

***Image Based Typographic
Analysis of Documents***

David S. Doermann
Richard Furuta

August 1993
TAMU-HRL 93-008

Image Based Typographic Analysis of Documents

David S. Doermann
Richard Furuta

Hypermedia Research Lab

AUGUST 1993
TAMU-HRL 93-008

Image Based Typographic Analysis of Documents*

David S. Doermann
Document Processing Group
Center for Automation Research
University of Maryland
College Park, MD 20742

Richard Furuta
Hypermedia Research Laboratory
Department of Computer Science
Texas A&M University
College Station, TX 77843

Abstract

In this paper we provide an approach to image based typographic analysis of documents. The problem requires a spatial understanding of the document layout as well as knowledge of the proper syntax. Our system performs a page synthesis from the stream of formatting commands defined in a DVI file. Since the two-dimensional relationships between document components is not explicit in the page language, we develop a representation which preserves the two-dimensional layout, the read-order and the attributes of document components. From this hierarchical representation of the page layout we extract and analyze relevant typographic features such as margins, line and character spacing, and figure placement.

1 Introduction

The task of typographical analysis is one that is addressed on a regular basis by editors, typographers and compositors who attempt to isolate instances of writing, typesetting and printing errors in documents. Some common typographic errors include overlapping and touching characters, inconsistent margins, abrupt changes in typeface, and widows and orphans. Such errors detract aesthetically from the document. Although in many cases these errors¹ are subjective, they can often be quantified in a meaningful way [2, 3]. This paper addresses the automatic detection of errors by typographic analysis of standard device independent typesetting representation (e.g. DVI or PostScript).

In Section 2, a subset of the tasks that may be performed by a typography checker are presented. Section 3 describes the rendering process and provides

a representation for the rendered document components. Section 4 outlines the detection algorithms for typographic features and Section 5 provides a proposed approach for interactive correction of typographic errors.

2 Typographic Features

Typographic features can be weakly classified into two categories: *syntactic* and *stylistic*. Syntactic features (e.g. spelling, punctuation, capitalization and abbreviations) are typically associated with the requirements of the language, while stylistic features (e.g. font selection, line and word spacing, margins, indentations, and the printing resolution) deal with the perception of the layout and resulting quality of the document. This paper concentrates primarily on the stylistic aspects of typographical analysis.

Each type of error identified is classified according to its relevance to a pixel/stroke, character, word, line, paragraph or figure, page or document. Some examples include:

Pixel/stroke

- component attributes or styles that are inconsistent with printer capabilities.
- incompatibilities between printing language and hardware such as attempting to print a color document on a B/W printer or attempting to render halftones on a printer with insufficient resolution.

Character/word

- too little, too much or inconsistent spacing resulting in touching characters or abnormally large gaps
- user/printer-supplied font discrepancies.

Line

- inconsistent spacing
- line overlap or touching vertical characters
- skewed text and baseline

*To appear in the International Conference on Document Analysis and Recognition 1993

¹The term *error* is used to refer to a document feature or attribute which strays from a (possibly weakly constrained) set of rules about how an ideal document should appear.

- hyphenation that leaves fewer than two characters at the end of a line, or moves fewer than three characters to the next line
- equations that do not line up vertically.

Paragraph

- inconsistent indentations
- ridges/valleys
- multiple hyphenated lines in sequence
- last line that consists of a word with less than four characters or which is hyphenated.

Figure

- graphics touching characters
- graphics not centered/justified in allocated space
- label or caption typefaces that are incompatible with surrounding text.

Page

- inconsistent margins
- inconsistent page number positioning
- two-column page with lines that do not align
- widows/orphans

Document

- global inconsistencies in page and low-level styles

A further classification is appropriate into spatial and non-spatial classes of typographic features. A majority of the features associated with the logical document organization are defined with respect to spatial relationships between components and include margins, line, word and character spacing. For example, consider a figure which is produced with a drawing package and imported into a document. If centering is desired, the parameters may need to be computed manually in the absence of high level formatting capabilities. Errors in formatting are common and must be checked typographically.

Non-spatial pixel level features, as well as other component attributes such as width, font or style, focus primarily on the perceptual effects of the rendered components and on language/hardware compatibility issues.

3 Document Rendering and Representation

Our approach to typographic analysis centers on the ability to (1) analyze the rendered document with respect to the “ideal” or intended document; and (2) provide a representation for the rendered image that can be used to analyze the spatial aspects of the problem.

For typographical features, it is necessary to consider the two-dimensional spatial relationships between document entities, in addition to linguistic and graphical syntax which can be examined sequentially in the text. Unfortunately, typesetting languages typically do not represent these two-dimensional relationships explicitly, and may therefore be able to typeset the same document in many ways. In order to address the two dimensional aspects of this typography problem, it is necessary to perform analysis on the rendered bitmap, with appropriate links to the formatted document.

Non-spatial features are derived from each component (e.g. the bitmap) or by examining the compatibility of document attributes with hardware specifications. No special representation is necessary other than the bitmap and the trace of the current “context” in which the components are being rendered.

Spatial features require knowledge of both the logical decomposition of the document into components such as words, lines, paragraphs, figures and columns as well as the position of the component on the page. Since the typesetting language does not explicitly represent these logical components, the system must derive the page decomposition while rendering the document.

Three types of components are of primary interest: characters or symbols, graphics and halftones. The general approach is to group characters and symbols up to the word/line level, then place them in a spatial data structure along with appropriate attributes. Graphic and image components are rendered and placed directly into a spatial data structure. The inference of features, discussed in Section 4, uses the spatial representation.

3.1 Representation

The core of the representation is a bitmap that is kept for each component rendered by the system. A higher level representation is required that allows the inference of the page decomposition from the sequentially rendered image, and must be able to preserve the traditional logical document components (e.g. characters, words, lines, paragraphs) in addition to the two-dimensional spatial relationships (layout structure). Each of the three types of components, characters/symbols (text), graphics and halftones are treated independently.

Text tends to fall into horizontal lines arranged within a higher level structure such as paragraphs or columns. To facilitate organization, characters and symbols are placed into a linked mesh before being

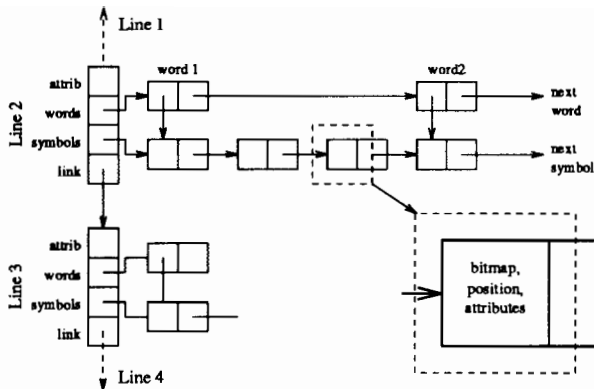


Figure 1: Linked Mesh

grouped into more complex components (Figure 1). When a character is rendered, the centroid of a bounding box is computed. The coordinates of this point are used to place the bitmap into a mesh, first by row (line), then by column (position on line). The height of a line of text is defined by the first character in the line. By knowing the character, we can hypothesize the zone covered and estimate the height of the line. The centroids of subsequent characters must fall within the bounds defined by the height of the bounding rectangle of the first character.

If a line of text exists which is consistent with the new character, it is placed with the existing row; otherwise, a new row is started. Symbols which fall less than δ (a factor of the maximum expected word spacing) away from the existing word or phrase in the horizontal direction are grouped as belonging to the same line. Otherwise, a new line is started and placed at the end of the current line in the list. The word-level organization is refined after all characters are rendered, with preference to characters which are rendered consecutively. Blocks of text can be placed into the spatial data structure by their bounding boxes.

As a graphic or image component is rendered, its bounding box is put as a rectangle into a quadtree, with a bitmap image and appropriate attributes attached. Similarly, components of unknown class are placed with their bounding boxes in the quadtree. As they are placed, their intersection with the current document is explored. If the two rendered components intersect, the region of intersection is noted.

3.2 Spatial Data Structures

The “read-order” of the document text is preserved in the linked list structures described above. In order to facilitate the efficient analysis of the interaction be-

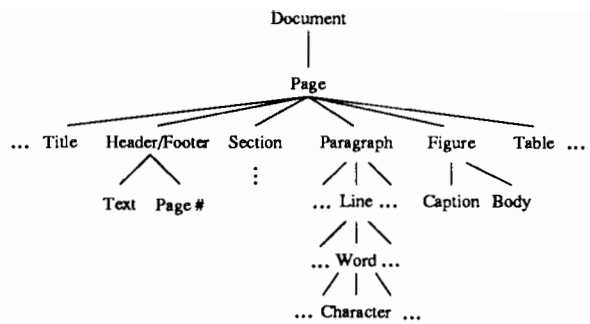


Figure 2: Typical decomposition hierarchy

tween these symbols, we store each rectangular “zone” in a quadtree. A zone may correspond to various components of the physical structure such as a line, paragraph or column depending on the type of document and the typographic features analyzed.

For an overview of algorithms for representing and operating on collections of rectangles, see Chapter 3 of Samet’s *The Design and Analysis of Spatial Data Structures* [9].

3.3 Document Models

A partial document decomposition hierarchy is shown in Figure 2. It is clear that there is a continuum of ways in which such models can be used to test typographic correctness. Models can be designed to represent a range of details from *fixed* formatting, where exact margins, spacing and font style are used, to *generic* formatting where only the consistency and appropriateness of margins, spacing, and font are considered. Fixed format models may be required for checking specific thesis, journal, or proposal submissions, while generic format models may be used for more general documents.

Recall that a page synthesis process must occur in order to derive meaningful components to determine “class” (memo, letter, form, one/two column, ...) of a document. An increasingly common form of language used to identify elements in a document being prepared for formatting (the *document markup*) is called “generic coding.” Examples include the languages defined by the ISO SGML document markup standard [7], the ISO ODA document interchange standard [6], the commercially-available Interleaf product, the commercially-available Microsoft Word product, the commonly-used L^AT_EX set of T_EX macros [8] as well as many others².

²See [1], [5], and [4] for more detailed discussions of the underlying differences between the models used by these systems.

In the current system, we limit our analysis to simple documents consisting of headers, regions of text, tables and figures and specific typographic features associated with standard formatting attributes such as character, word and line spacing, font size, margins, etc.

4 Inference of Typographic Features

Recall that typographic analysis is performed on the bitmap components of the rendered document. The components can be generated by a `DVIBitmap` program which interprets a DVI file and produces an attributed bounding box representation of each character as it is rendered³. The attributes of each include position, font, size and the ASCII representation of the character. The analysis is performed on the output of the program.

4.1 Algorithms

There are four classes of algorithms which play fundamentally different roles in this system.

Quadtree queries are used to locate margins, check intersections, verify centering/justification, and in general provide access to hierarchical document structure.

Spatial Transformations are used to identify patterns in collections of individual components. One example includes the identification of valleys of white space and ridges of the same character or word aligning vertically on the page.

Search Algorithms are used to locate typographic errors which can be identified sequentially in the text, such as widows and orphans, errors in hyphenation, and spacing inconsistencies. The algorithms are applied to all relevant zones in the quadtree.

Detailed Analysis algorithms are provided for measuring pixel level properties such as stroke width and intersection of characters. For example, when detecting the existence of touching text/graphics, overlapping lines and touching characters, a hierarchy of tests are performed at the zone level. If the bounding boxes intersect, a more detailed analysis is performed, and if necessary, the bitmaps are examined for common pixels.

³A `PSBitmap` option is being explored because of the emergence of PostScript as a standard mixed text and graphics language.

4.2 Example Analysis

The output of the system consists of a graphic and a textual description of the resulting ambiguity. The description consists of the point or region of the image where the error occurs, the type of error and a possible solution. An approach to the interactive correction of errors is outlined in Section 5. More advanced techniques will allow the errors to be corrected in the document rendering language (e.g. PS or DVI) as well.

Figure 3a shows a portion of a document that contains several typographic errors and Figure 3c shows page synthesis up to the word level. The algorithm groups words into lines, and lines into paragraphs. A two column interpretation of the text is made and the margins are approximated. Figure 4a shows fine zones which were detected as anomalous. The row numbers are arranged from bottom to top. Rows 5 and 7 (the top two shown) in the second column have words which have abnormally large word spacing. Row 8 has two lines that extend into the margin, and row 13 has a word that extends into the margin, and whose baseline is inconsistent with the other words in that row.

Although the errors shown in this example are clear, other errors such as isolated pairs of touching characters, baseline jitter, or rivers and valleys are not so easy to identified with the untrained eye, yet still detract significantly from the document.

5 Correction of Typographic Errors

The output of the typographical analysis system is a set of features of the zones of the typeset documents which are not consistent with the model specifications. Since a majority of the decisions are subjective, it would be useful to provide an interactive environment for editing the document, taking into account interpretations made by the system. A graphical interface will be provided that allows components to be modified at any level of the logical or physical document hierarchy. The attributes can be changed on-line and the document re-rendered. A pop-up window will be attached to each document component to provide access to the attributes.

6 Conclusions

In this paper we have provided an approach to image-based typographic analysis of documents. The

The gray level inconsistency in the "A" suggests that the character was started at the bottom, and consistency is used to trace through the letter "a". A light end and feather are used to define the direction of the "g". The CCW motion of the "o" is suggested by the discontinuity at the top and the direction which the *Downward* can be determined from the light end points.

7 Conclusions

We have addressed the problem of extracting temporal information from static images of handwriting, and described a taxonomy of classes which have proven useful for this purpose. We have shown that a solution requires that special at-

[4] V. Govindaraja and S. Srikan. Separating hand written text from overlapping non-textual contents. In *Proc. of the Int. Workshop on Frontiers in Handwriting Recognition*, pages 111-119, 1991.

[5] T.S. Huang. *Image Sequence Analysis*. Berlin/Heidelberg: Springer-Verlag, 1981.

[6] J. C. Pan and S. Lee. Office tracing applications of signatures. In *Proc of CVPR*, pages 679-686, 1991.

[7] J. C. Simon and K. Zerhouni. Robust description of a line image. In *Proc of the First Int Conf on Document Anal and Recog*, pages 3-14, 1991.

[8] M. A. Thinning method based on cell structure. In *Proc. of Frontiers in Handwriting Recog*, pages 29-32, 1990.

(a)

The gray level inconsistency in the "A" suggests that the character was started at the bottom, and consistency is used to trace through the letter "a". A light end and feather are used to define the direction of the "g". The CCW motion of the "o" is suggested by the discontinuity at the top and the direction which the *Downward* can be determined from the light end points.

7 Conclusions

We have addressed the problem of extracting temporal information from static images of handwriting, and described a taxonomy of classes which have proven useful for this purpose. We have shown that a solution requires that special at-

[4] V. Govindaraja and S. Srikan. Separating hand written text from overlapping non-textual contents. In *Proc. of the Int. Workshop on Frontiers in Handwriting Recognition*, pages 111-119, 1991.

[5] T.S. Huang. *Image Sequence Analysis*. Berlin/Heidelberg: Springer-Verlag, 1981.

[6] J. C. Pan and S. Lee. Office tracing applications of signatures. In *Proc of CVPR*, pages 679-686, 1991.

[7] J. C. Simon and K. Zerhouni. Robust description of a line image. In *Proc of the First Int Conf on Document Anal and Recog*, pages 3-14, 1991.

[8] M. A. Thinning method based on cell structure. In *Proc. of Frontiers in Handwriting Recog*, pages 29-32, 1990.

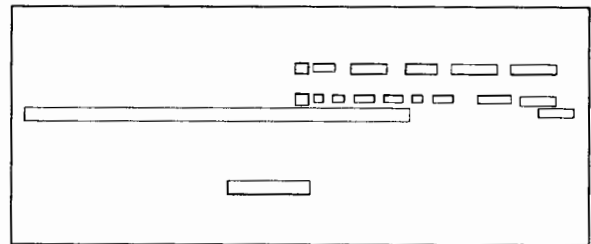
(b)

(c)

Figure 3: a) The bitmap image, b) character, and c) word level components.

problem requires a spatial understanding of the document layout as well as knowledge of the proper syntax. Our system performs a page synthesis from the stream of formatting commands defined in a DVI file. We have developed a representation which preserves the two-dimensional layout, the read-order and the component attributes. From this hierarchical representation of the page layout we can extract and analyze relevant typographic features such as margins, line and character spacing, and figure placement, among others.

In many cases, the distinction between the class of typographic *errors* and what can be termed *stylistic variation* is subjective, especially when generic models are used. Because of this, it is not realistic to provide fixed rules for interpretation, but rather to provide a method for analysis and the derivation of ambiguities between the models and the document instance. An interactive capability for the manipulation of the var-



(a)

Row	Zone	Element	Description
5	2	-	word spacing
7	2	-	word spacing
8	1	1	right margin
8	2	6	right margin
13	1	7	right margin
13	1	7	baseline shift

Figure 4: a) Regions detected with inconsistent attributes and the textual description of the ambiguities

ious possible interpretations has been proposed.

References

- [1] Jacques André, Richard Furuta, and Vincent Quint, editors. *Structured Documents*. Cambridge University Press, 1989.
- [2] U.S. Government Printing Board, editor. *A Manual of Style*. Gramercy Publishers, NY, 1986.
- [3] Robert Bringhurst. *The Elements of Typographic Style*. Hartley & Marks, Vancouver, BC, 1992.
- [4] Richard Furuta. Important papers in the history of document preparation systems: Basic sources. *Electronic Publishing: Origination, Dissemination, and Design*, 5(1):19-44, March 1992.
- [5] Richard Furuta, Vincent Quint, and Jacques André. Interactively editing structured documents. *Electronic Publishing: Origination, Dissemination, and Design*, 1(1):19-44, April 1988.
- [6] International Standard Organisation. *Text and Office Systems—Office Document Architecture (ODA) and Interchange Format*, 1989. International Standard 8613.
- [7] ISO. *Text and Office Systems—Standard Generalized Markup Language*, October 1986. Document ISO 8879-1986(E).
- [8] Leslie Lamport. *L^AT_EX: A Document Preparation System*. Addison-Wesley, Reading, MA, 1985.
- [9] H. Samet. *The Design and Analysis of Spatial Data Structures*. Addison-Wesley, Reading, MA, 1990.