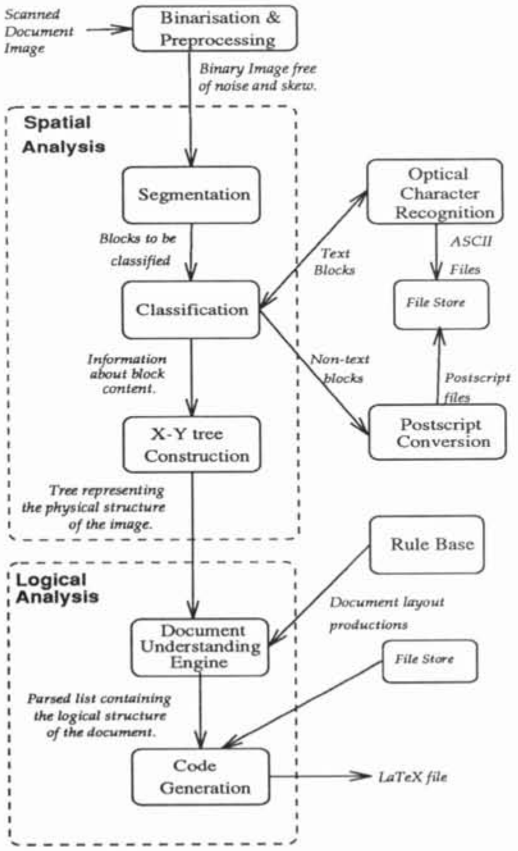

Scanned Document Image
Binarisation & Preprocessing
Binary Image free of noise and skew.
Spatial Analysis
Segmentation
Blocks to be classified
Classification
Information about block content.
X-Y tree Construction
Tree representing the physical structure of the image.
Optical Character Recognition
Text Blocks
ASCII Files
File Store
Non-text blocks
Postscript files
Postscript Conversion
Rule Base
Logical Analysis
Document layout productions
Document Understanding Engine
File Store
Parsed list containing the logical structure of the document.
Code Generation
LaTeX file

Figure 1: *System design*

The IDUR system consists of five stages: • Binarisation and Preprocessing, • Page Layout Analysis, • Optical Character Recognition, • Document Understanding, and • Document Reproduction. The first three stages perform the spatial analysis. Existing techniques have been adapted in these stages. For the final two stages, which perform the logical analysis and the code generation, original techniques have been developed. Figure 1 shows the relationship between these stages.

The rest of this paper is organised as follows. Section 2 describes image preprocessing required before document analysis can commence. Section 3 describes the spatial analysis component of the document analysis, and Section 5 deals with the logical analysis. Section 4 explains how optical character recognition interacts with the spatial and logical analysis. Concluding remarks are given in section 6.

2 Binarisation and Preprocessing

Binarisation is the process by which the image from a scanner is converted to a binary bitmap image. The scanner image will most likely contain noise and skew. Preprocessing corrects these problems.

Noise in an image is caused by corresponding noise in the printed document, and by defects in the scanning hardware. There are many types of noise, the most harmful of which is salt-and-pepper noise. The the IDUR system uses the kFill filter [STO92] to remove salt-and-pepper, and other, noise.

Skew occurs in images that have been scanned such that the scan axis is not perpendicular to the dominant text baseline. Skew correction involves first determining the skew angle and then rotating the image in the opposite direction. The Hough Transformation method described in Hind et al. [HIN90] is used for determining the skew angle. Care must be taken when rotating bitmaps, as many methods deform the contents of an image. This can be particularly harmful for optical character recognition. In the IDUR system the pseudo-rotation is used [BAI90].

3 Spatial Analysis

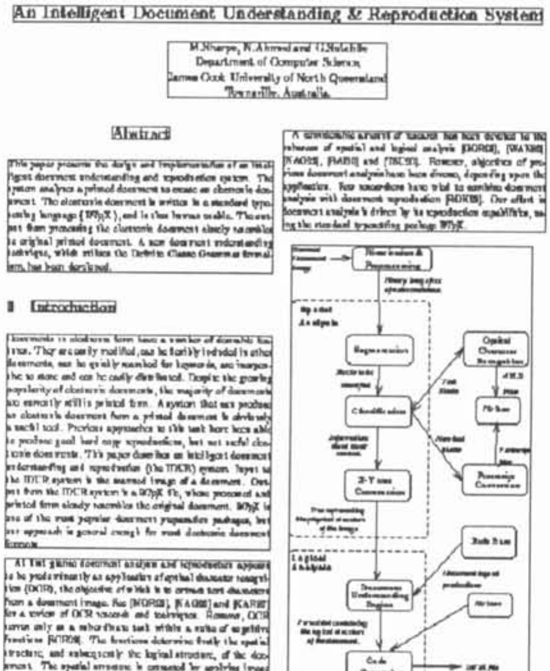
3.1 Page Layout Analysis

The page layout analysis section of the design has three components, namely (i) Segmentation, (ii) Classification and (iii) X-Y Tree Construction.

3.1.1 Segmentation

Each page of the document image is segmented into blocks. The Run Length Smoothing Algorithm (RLSA)[CAS82] and the Projection Profile Method (PPM)[BAI90] are used for this purpose. The aim of segmentation is to extract blocks that contain only one logical structure. Examples of logical structures are headings, paragraphs, captions, and pictures. Refinements may be made to the segmentation during the logical analysis phase of the IDUR process (when additional information is known), to further segment or to join blocks that do not contain one logical structure.

Efficient one pass routines that implement the RLSA and the PPM have been developed. It is observed that the RLSA is sensitive to small amounts of noise (even after preprocessing some noise may remain). The PPM on the other hand is less sensitive to noise, but sometimes treats some non-noise components of the document image as noise and produces poor segmentation. A combination of RLSA and PPM has been used for segmentation, thus overcoming the weaknesses of each method. Figure 2 is the segmentation that the IDUR system obtains when applied to the first page of this paper.



An Intelligent Document Understanding & Reproduction System

M.Sharpe, N.Ahmed and G.Sutcliffe
Department of Computer Science,
James Cook University of North Queensland
Townsville, Australia.

Abstract

1 Introduction

Figure 2: *Segmentation of the first page of this document.*

3.1.2 Classification

After segmentation, each block is classified according to its contents. A modification of the classification scheme presented in [WAN89] and [CAS82] is used to determine whether each block contains text, pictures or other information (table, equations, etc). Their method assumes that blocks that contain text only contain one line of text. We have relaxed this assumption to allow blocks containing many lines of text.

The classification of blocks is based on image features. Measures such as pixels counts, transition counts, vertical and horizontal cuts, text characteristics, and pixel run-lengths, are used to compute the following image features:

- Block eccentricity :- $\Delta x/\Delta y$, where Δx and Δy are the block dimensions.
- Transition density :- is the number of transitions expressed as a percentage of the maximum number of transitions. Transition densities are measured in the horizontal and vertical directions.
- Black pixel density:- is the number of black pixels expressed as a percentage of the total number of pixels in a block. A corrected black pixel density is also calculated; in this calculation the total number of pixels is reduced by an amount proportional to the number of blank rows in the block.

- Run length density :- is the average black pixel run length expressed as a percentage of the maximum possible run length. Run length densities are measured in the horizontal and vertical directions.
- Cut count :- is the number of all white lines through a block. Cut count is measured in the horizontal and vertical directions.

The size and position of each block is also known from the segmentation.

The image features are used to classify blocks containing either text, pictures, line drawings and horizontal and vertical lines as follows,

- Text Blocks:- are characterised by high transition density, low average run length densities, and many horizontal cuts. A line of text can be identified by many vertical cuts. Text blocks tend also to have a low to medium black pixel density.
- Picture Blocks :- are characterised by high average run-length densities, no horizontal or vertical cuts, and low transition density.
- Line-drawing Blocks :- are characterised by high white run-lengths, low black run-lengths, and a very low black pixel density.
- Horizontal and Vertical lines :- are characterised by extreme eccentricity values; very low for vertical lines and very high for horizontal lines.

After the classification of blocks, blocks containing text are passed to OCR. All other blocks are converted to postscript form, to be included into the electronic document at a later stage. It is of course possible to process non-text blocks more thoroughly in order to extract more information, such as tables and equations.

3.1.3 X-Y Tree Construction and Transformations

At this stage information about each block is recorded in the node of an X-Y tree [NAG84], one X-Y tree per document page. Each X-Y tree node contains the position of the block in the document image, the block's classification, and other information which is dependent on the block's classification. For example, for a text block, the number of lines of text and the dominant font size. The name of the file that contains the contents of the block is also stored in each node. If the node corresponds to a text block the file contains the text extracted by OCR, otherwise it contains the postscript form the the block.

The X-Y tree is an intermediate representation of the document, and represents the syntax of the document. At this point all the necessary image processing is complete.

The X-Y tree corresponding to the first page of this document is displayed in Figure 3. The section number and the 'Introduction' sub-title have been separated into columns.

After page layout analysis the spatial structure of each page is known. The X-Y trees corresponding to each page are merged to build the spatial structure of the entire document. Document understanding is then used to determine the logical structure of the document as a whole.

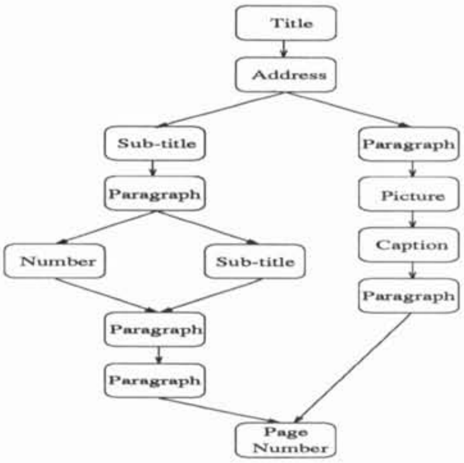


Figure 3: *X-Y tree corresponding to the first page of this document.*

4 Optical Character Recognition

The OCR in the IDUR system provides useful data that is used by the rest of the system. The extraction of text is the most important task of the OCR unit. The text extracted from a block is placed in an ASCII file, and the filename is returned to the IDUR system. Other information extracted during OCR processing includes :

- Font information :- is information about the fonts and font sizes used in the text block. This information is necessary not only for reproducing the document, but is also useful for identifying logical components within the document. For example, titles are often larger than the rest of the text in a document, and are often printed in bold font or are underlined. Words at the start of a text block may also be printed in a different font for highlighting purposes.
- Keywords in key-positions :- often aid the identification of a block. The keyword 'Abstract' in the abstract of this document is an example of this. The keyword 'Reference' is useful for identifying the reference list in a journal article.
- Data type information:- may also aid block identification. It is useful to know if the primary data type is alphabetic, numeric or symbolic, or if the first few characters are of this type. Subtitles often start with a number, and list elements often start with a numeric or symbolic character (e.g., this list starts with bullets).

5 Logical Analysis

5.1 Document Understanding

Document understanding is the process whereby the semantics of a document is extracted. We have developed a document understanding engine (DUE) to perform this task in the IDUR system. The DUE is implemented in Prolog. Knowledge about the basic structure of a variety of documents is encoded in a rule base. Knowledge in the rule base is used to effectively parse the X-Y tree, while analysing and identifying the logical components of the document.

Input to the DUE is the X-Y tree. Output from the DUE is LaTeX typesetting codes, which are written to an output file. The rule base is maintained in the form of a Definite Clause Grammar (DCG) [PER80]. Since the DCG formalism requires a list as input, the X-Y tree is transformed to produce a X-Y list. The transformation corresponds to those presented in [TSU90]. We have chosen to transform the tree assuming a left to right, top to bottom reading order (however, other reading orders could be used). The DUE parses the X-Y list to determine the logical structure of the document. The X-Y list corresponding to the X-Y tree in Figure 3 is shown in Figure 4.

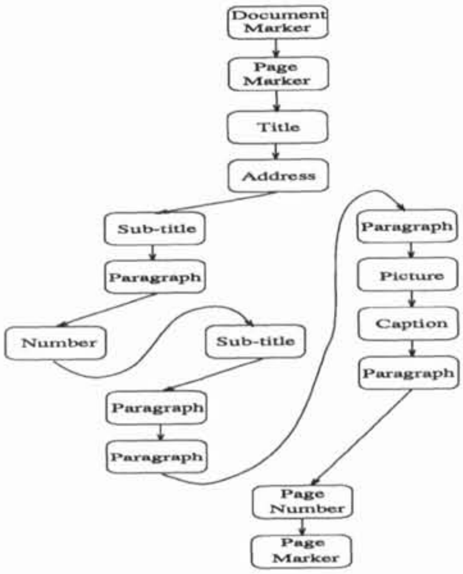


Figure 4: *X-Y list corresponding to the X-Y tree in Figure 3*

Some extra **marker** nodes are inserted into the X-Y list representation during the transformation process. These **marker** nodes provide the DUE with global and local information which is available at different points during the parse.

An example of a part of the DCG grammar used for technical journals follows (it is somewhat simplified for illustration purposes).

```
document            --> [document_marker(start)],
                        journal_paper,
                        [document_marker(end)].

journal_paper       --> journal_pages.

journal_pages       --> journal_page,
                        journal_pages.

% Journal article with an abstract on first page
journal_page        --> [page_marker(Page_Info)],
                        title,
                        optional(address),
                        optional(date),
                        abstract,
                        journal_components.

% Detect end of a page.
journal_components --> [page_marker(end)],!.

% Typical journal components
journal_components --> (picture;paragraph;title;
                        sub_title;list),
                       journal_components.

% Associate a section number with a subtitle.
% This fixes the segmentation problem
sub_title           --> section_number,
                        section_name.

% subtitle without a section number.
subtitle            --> section_name.

abstract            --> subtitle,
                        paragraphs.

paragraphs          --> paragraph,
                        paragraphs.
paragraphs          --> [].

title                --> [X],
                         {is_title(X)}.

% Check for centred title.
is_title(block(Position,OCR_Info,_)):-
      centred(Position),
      is_bold(OCR_Info).
```

The grammar used here is quite intuitive. Errors in the segmentation can be are tolerated by appropriate coding of the grammar, as is done with the **sub_title** predicate. The **document_markers** and **page_marker** are examples of the extra nodes inserted into the X-Y list. The **document_marker** is used to inform the DUE about global information regarding the document being parsed, and also information about how the document should be reproduced. The **page_marker** is used to inform the DUE about information global to a page (for example, information about the average font-size & type, page margins etc.). It can also be used to make local changes to the reproduction.

5.2 Code Generation

As the document is being parsed by the DUE, LaTeX typesetting codes are generated (this is not shown in the DCG code above). These codes form the electronic version of the document.

6 Conclusion

An intelligent document understanding and reproduction system has been described. To be able to produce a useful electronic document it is necessary to analyse the physical structure of the document (spatial analysis), and to identify the logical components of the document within logical structure of the document as a whole (logical analysis). Standard image processing techniques have been successfully employed for the spatial analysis. The IDUR system uses the DCG formalism to encode the grammar of documents, and uses this grammar as the basis for the logical analysis. This is an original technique for this task.

References

[BAI90] Baird, H.S., Thompson, K., Reading chess, IEEE Tr. on Pattern Analysis and Machine Intelligence, 12(6), pp. 552-559, 1990.

[BOK92] Boker, M., Omnidocument technologies, Proc. of IEEE, 80(7), pp. 1067-1078.

[CAS82] Casey, R.G., Wong, K.Y., Wahl, F.M., Document analysis systems, IBM Journal of Research and Development, 26(6), pp. 647-656, Nov., 1982.

[GOR93] O'Gorman, L., The document spectrum for page layout analysis, IEEE Tr. on Pattern Analysis and Machine Intelligence, 15(11), pp. 1162-1173, 1993.

[HIN90] Hinds, S.C., Fisher, J.L., D'Amato, D.P., A document skew detection method using run-length encoding and the Hough transform, Proc. 10th Intl. Conf. on Pattern Recognition (ICPR), Atlantic City (NJ), pp. 464-468, June, 1990.

[KAH87] Kahan, S., Pavlidis, T., Baird, H.S., On the recognition of printed characters of any font and size, IEEE Tr. on Pattern Analysis and Machine Intelligence, 9(2), pp. 274-288, 1987.

[MOR92] Mori, S., Suen, C.Y., Yamamoto, K., Historical review of OCR research, Proc.of IEEE, 80(7), pp. 1029-1057, 1992.

[NAG84] Nagy, G., Seth, S., Hierarchical representation of optically scanned documents, 7th Intl. Conf. on Pattern Recognition, pp. 347-349, 1984.

[NAG92] Nagy, G., At the frontiers of OCR, Proc. of IEEE, 80(7), pp. 1093-1100, 1992.

[PER80] Pereira, F.C.N., Warren, D.H.D., Definite clause grammars for language analysis - a survey of the formalism and a comparison with augmented transition networks, Artificial Intelligence, 13, pp. 231-278, 1980.

[SCH92] Schurmann, J., Bartneck, N., Bayer, T., Franke, J., Mandler, E., Oberlander, M., Document Analysis - from pixel to contents, Proc. of IEEE, 80(7), pp. 1101-1119, 1992.

[STO92] Story, G.A., O'Gorman, L., Fox, D., Schaper, L.L., Jagadish, H.V., The RightPages image-based electronic library for alerting and browsing, IEEE Computer, 25(9), pp. 17-26, 1992.

[TSU90] Tsujimoto, S., Asada, H, Understanding multi-articled documents, Proc. 10th Intl. Conf. on Pattern Recognition (ICPR), Atlantic City (NJ), pp. 551-563, June, 1990.

[WAN89] Wang, D., Srihari, S.N., Classification of newspaper block using texture analysis, Computer Vision Graphics and Image Processing, 47, pp. 327-352, 1989.