An Optimization Methodology for Document Structure Extraction on Latin Character Documents

Jisheng Liang, *Member*, *IEEE Computer Society*,
Ihsin T. Phillips, *Senior Member*, *IEEE*, and Robert M. Haralick, *Fellow*, *IEEE*

Abstract—In this paper, we give a formal definition of a document image structure representation and we formulate document image structure extraction as a partitioning problem: Finding an optimal solution partitioning the set of glyphs of an input document image into a hierarchical tree structure where entities within the hierarchy at each level have similar physical properties and compatable semantic labels. We present a unified methodology that is applicable to construction of document structures at different hierarchical levels. An iterative, relaxation-like method is used to find a partitioning solution that maximizes the probability of the extracted structure. All the probabilities used in the partioning process are estimated from an extensive training set of various kinds of measurements among the entities within the hierarchy. The offline probabilities estimated in the training then drive all decisions in the online document structure extraction. We have implemented a text line extraction algorithm using this framework. The algorithm was evaluated on the UW-III database of some 1,600 scanned document image pages. An area-overlap measure is used to find the correspondence between the detected entities and the ground-truth. For a total of 105,020 text lines, the text line extraction algorithm identifies and segments 104,773 correctly, an accuracy of 99.76 percent. The detail of the algorithm is presented in this paper.

Index Terms—Document image analysis, statistical pattern analysis, text line extraction, performance evaluation.

1 Introduction

The technology of the OCR (Optical Character Recognition) and document image analysis systems today has reached far beyond the simple transformation of textual document images into sequences of characters and words. Instead, the technology has moved onto the development of the correct detection of the structure of the document, as it is the first step in almost every document image understanding task such as information retrieval, document routing and archiving, and perhaps even for the duplicate document detection. In this paper, we present a methodology for formulating and solving the document structure detection problem.

Document structures can be represented in a hierarchy. Given a document image, the end result of a document structure detection algorithm is a hierarchical structure that captures the physical layout and the logical meaning of the document. The top of the hierarchical structure presents the entire page, and the bottom of the structure includes all glyphs on the document. Entities in the hierarchy are labeled and are associated with a set of attributes describing the nature of the entities. For example, the extracted characters would reside at the bottom of the hierarchy, each character would be labeled as a "glyph" and the attributes for the glyph may be the ASCII/UNICODE value, the font style, and the position of the character. The next level up may be words, then, text lines, and then, text and nontext zones, and so forth.

As posed in this paper, the problem is to find an optimal solution partitioning of the set of glyphs of an input document into a hierarchical tree structure where entities within the hierarchy at each level have similar properties and compatible semantic labels. A Bayesian framework is used to assign and update the probabilities during the partitioning process. An iterative, relaxation-like method is used to find the partitioning solution that maximizes the probability of the extracted structure. All the probabilities used in the partitioning process are estimated from an extensive training set of various kinds of measurements among the entities within the hierarchy. The offline probabilities estimated in the training then drive all decisions in the online document structure extraction.

Using this methodology, we have implemented a text line extraction algorithm. The algorithm was evaluated on the 1,600 scanned document images of the UW-III database [23], [19]. For a total of 105,020 text lines, the text line extraction algorithm identifies and segments 104,773 correctly, an accuracy of 99.76 percent.

2 Literature Review

The construction of a document hierarchy, given an input document, consists of two major steps. The first step is the correct detection and partitioning of entities within the

- *J. Liang is with Insightful Corporation, 1700 Westlake Ave. N., Suite 500, Seattle, WA 98109. E-mail: jliang@insightful.com.*
- *I.T. Phillips is with the Department of Computer Science, Queens College, City University of New York, 65-30 Kissena Blvd., Flushing, NY 11367. E-mail: {yun@image.cs.qc.edu,}*
- *R.M. Haralick is with the Department of Computer Science, Graduate Center, City University of New York, 365 Fifth Ave., New York, NY 10016. E-mail: haralick@gc.cuny.edu.*

Manuscript received 22 July 1999; revised 3 Dec. 2000; accepted 10 Jan. 2001. Recommended for acceptance by D. Dori.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 110286.

hierarchy, and the second step is the correct classification of those detected entities.

Most of the current document image analysis algorithms in the literature can be categorized either by the direction that the algorithms take to construct document hierarchies or by the methods or strategies that the algorithms employ in constructing the hierarchies. Algorithms can be classified into two major direction classes: 1) bottom-up and 2) top-down. There are algorithms that use a mixture of the two (hybrid), however, their main construction directions are still within these two classes.

Algorithms taking the bottom-up approach, by-and-large are the majority, the document structure extraction task is done by recursively grouping smaller document entities into larger document entities. For example, the characters are extracted from the connected-components, the basic document entities. Extracted characters are grouped into words; words are grouped to form text lines, and text lines are grouped into text blocks and text columns, and so forth. For algorithms using the top-down approach, the segmentation task is done in a reverse order, i.e., by recursively dividing the document from larger document entities to smaller entities. The classic example of this approach is Nagy and Seth's X-Y tree [2].

Relative to the strategies, the algorithms can be classified into two major categories: 1) rule-based/grammar-driven and 2) statistical-based (either parametric or nonparametric.) The rule-based/grammar-driven algorithms use a set of ad hoc rules or predefined syntax rules of the grammars to derive decisions in the process of dividing or grouping document entities. The number of rules used can range from a few to a very large set. And the ad hoc rules or the syntax of the grammar can be very domain specific. As for the statistical-based algorithms, their free parameters are estimated in the offline training processes. The estimated parameters are used in the decisions which govern the processes of dividing or grouping the document entities.

Most of the current document structure analysis algorithms assume some prior knowledge of the typographical and layout conventions of the document. One way of knowledge acquisition is manually generating the heuristic rules, or grammars, by carefully looking through a set of representative document pages. Not only is this task tedious and requires much expertise, but the handcrafted rules tend to be brittle and only work for a specific kind of document. This means that these algorithms successfully analyze only those documents having previously defined formatting properties and may fail on documents that do not meet the formatting assumptions. Moreover, most of the rule or grammar-based algorithms use fixed, global distance thresholds for grouping or dividing document entities. These thresholds are either ad hoc given or empirically determined. However, the measurements made on the document entities may have errors and the knowledge about the document style may be ambiguous. The problem of uncertainty in knowledge and evidence and the information integration of multiple evidences have to be solved. Hence, to build a document structure extraction algorithm that is error-tolerant and robust remains a challenging task for the researchers and developers in the field. The following are a handful of selected algorithms within the above defined categories.

Ittner and Baird [5] developed a system for isolating blocks, lines, words, and symbols within images of machine-printed textual documents. Their algorithm is independent of language and writing systems by using a small number of nearly universal typesetting and layout conventions. They use a wide variety of techniques, including Fourier theory, computational geometry, and statistical decision theory. The algorithm achieves a measure of robustness by following a "global-to-local" strategy.

Dengel and Dubiel [6] developed a system for partitioning raster images of business letters into logically labeled area items. It employs various knowledge resources utilizing spatial and geometric characteristics about the formal style of business letters. Due to the unsupervised learning of document models, the system allows the establishment of a specific decision tree classifier which serves as the knowledge representation.

Chen [11] describes a text word, line, and block segmentation algorithm for horizontal rectangular layouts. The morphological closing transform is applied to the binary document image. The extracted segments are classified as words or nontext according to their size. The algorithm groups extracted words into text lines based on statistical models of the colinearity and equal spacing of words within the text lines. The text lines are then merged into text blocks according to a statistical model of the homogeneity of height, width, leading, and justification within text blocks.

Ha et al. [3], Liang et al. [4] describe an algorithm based on recursive cutting of connected component projection profiles for segmenting binary document images into zones; zones are classified as textual and nontextual; then the text zones are decomposed to text blocks, text lines, and words. Empirically determined thresholds are used at each cutting step.

Palumbo et al. [12] developed a postal automation system that locates and interprets destination address blocks on letter mail pieces with a high success rate and high speed. The system is based on data-driven processing. It extracts primitive information from the image and groups the information into possible address blocks. The evidence combination tool integrates pieces of evidence generated for a block into a single block labeling hypothesis. The best candidate block is selected by verifying the consistency of labeling hypotheses by using spatial relations.

Etemad et al. [13] present an algorithm for layout-independent document page segmentation based on document texture using multiscale feature vectors and fuzzy local decision information. Multiscale feature vectors are classified locally using a neural network to allow soft/fuzzy multiclass membership assignments. Segmentation is performed by integrating soft local decision vectors to reduce their "ambiguities."

Kise et al. [8] present a method of page segmentation based on the approximated area Voronoi diagram. The Voronoi diagram helps obtain the candidates of boundaries of document components from page images with non-Manhattan layout and a skew. Then, the candidates are utilized to estimate the intercharacter and interline gaps without the use of domain-specific parameters to select the boundaries.

Wang and Yagasaki [9] present a page segmentation method called block selection which segments the page image into categorized blocks and provides a tree structure to represent the page blocks for selection. Block selection identifies the major document elements such as text, picture, table, frame, and line. The direction of text could be horizontal, vertical, slanted, or mixed. No skew correction is involved regardless of the document style.

Jain and Yu [10] use the traditional bottom-up approach based on the connected component extraction to efficiently implement page segmentation and region identification. They use a document model that preserves top-down generation information. This method is applicable to documents from various technical journals and can accommodate moderate amounts of skew and noise.

In order to build a near-perfect system, it is necessary to replace heuristics by systematics and rely on mathematical optimization rather than intuition. Only the algorithm of Chen [11] takes a systematic approach to integrating the evidence sources they use. Their algorithm computes Bayesian estimates of the text lines and text blocks based on statistical models. However, their algorithm's underlying models ignore the fact that some of the parameters are dependent on the text's font size. Thus the algorithm may achieve poor results for documents containing a wide variety of fonts and sizes.

Well-defined and unambiguous performance measures, combined with reproducible experiments, are necessary in every area of science and technology. However, it is clear that many of the early works on document analysis systems provide illustrative results and hardly any have their techniques tested on significant sized data sets and give quantitative performance measures [1]. The main reasons are the lack of precise and concise mathematical document image models, the lack of accurate document ground truth data to train and test the algorithms, and the lack of appropriate and quantitative performance metrics, and evaluation protocol.

Currently, no standard testing procedures exist for measuring and comparing algorithms within a document analysis system. Randriamasy and Vincent [18] proposed a pixel-level and region-based approach to compare segmentation results and manually generated regions. An overlap matching technique is used to associate each region of one of the two sets, to the regions in the other set it has a nonempty intersection. Since the black pixels contained in the regions are counted rather than the regions themselves, this method is independent of representation scheme of regions. Quantitative evaluation of segmentation is derived from the cost of incorrect splittings and mergings. Working on the bit-map level, their technique involves extensive computation. They assume there is only one text orientation for the whole page. Kanai et al. [16] proposed a text based method to evaluate the zone segmentation performance. They compute an edit distance which is based on the number of edit operations (text insertions, deletions, and block moves) required to transform an OCR output to the correct text. The cost of segmentation itself is derived by comparing the costs corresponding to manually and automatically zoned pages. This metric can be used to test "black-box" commercial systems, but is not able to help users categorize the segmentation errors. This method only deals with text regions. They assume the OCR performance is independent of the segmentation performance. Garris [17] proposed a scoring method which computes the coverage and efficiency of zone segmentation algorithm. The box distance and box similarity between zones are computed to find the matching pairs. This technique provides some numbers (scores), which are not able to help users analyze errors.

In this paper, we adopt an engineering approach to systematically characterizing the document structures based on a large document image database, and develop statistical methods to extract the document structure from the image. In Section 6, we develop suitable quantitative performance measures to evaluate our document structure analysis algorithms. In Section 7, we report the results of our optimization methodology with a study of its application to text line extraction. The next sections give a formal definition for the document hierarchy we use and the unified methodology for document structure extraction.

3 Document Structure Representation

3.1 Polygonal Structure

A *polygonal area* on a document page is given by a pair (θ, I), where $\theta \in \Theta$ specifies the label that designates the physical type of content, i.e., text block, text line, word, table, equation, drawing, halftone, etc., and I is the area enclosed by boundary of the polygon. The boundary of the polygon is given as the sequence of line segments connecting the successive vertices of the polygon. The vertices are given as a clockwise ordered list of points. A polygon is *homogeneous* if all its area is of one physical type and there is a standard reading order for the content within the area. Two polygons are physical *adjacent* if each has a significant length side that are nearly parallel to each other, and are separated by a divider. A set $\mathcal{A}$ of nonoverlapping homogeneous polygonal areas and the properties associated with $\mathcal{A}$ is called a *polygonal structure*. $V : \wp(\mathcal{A}) \rightarrow \Lambda$ specifies measurement made on subset of $\mathcal{A}$, where Λ is the measurement space.

A polygonal structure is associated with the following basic sets and mappings.

- Content Type Θ is the set of physical types (text block, text line, word, table, equation, drawing, etc.). $C : \mathcal{A} \rightarrow \Theta$ associates polygonal areas with their physical types of content. Γ is the set of functional types (paragraph, section, word, title, heading, caption, abstract, author, footnote, page number, etc.). $M : \mathcal{A} \rightarrow \Gamma$ associates polygonal areas with their functional types of content.
- Content. Σ is the alphabet consisting of symbols. Let Σ^* be the set of all sequences of symbols from Σ. Let $\mathcal{A}_t \subseteq \mathcal{A}$ be the set of text polygonal areas. $O : \mathcal{A}_t \rightarrow \Sigma^*$ associates text polygonal areas with their contents.
- Format Attribute. We denote by $\mathcal{F}$ the set of format attributes (font type, font size, font style,

justification, indentation, etc.). $S: \Gamma \rightarrow \mathcal{F}$ specifies the format attributes for each functional type of content.

- Location. We denote by $\mathcal{L}$ the set of qualitative locations (top left, bottom, middle, etc.). $\mathcal{P} \subseteq \Gamma \times \mathcal{L}$ specifies the permitted locations of different types of content.
- Spatial Relation. $\mathcal{D}$ is the set of dividers (white space, ruling, etc.). $T: \Gamma \times \Gamma \rightarrow \mathcal{D}$ specifies the convention for the type of divider used between different types of content (interline spacing, intercolumn spacing, spacing between figure and caption, etc.).
- Reading Order. Let $\mathcal{A} = \{A_1, \cdots, A_K\}$ be the set of polygonal areas. The reading order R of $\mathcal{A}$ is a tuple $(r_1, \cdots, r_K)$ which is a permutation of $(1, \cdots, K)$. We define a binary relation $ReadBefore(a, b)$ (is to be interpretated as a is read before b). $ReadBefore$ is a partial ordering relation, since if $r_i \in R$ and $r_j \in R$, then

1. $ReadBefore(r_i, r_i)$ is false,
2. if
$$ReadBefore(r_i, r_j) \rightarrow ReadBefore(r_j, r_i)$$
is false,
3. if $ReadBefore(r_i, r_j)$, and
$$ReadBefore(r_j, r_k) \rightarrow ReadBefore(r_i, r_k).$$

3.2 Document Structure Hierarchy

We represent a document structure as a hierarchical structure.

Let Φ be the set of all polygonal structures on a given document page. We define the relation $\subseteq_\Phi$ in Φ as: $x \subseteq_\Phi y$ if and only if $x \subseteq y$ and $x, y \in \Phi$. This relation is a partial ordering E and satisfies

1. $(a, a) \in E, \forall a \in \Phi$,
2. $(a, b) \in E, \text{and} (b, a) \in E \Rightarrow a = b$,
3. $(a, b) \in E, (b, c) \in E \Rightarrow (a, c) \in E$.

We denote by $\Theta = \{\theta_1, \theta_2, \cdots, \theta_k\}$ the set of physical content types on the document page. Let $\{\Phi_1, \cdots, \Phi_k\}$ be a partition of Φ, where each Φ_i is a set of mutually disjoint polygonal structures

$$\Phi_i = \{\phi | \phi \in \Phi, C(\phi) = \theta_i\}$$

and if there exists Φ_j and $i > j$, then $\forall\ \phi_p \in \Phi_i$, either $\phi_p \subseteq_\Phi \Phi_j$, or ϕ_p and Φ_j are disjoint. An example of the document hierarchy is shown in Fig. 1.

Problem Statement. Given a document page Φ_1, extract a hierarchical document structure $\{\Phi_2, \Phi_3, \cdots \Phi_k\}$, that maximizes the conditional probability $P(\Phi_2, \Phi_3, \cdots \Phi_k | \Phi_1)$ of the extracted structure given the document image page.

The problem of document hierarchy construction is usually decomposed into subproblems, based on some assumptions on conditional independence and ordering. For example, the joint probability may be decomposed as

$$\begin{aligned} &P(\Phi_2, \Phi_3, \cdots \Phi_6 | \Phi_1) = P(\Phi_6 | \Phi_1) P(\Phi_4, \Phi_2 | \Phi_6) \\ &P(\Phi_3 | \Phi_4, \Phi_2) P(\Phi_5 | \Phi_4, \Phi_6), \end{aligned} \tag{1}$$

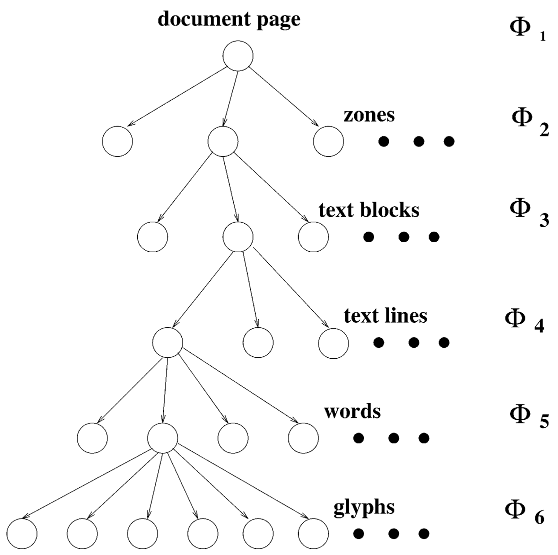


Fig. 1. Illustrates a document hierarchy.

where $\Phi_1, \Phi_2, \cdots, \Phi_6$ represent page, zone, text block, text line, word, and glyph structures, respectively. Assuming the decomposition in (1) is valid, the problem exhibits an optimal substructure and a dynamic programming can be applied to determine an optimal solution that maximizes the probability. However, the validation of the decomposition and the optimal solution of the joint probability are out of scope of this paper.

In this paper, we introduce a unified approach that is applicable to each subproblem, which finds an optimal solution that maximizes each individual probability in (1). The problem of extraction of document hierarchy levels is formulated and a generic algorithm is presented in the following sections.

3.3 Document Structure Analysis Problem

Let Φ_i be the source level and let Φ_j be the target level in a document hierarchy. And let $\mathcal{A}$ be the set of polygonal areas in Φ_i. Let Π be a partition of $\mathcal{A}$ and L be a set of labels that can be assigned to elements of the partition. Function $f: \Pi \rightarrow L$ associates each element of Φ_j with a label. $V: \wp(\mathcal{A}) \rightarrow \Lambda$ specifies measurement made on subset of $\mathcal{A}$, where Λ is the measurement space.

The document structure extraction problem can be formulated as follows: *Given the initial set $\mathcal{A}$, find a partition Π of $\mathcal{A}$, and a labeling function $f: \Pi \rightarrow L$ that assigns each element $\tau \in \Pi$ a label in L, that maximizes the probability,*

$$\begin{aligned} &P(V(\tau): \tau \in \Pi, f, \Pi | \mathcal{A}) = P(V(\tau): \tau \in \Pi | \mathcal{A}, \Pi, f) \\ &P(\Pi, f | \mathcal{A}) = P(V(\tau): \tau \in \Pi | \mathcal{A}, \Pi, f) P(f | \Pi, \mathcal{A}) P(\Pi | \mathcal{A}). \end{aligned} \tag{2}$$

By making the assumption of conditional independence that when the label $f(\tau)$ is known, then no knowledge of other labels will alter the probability of $V(\tau)$, we can decompose the probability in (2) into

$$\begin{aligned} &P(V(\tau): \tau \in \Pi, f, \Pi | \mathcal{A}) \\ &= \prod_{\tau \in \Pi} P(V(\tau) | f(\tau)) P(f | \Pi, \mathcal{A}) P(\Pi | \mathcal{A}). \end{aligned} \tag{3}$$

The possible labels in set L is dependent on the target_level and on the specific application. For example, $l \in L$ could be text content, functional content type, style attribute, and so forth.

The above proposed formulation can be uniformly applied to the construction of the document hierarchy at any level, e.g., text word, text line, and text block extractions and so forth. For example, as for text line extraction, given a set of glyphs, the goal of the text line extraction is to partition glyphs into a set of text lines, each text line having homogeneous properties, and the text lines' properties within the same region being similar. The text lines' properties include, deviation of glyphs from the baseline, direction of the baseline, text line's height, and text lines' width, and so forth. As for text block segmentation, given a set of text lines, text block segmentation groups text lines into a set of text blocks, each block having homogeneous formatting attributes, e.g., homogeneous leading, justification, and the attributes between neighboring blocks being similar.

Note that, a brute-force method for finding the optimal solution for (3) is to search through all possible partitions with all possible labels and select the configuration which produces the highest conditional probability in (3). The number of partitions of an n-set is called a Bell number and it gets very large. Fortunately, the entities on a printed document are usually aligned with certain reading order. For example, glyphs within a text line are within a proximity of one another. A glyph on the top of the document never needs to be hypothesized as grouped with those glyphs on the bottom of the document. Thus, with the ordering constraint, the partitioning and labeling problem can be reformulated. A unified methodology for solving the document structure analysis problem is given in the next section.

4 A Unified Methodology

Given an initial set $\mathcal{A}$, we first construct the reading order of the elements of $\mathcal{A}$. Let $A = (A_1, A_2, \cdots, A_M)$ be a linearly ordered set (chain in $\mathcal{A}$) of input entities. Let $\mathcal{G} = \{Y, N\}$ be the set of grouping labels. Let A^P denote a set of element pairs, such that $A^p \subset A \times A$ and

$$A^p = \{(A_i, A_j) | A_i, A_j \in A \text{ and } j = i + 1\}.$$

Function $g : A^p \to \mathcal{G}$, associates each pair of adjacent elements of A with a grouping label, where $g(i) = g(A_i, A_{i+1})$. Then, the partition probability $P(\Pi|A)$ can be computed as follows,

$$\begin{aligned}
&P(\Pi|A) \\
&= P(g|A) = P(g(1), \cdots, g(N-1)|A_1, \cdots, A_N) \\
&= P(g(1)|A_1, A_2) \times \cdots P(g(N-1)|A_{N-1}, A_N) \\
&= \prod_{i=1}^{N-1} P(g(i)|A_i, A_{i+1}).
\end{aligned}$$

Therefore, the joint probability can be further decomposed as

$$\begin{aligned}
P(V(\tau) : \tau \in \Pi, f, \Pi|A) = \prod_{\tau \in \Pi} P(V(\tau)|f(\tau)) \\
\times P(f|\Pi, A) \prod_{i=1}^{N-1} P(g(i)|A_i, A_{i+1}).
\end{aligned} \tag{4}$$

In principle, we want to find the joint pair of functions (g, f) that maximizes the probability in (4). Such a search could be done by brute force. Each of the 2^{N-1} different g functions determine a partition. Once a partition is given, the optimal f function can be determined. To avoid the exponential search, we use an iterative search method for finding the consistent partition and labeling that monotonically increases the joint probability of (4).

Algorithm 4.1 *Consistent partition and labeling.*

```
1. Determine initial partition
       Let $t = 0, \Pi^t = \{\{A_m\}\}_{m=1}^{M}$.
   a. Compute $P_i^0(Y) = P(g(i) = Y|A_i, A_{i+1})$ and
          $P_i^0(N) = P(g(i) = N|A_i, A_{i+1})$,
      where $1 \le i \le M - 1$.
   b. Let $R \subseteq A \times A$ and
          $R = \{(A_i, A_{i+1}) | P_i^0(Y) > P_i^0(N)\}$.
      Update partition
          $\Pi^{t+1} = \{\tau | \tau = \{A_i, A_{i+1}, \cdots, A_j\}$, where
          $(A_k, A_{k+1}) \in R, k = i, \cdots, j - 1\}$          (5)
2. Let $t = 1$. Search for optimal partition adjustment
   Repeat
   • For $i = 1$ to $M - 1$ Do
     - If $A_i \in U,\ A_{i+1} \in W,\ U \neq W$ where
           $U, W \in \Pi^t$,
       Then,
       a. Let $T = U \bigcup W$, and
              $\hat{\Pi} = T \bigcup (\Pi^t - U - W)$
       b. Find labeling $f$ by maximizing
              $P_{label} = \prod_{\tau \in \hat{\Pi}} P(V(\tau)|f(\tau)) P(f|A, \hat{\Pi})$
       c. $P_i^t(\hat{Y}) \propto P_i^0(Y) \times P_{label}$, and
              $P_i^t(\hat{N}) = P_i^{t-1}(N)$.
     - Else If $A_i \in W$ and $A_{i+1} \in W$, where
           $W = \{A_k, \cdots, A_i, A_{i+1}, \cdots, A_j\}$,
       Then,
       a. $S = \{A_k, \cdots, A_i\}$ and
              $T = \{A_{i+1}, \cdots, A_j\}$
              $\hat{\Pi} = (\Pi^t - W) \bigcup S \bigcup T$.
```

```
        b. Find labeling $f$ by maximizing
               $P_{label} = \prod_{\tau \in \hat{\Pi}} P(V(\tau)|f(\tau))P(f|A, \hat{\Pi})$
        c. $P_i^t(\hat{N}) \propto P_i^0(N) \times P_{label}$, and
               $P_i^t(\hat{Y}) = P_i^{t-1}(Y)$
      End (For-loop)
    • Select $k$ such that,
               $k = \arg\max_i(\max\{\hat{P}_i^t(Y), \hat{P}_i^t(N)\})$
    • If $P_k^t(\hat{Y}) > P_k^t(\hat{N})$, Then
        - $T = U \bigcup W$ where $A_k \in U, A_{k+1} \in W$
        - $\Pi^{t+1} = (\Pi^t - U - W) \bigcup T$
      Else, $W = \{A_i, \cdots, A_k, A_{k+1}, \cdots, A_j\}$,
        - Let $S = \{A_i, \cdots, A_k\}$ and $T = \{A_{k+1}, \cdots, A_j\}$
        - $\Pi^{t+1} = (\Pi^t - W) \bigcup S \bigcup T$
    • If $P(V, f, \Pi^{t+1}|A) \leq P(V, f, \Pi^t|A)$, end and return $\Pi^t$.
      Else, let $t = t + 1$ and continue.
End (Repeat)
```

All the probabilities used in this algorithm can be estimated from a large set of data. The probability estimation procedure is given in the next section.

5 Probability Estimation

Controlled experiments are an important component of computer vision, for the controlled experiment demonstrates that the algorithm, designed by the computer-vision researcher, recognizes, locates, and measures what it is designed to do from image data [20]. A properly designed scientific experiment provides evidence to accept or reject the hypothesis that the algorithm performs on a specified accuracy level.

The discrete contingency tables are used to represent the joint and conditional probabilities (see Section 5.1). In Section 5.2, we describe the experimental protocol for estimating probabilities used in the document analysis algorithms, from a significant sized training set.

5.1 Probability Representation

Discrete contingency tables are used to represent the joint and conditional probabilities used in our unified methodology. Each variable of the table has a finite number of mutually exclusive states. If A is a variable with states $a_1, \cdots, a_n$, then $P(A)$ is a probability distribution over these states:

$$P(A) = (x_1, \cdots, x_n), \; x_i \geq 0, \; \sum_{i=1}^{n} x_i = 1,$$

where x_i is the probability of A being in state a_i [14]. If the variable B has states $b_1, \cdots, b_m$, then $P(A|B)$ is an $n \times m$ table containing numbers $P(a_i|b_j)$. $P(A, B)$, the joint probability for the variables A and B, is also an $n \times m$ table.

TABLE 1
Illustrates an Example of the Cell Count Table

		B	No		Yes	
		A	No	Yes	No	Yes
E	D	C				
No	No	No	52	123	24	14
		Yes	12	56	11	56
	Yes	No	12	78	56	23
		Yes	45	67	87	23
Yes	No	No	23	32	12	123
		Yes	45	65	23	89
	Yes	No	90	12	234	128
		Yes	9	12	1	3

It consists of a probability for each configuration (a_i, b_j). When the fundamental rule $P(A|B)P(B) = P(A, B)$ is used on variables A and B, the procedure is to apply the rule to the $n \times m$ configurations (a_i, b_j): $P(a_i|b_j)P(b_j) = P(a_i, b_j)$. This means that in the table $P(A|B)$, for each j the column for b_j is multiplied by $P(b_j)$ to obtain the table $P(A, B)$.

Once definitions for the discrete states are established, the training data can be used to estimate a joint probaiblity by simply counting. If the training data is given as a matrix D,

$$D = \begin{pmatrix} d_{11} & \dots & d_{1K} \\ & \vdots & \\ d_{I1} & \dots & d_{IK} \end{pmatrix}, \tag{6}$$

where d_{ik} is the state value for the kth variable of the ith record, then

$$P(s_1, \dots, s_k) = \frac{\#\{i \mid (d_{i1}, \dots, d_{iK}) = (s_1, \dots, s_K)\}}{I}.$$

An example of the cell count table is shown in Table 1. It has five variables and two states for each variable. Therefore, the table has total of 32 cells.

5.2 Estimation of Probability Distribution

Creating a model usually involves determining the variables to observe, collecting and recording the data observations, studying graphics and summaries of the collected data to reveal low-dimensional relationships between variables, and choosing a model describing the important relationships seen or hypothesized in the data [15]. We quantize the value of each continuous variable into a finite number of mutually exclusive states and compute the cell count table from the data. A tree structure quantization is used to partition the value of a variable into bins. At each node of the tree, we search through all possible threshold candidates on each variable and select the one which gives the minimum value of entropy. In growing a tree, the binary partitioning algorithm recursively splits the data in each node until either the node is homogeneous or the node contains too few observations. In order to construct the quantized table from the tree, one follows the path from the root to the leafs within a certain level and record the splits made on each variable. The bins on each variable form the cells in the space. The total number of cells, is predetermined, based on the memory limitation and the number of samples in the training set. Given this number, one can

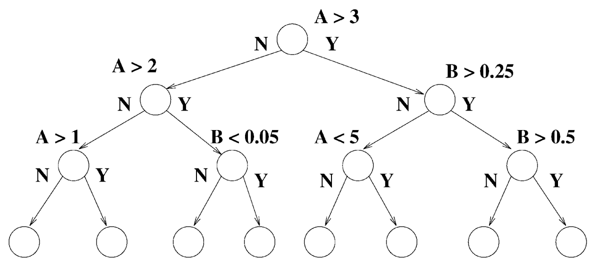


Fig. 2. Display a constructed tree-based model used for quantization.

determine how many levels of nodes will be used to form the cells. For example, suppose a domain has two variables A and B. First, we collect and record the data observations. Then, a classification tree training process is applied to the observed data and the constructed tree in shown in Fig. 2. If we want to limit the total number of cells under 20, the nodes with depth up to three are used to construct the table. In our example, four thresholds are determined on variable A and the variable B is split three times. Therefore, A and B are quantized into five bins and four bins, respectively. The constructed table with 20 cells is shown in Table 2.

6 Evaluation Criteria and Measurements

In general, the performance evaluation of any algorithm can be done by testing the algorithm on a selected testing data set and then evaluating its results against the corresponding ground-truth of the test set. On selecting the test data set, a large quantity of ground-truth data, varying in quality, is required in order to give an accurate measurement of the performance of an algorithm under different conditions. On the evaluation, a set of well-defined metrics needs to be established to measure the performance of the algorithm.

There are three steps involved in the evaluation. The first step is finding the correspondences between the ground-truth entities and the detected entities. The second step is the measurements on the goodness of the matches and the third step, is the assignment of an overall performance score to the algorithm. These steps are given next.

6.1 Matching

Suppose we are given two sets $\mathcal{G} = \{G_1, G_2, \cdots, G_M\}$ for the ground-truthed entities and $\mathcal{D} = \{D_1, D_2, \cdots, D_N\}$ for the detected entities. The comparison of $\mathcal{G}$ and $\mathcal{D}$ can be made in terms of the following two kinds of measures:

$$\sigma_{ij} = \frac{\mathrm{Area}(G_i \cap D_j)}{\mathrm{Area}(G_i)} \text{ and } \tau_{ij} = \frac{\mathrm{Area}(G_i \cap D_j)}{\mathrm{Area}(D_j)}, \tag{7}$$

where $1 \le i \le M$, $1 \le j \le N$, and $\mathrm{Area}(A)$ represents the area of A. The measures in the above equation constitute two matrices $\Sigma = (\sigma_{ij})$ and $T = (\tau_{ij})$. Notice that, σ_{ij} indicates how much portion of G_i is occupied by D_j, and τ_{ij} indicates how much portion of D_j is occupied by G_i. Our strategy of performance evaluation is to analyze these matrices to determine the correspondence between two sets of polygonal areas:

- one-to-one match ($\sigma_{ij} \approx 1$ and $\tau_{ij} \approx 1$);
- one-to-zero match ($\sigma_{ij} \approx 0$ for all $1 \le j \le N$);
- zero-to-one match ($\tau_{ij} \approx 0$ for all $1 \le i \le M$);
- one-to-many match
$$(\sigma_{ij} < 1 \text{ for all } j, \text{ and } \textstyle\sum_{j=1}^{N} \sigma_{ij} \approx 1);$$
- many-to-one match
$$(\tau_{ij} < 1 \text{ for all } i, \text{and} \textstyle\sum_{i=1}^{M} \tau_{ij} \approx 1);$$
- many-to-many match (others).

An example of matching between a set of ground truth entities and the detected entities is illustrated in Fig. 3. By computing their area overlap, we construct two matrices, $\Sigma = (\sigma_{ij})$ and $T = (\tau_{ij})$, shown in Table 3. In this example, we find a one-to-one match (G_1 to D_1), a one-to-many match (G_2 to D_2 and D_3), a one-to-zero match (G_3 to nothing), and a many-to-many match (G_4, G_5 and G_6 to D_4 and D_5).

6.2 Performance Measurement

Once the matching between detected structures and ground-truth structures is established, a performance measure can be computed. A one-to-one match means an object G_i is correctly identified by the segmentation process as D_j. A one-to-zero match is the case when a certain object G_i is not detected by the segmentation (misdetection) and vise versa for the zero-to-one match (false alarm). If an entity G_i matches to a number of detected entities, we call it a splitting detection. It is a merging detection when two or more objects in $\mathcal{G}$ are identified as an object D_j. The many-to-many matches are called spurious detections.

Let us denote the probability of a matching ($G^i \subset G$ is identified as $D^i \subset D$ in the sample) as

$$P^i(G, D) = \frac{\#G^i + \#D^i}{\#G + \#D}. \tag{8}$$

Then, the performance measure of a partition algorithm is defined as

$$f_{partition} = \sum_{i \in M} W^i P^i(G, D), \tag{9}$$

TABLE 2
Illustrates a Contingency Table, Where the Quantization of Variables Are Determined Using the Tree Structure Shown in Fig. 2

	$A < 1$	$1 \le A < 2$	$2 \le A < 3$	$3 \le A < 5$	$A \ge 5$
$B < 0.05$					
$0.05 \le B < 0.25$					
$0.25 \le B < 0.5$					
$B \ge 0.5$					

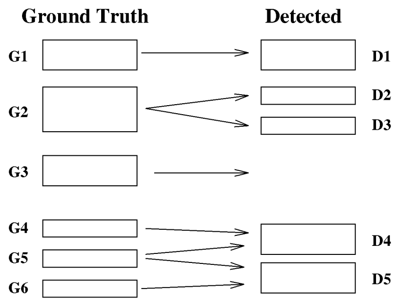


Fig. 3. Illustrates correspondence between ground-truth and detected structures.

where

$$M = \{correct, miss, false, merging, splitting, spurious\}$$

is the set of possible matching, and W^i is the weight assigned to each type of matching.

7 An Implementation: Text-Line Extraction Algorithm

We applied the methodology introduced in this paper to the text line extraction problem. Without loss of generality, we assume that the reading direction of the text lines on the input page is left-to-right and top-to-bottom. The text line extraction algorithm starts with the set of connected-component (glyph) bounding boxes from a textual document page's binary image. The glyphs in the set are sorted by their locations on the page. For each glyph in the sorted set, we locate its adjacent right neighbor. Associated with each glyph pair is a grouping probability indicating how probable a pair with its horizontal and vertical distance relationships will be within the same text line. An initial set of text lines are constructed based on the local grouping probabilities.

Next, the baseline direction is computed for each text line in this set and the median of all the baselines computed is taken as the page skew angle. If the skew angle is greater than a predetermined threshold, the page is rotated, according to the skew angle, and the process repeats.

Then, we start to assign labels to extracted text line segments and adjust the grouping, by maximizing the homogeneous labeling probabilities. The measurements that we make on each line segment are, the text line's x-height, deviation of glyphs from the text line's baseline, and the baseline's direction. We group extracted text lines into zones, where the text lines share the similar left, center, or right edges. Then, the text lines within each text zone are examined, statistically, to see whether the text lines are consistent with their neighbors or need to be split or merged. For example, we may split a text line when it "crosses" two horizontal text zones. Or, we may split a text line when its x-height is double in size in comparison to its vertical neighboring text lines within the same zone. Similarly, we may merge two adjacent text lines (in horizontal or vertical directions) within the same zone if the merge results in a better statistical fit with the global trend of the text zone it belongs.

The homogeneous labeling and the grouping adjustment process is repeated until no improvement on the grouping and labeling probability is achieved. Fig. 4 gives an overview of the text line extraction algorithm.

Algorithm.

1. Extract and order glyphs. We apply a standard connected-component analysis algorithm to the input image to obtain the glyph set, $\mathcal{A} = \{A_1, A_2, \cdots, A_M\}$. For each $A_i \in \mathcal{A}$, we search for its adjacent right neighbor, A_j. The glyph set $\mathcal{A}$ is rearranged into a set, partially ordered by this adjacency relationship, which consists of a group of linearly ordered glyph sequences. The right adjacency is defined in Section 7.1.
2. Compute local grouping probabilities. Let $A = \{A_1, A_2, \cdots, A_N\}$ be a linearly ordered subset of $\mathcal{A}$. For each pair of adjacent glyphs (A_i, A_{i+1}), we compute the probability that they are within the same text line:
$$P(\text{SameLine}(i, i+1) | A_i, A_{i+1}), \qquad (10)$$
based on observations on the spatial relationships between the pair. The measurements are described in Section 7.1
3. Group adjacent glyphs. Given the linking probability $P(g(i))$ between a pair of adjacent glyphs A_i and A_{i+1}, if $P(g(i) = Y) > P(g(i) = N)$, we group the pair into a text line; otherwise, A_i and A_{i+1} are in different lines.
During the initial partition,
$$P(g(i)) = P(\text{SameLine}(i, i+1) | A_i, A_{i+1}),$$
and this step yields our initial text line set, $T = \{T_1, T_2, \cdots, T_K\}$.
4. Assign a homogeneous label to text lines. In this step, we compute the likelihood of an extracted segment T_k having the homogeneous properties of a text line:
$$P(V(T_k) | f(T_k)), \qquad (11)$$

TABLE 3
Illustrates the Area Overlap Matrices Computed from the Example in Fig. 3

	D_1	D_2	D_3	D_4	D_5
G_1	0.95				
G_2		0.45	0.51		
G_3					
G_4				0.7	
G_5				0.37	0.29
G_6				0.8	

$\Sigma = (\sigma_{ij})$

	D_1	D_2	D_3	D_4	D_5
G_1	0.9				
G_2		0.85	0.91		
G_3					
G_4				0.4	
G_5				0.25	0.3
G_6					0.45

$T = (\tau_{ij})$

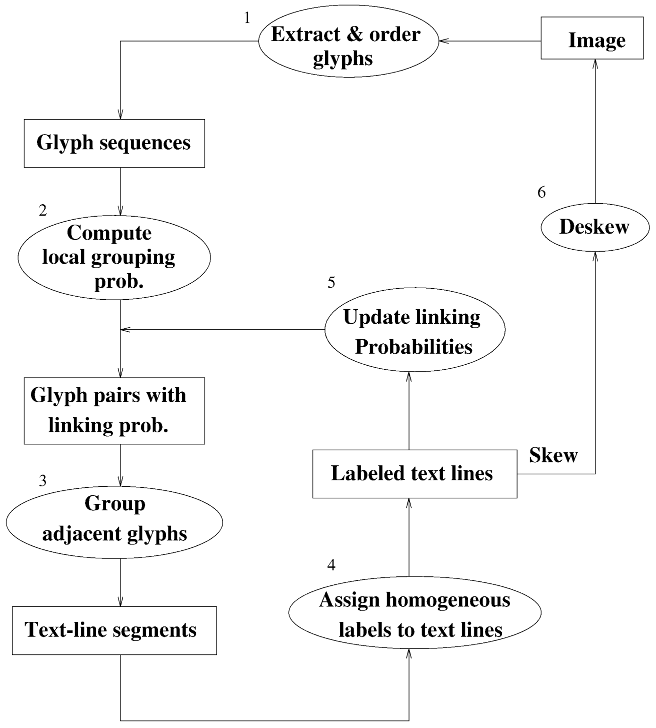

1
Extract & order glyphs
Image
Glyph sequences
2
Compute local grouping prob.
6
Deskew
5
Update linking Probabilities
Glyph pairs with linking prob.
Labeled text lines
Skew
3
Group adjacent glyphs
4
Assign homogeneous labels to text lines
Text-line segments

Fig. 4. Illustrates the processing steps of the text line extraction algorithm.

where the measurement $V(T_k)$ includes deviation of glyphs from the text line's baseline, x-height of the text line, and direction of the baseline. The computation of baseline and x-height is given in Section 7.2.

The text lines within the same neighborhood usually have similar physical properties, such as text line height, left and right edges, etc. We group text lines into zones by clustering the edges of all text lines, where the text lines within the same zone share the similar left, center or right edges. Given a group of extracted text lines, we determine f, assignment of the homogeneous text line labels to extracted line segments, by maximizing

$$P_{label} = \prod_{T_k \in T} P(V(T_k)|f(T_k))P(f|T, A). \tag{12}$$

The grouping of text lines into zones and computation of the probability $P(f|T, A)$ are given in Section 7.3.

5. Update linking probability and adjust partition. Given the computed labeling probabilities, we update the linking probability $P(g(i))$, between each pair of adjacent glyphs:

$$P(g(i)) \propto P(\text{SameLine}(i, i+1)|A_i, A_{i+1})P_{label}. \tag{13}$$

During each iteration, the adjustment which produces the maximum improvement of the linking probability is selected. Then, we continue to Step 3 and adjust the grouping according to the updated linking probabilities. If there is no improvement on the linking probability, we stop the iteration and return the extracted text lines. The detail of this step is given in Section 7.4.

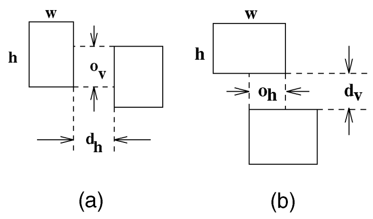


Fig. 5. Illustrates the spatial relations between two bounding boxes that are (a) horizontally adjacent and (b) vertically adjacent.

6. Page deskew. The median of all the computed baselines' direction for the entire set T is taken as the page skew angle, θ_{skew}. If the $\theta_{skew} > threshold_{skew}$, we rotate the image by $-\theta_{skew}$, and the process repeats from Step 1.

Fig. 9a and Fig. 9b illustrate the text line detection process. Fig. 9a(a) shows a set of connected component bounding boxes. The extracted initial text line segments by merging pairs of connected components are illustrated in Fig. 9a(b). We notice some text lines are split while some are merged across different columns. Fig. 9b(c) plots the extracted text zones by grouping the edges of text segments. Finally, the corrected text lines given the observations on text zones are shown in Fig. 9b(d).

The following sections give detailed description of steps in the text line extraction algorithm.

7.1 Adjacent Glyphs and Grouping Probability

Let $\mathcal{A} = \{A_1, A_2, \cdots, A_M\}$ be the set of glyphs extracted from a document page. Each glyph $A_i \in \mathcal{A}$ is represented by a bounding box (x, y, w, h), where x, y is the coordinate of top-left corner, and w and h are the width and height of the bounding box, respectively. The spatial relations between two adjacent boxes are shown in Fig. 5.

For a pair of bounding boxes a and b, the horizontal distance $d_h(a, b)$ and vertical distance $d_v(a, b)$ between them are defined as

$$\begin{aligned} d_h(a,b) &= \begin{cases} x_b - x_a - w_a & \text{if } x_b > x_a + w_a \\ x_a - x_b - w_b & \text{if } x_a > x_b + w_b \\ 0 & \text{otherwise} \end{cases} \\ d_v(a,b) &= \begin{cases} y_b - y_a - h_a & \text{if } y_b > y_a + h_a \\ y_a - y_b - h_b & \text{if } y_a > y_b + h_b \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{14}$$

The horizontal overlap $o_h(a, b)$ and vertical overlap $o_v(a, b)$ are defined as

$$o_h(a,b) = \begin{cases} x_a + w_a - x_b & \text{if } x_b > x_a, x_b < x_a + w_a \\ x_b + w_b - x_a & \text{if } x_a > x_b, x_a < x_b + w_b \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

$$o_v(a,b) = \begin{cases} y_a + h_a - y_b & \text{if } y_b > y_a, y_b < y_a + h_a \\ y_b + h_b - y_a & \text{if } y_a > y_b, y_a < y_b + h_b \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

Let $A_a = (x_a, y_a, w_a, h_a)$ and $A_b = (x_b, y_b, w_b, h_b)$ be two glyphs. We define A_b as a right neighbor of A_a if $A_b \neq A_a, x_b > x_a$, and $o_v(a, b) > 0$. Let B_a be the set of

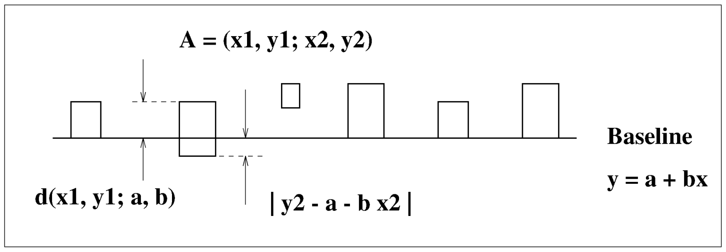


Fig. 6. Illustrates measurements made for estimating baseline and x-height.

right neighbors of A_a. The adjacent right neighbor of A_a is defined as

$$\arg\min_{A_i \in B_a}(d_h(a,i) | x_i > x_a, o_v(a,i) > 0).$$

For each linked pair, A_a and A_b, we associate it with the probability, $P(\text{SameLine}(a,b)|A_a, A_b)$, that indicates how probable they belong to the same text line. Given the observations of their heights and widths, and the distance and the overlaps between the pair: h_a, w_a, h_b, w_b, $d(a,b)$, $o(a,b)$, we compute the probability that A_a and A_b belong to the same text line as:

$$P(\text{SameLine}(a,b)|h_a, w_a, h_b, w_b, d(a,b), o(a,b)).$$

7.2 Baseline, X-Height, and Skew Angle

The baseline coordinate of a text line is estimated using a robust estimator. We fit a straight line $y(x;a,b) = a + bx$ through the bottom-right corner of glyph boxes. We use the bottom-right corner because ascenders are used more often in Latin texts than descenders (see Fig. 6).

The merit function to be minimized is

$$\sum_{i=1}^{N} |y_i - a - bx_i|. \quad (17)$$

The median d_M of a set of numbers d_i is also the value which minimizes the sum of the absolute deviations

$$\sum_i |d_i - d_M|. \quad (18)$$

It follows that, for fixed b, the value of a that minimizes the merit function is $a = \text{median}\{y_i - bx_i\}$, where

$$0 = \sum\nolimits_{i=1}^{N} x_i sgn(y_i - a - bx_i).$$

This equation can be solved by the bracketing and bisection method [21].

Given a set of baseline angles $\{\theta_1, \theta_2, \cdots, \theta_P\}$, the skew angle of page is estimated as

$$\theta_{skew} = \text{median}\ \{\theta_1, \theta_2, \cdots, \theta_P\}. \quad (19)$$

If skew angle θ_{skew} is larger than the threshold, $threshold_{skew}$, we rotate the page by $-\theta_{skew}$, using the technique given in [22].

For each given text line T_k and its estimated baseline (a,b), we compute the absolute deviation of glyph positions from the estimated baseline

$$\sigma(T_k, a, b) = \sum_{i=1}^{N} |y_i - a - bx_i|. \quad (20)$$

The x-height of a text line is estimated by taking the median of the distance from the top-left corner of each glyph box to the baseline

$$xh(T_k) = \text{median}\ \{d(x_i, y_i; a, b) | 1 \le i \le N\}. \quad (21)$$

Given the observations on the text line T_k, we can compute the likelihood that T_k has the homogeneous property of a text line

$$P(xh(T_k), \sigma(T_k, a, b)) | \text{TextLine}(T_k)). \quad (22)$$

The linking probability $P(g(i,j))$ between a pair of glyphs, $A_i, A_j \in T_k$, is then updated according to the probability of T_k having homogeneous text line property

$$\begin{aligned} &P(g(i,j)) \propto P(\text{SameLine}(i,j)|A_i, A_j) \\ &P(xh(T_k), \sigma(T_k, a, b)) | \text{TextLine}(T_k)). \end{aligned} \quad (23)$$

7.3 Text-Zone Formation

Given a set of text line bounding boxes

$$T = \{T_1, T_2, \cdots, T_M\},$$

our goal is to group them into a set of zones, where the text lines within each zone share the similar edges.

Given an entity box (x, y, w, h), its horizontal projection (Fig. 7) is defined as

$$\text{horz-profile}[j] = \text{horz-profile}[j] + 1, x \le j < x + w.$$

And its vertical projection is

$$\text{vert-profile}[j] = \text{vert-profile}[j] + 1, y \le j < y + h.$$

We assign the left edge of T_i to be x_i, the right edge of T_i to be $x_i + w_i$, and the center of T_i to be $x_i + w_i/2$. The vertical edge projection on the three edges of the text line bounding boxes of all $T_i \in T$ is defined as (see Fig. 8)

$$\begin{aligned} C_{left}[j] &= C_{left}[j] + 1, j = x \\ C_{center}[j] &= C_{center}[j] + 1, j = x + w/2 \\ C_{right}[j] &= C_{right}[j] + 1, j = x + w. \end{aligned} \quad (24)$$

Algorithm 7.1 *Text-zone detection.*

1. Compute the horizontal projection profile of all text line boxes.
2. Segment the page into a set of large regions, by making cut at the gaps of horizontal projection profile, where the width of gap is larger than a certain threshold. The threshold is determined by the median height of detected text lines.
3. For each region

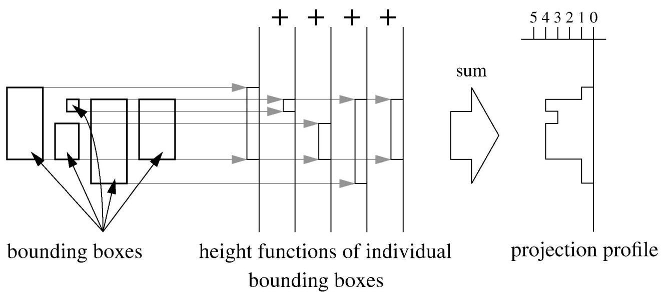


Fig. 7. Illustrates the horizontal projection of bounding boxes.

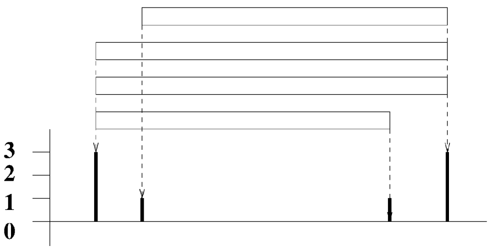


Fig. 8. Illustrates computation of vertical projection count to determine regions' dominant left and right edges.

a. Compute the vertical projection count C of the left edges E_l, right edges E_r, and center edges E_c of text line boxes.
b. Find a place which has the highest total count within its neighborhood of width w:

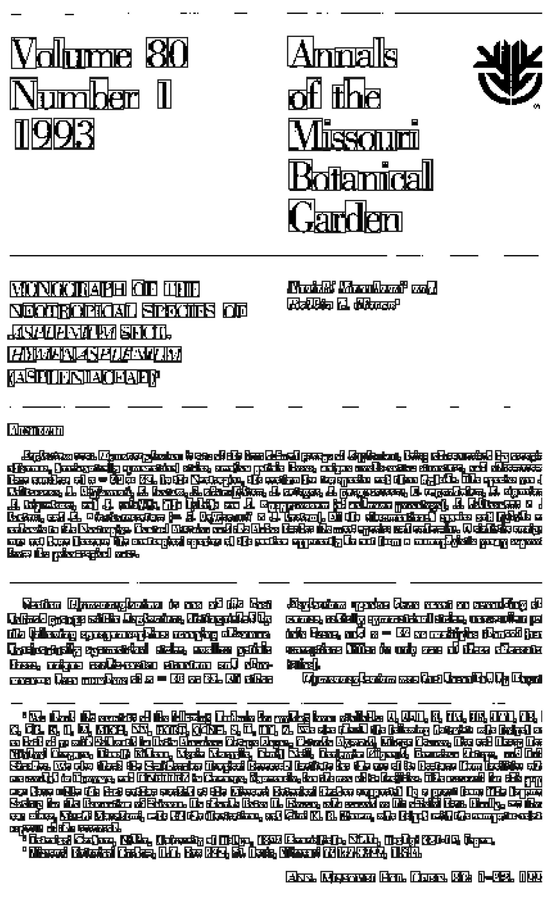
$$x = \arg_{i,j} \max\left(\sum_k C_{i,k}, i \in \{l, r, c\},\right.$$
$$\left. j - \frac{1}{2}w \le k < j + \frac{1}{2}w\right), \tag{25}$$

where w is determined by the dominant text line height within the region.

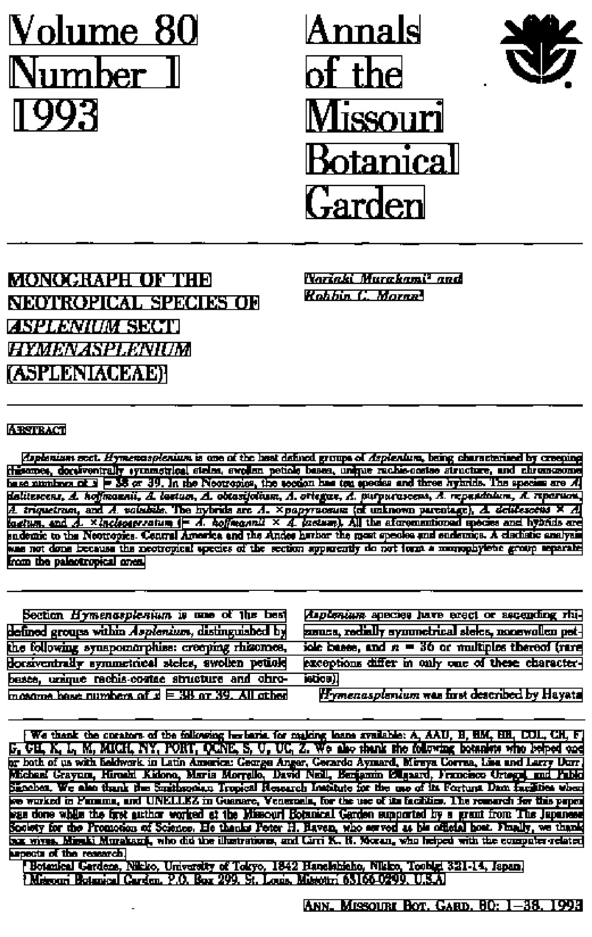
c. Determine the zone edge as the median of edges E_{ik}, within the neighborhood

$$j - \frac{1}{2}w \le k < j + \frac{1}{2}w.$$

d. For each edge E_{ik}, finding its corresponding edge of the other side of the box $E_{jk}, j \neq i$
e. Determine the other edges of this zone by taking the median of E_{jk}



(a)



Volume 80
Number 1
1993

Annals
of the
Missouri
Botanical
Garden

MONOGRAPH OF THE NEOTROPICAL SPECIES OF *ASPLENIUM* SECT. *HYMENASPLENIUM* (ASPLENIACEAE)[1]

Noriaki Murakami[2] and *Robbin C. Moran*[3]

ABSTRACT

Asplenium sect. *Hymenasplenium* is one of the best defined groups of *Asplenium*, being characterized by creeping rhizomes, dorsiventrally symmetrical steles, swollen petiole bases, unique rachis-costae structure, and chromosome base numbers of $x = 38$ or 39. In the Neotropics, the section has ten species and three hybrids. The species are *A. delitescens*, *A. hoffmannii*, *A. laetum*, *A. obtusifolium*, *A. ortegae*, *A. purpurascens*, *A. repandulum*, *A. riparium*, *A. triquetrum*, and *A. volubile*. The hybrids are *A. ×papyraceum* (of unknown parentage), *A. delitescens* × *A. laetum*, and *A. ×inclusgerratum* (= *A. hoffmannii* × *A. laetum*). All the aforementioned species and hybrids are endemic to the Neotropics. Central America and the Andes harbor the most species and endemics. A cladistic analysis was not done because the neotropical species of the section apparently do not form a monophyletic group separate from the paleotropical ones.

Section *Hymenasplenium* is one of the best defined groups within *Asplenium*, distinguished by the following synapomorphies: creeping rhizomes, dorsiventrally symmetrical steles, swollen petiole bases, unique rachis-costae structure and chromosome base numbers of $x = 38$ or 39. All other *Asplenium* species have erect or ascending rhizomes, radially symmetrical steles, nonswollen petiole bases, and $n = 36$ or multiples thereof (rare exceptions differ in only one of these characteristics).

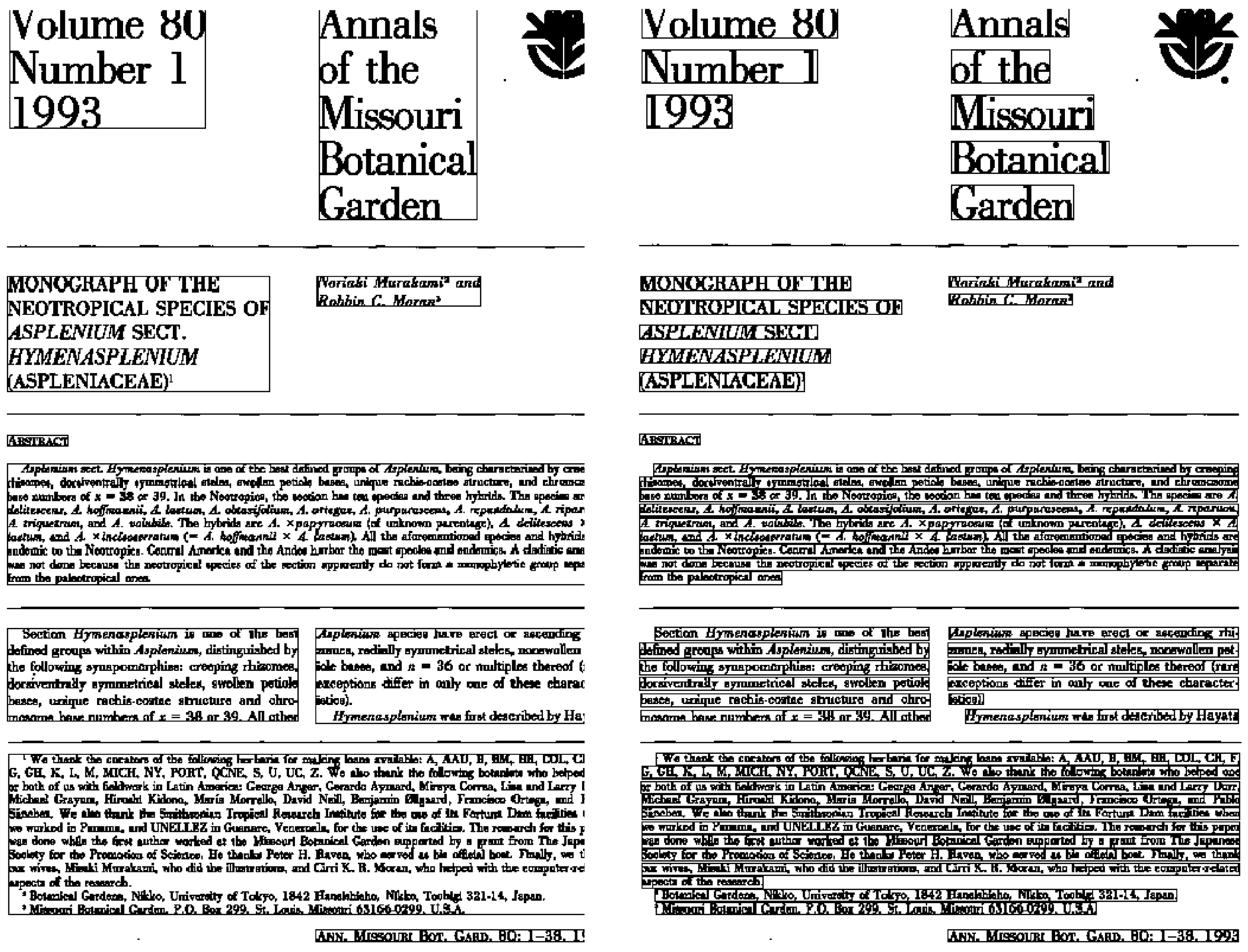
Hymenasplenium was first described by Hayata

[1] We thank the curators of the following herbaria for making loans available: A, AAU, B, BM, BR, COL, CR, F, G, GH, K, L, M, MICH, NY, PORT, QCNE, S, U, UC, Z. We also thank the following botanists who helped one or both of us with fieldwork in Latin America: George Argar, Gerardo Aymard, Mireya Correa, Lisa and Larry Dorr, Michael Grayum, Hiroshi Kidono, Maria Morello, David Neill, Benjamin Øllgaard, Francisco Ortega, and Pablo Sanchez. We also thank the Smithsonian Tropical Research Institute for the use of its Fortuna Dam facilities when we worked in Panama, and UNELLEZ in Guanare, Venezuela, for the use of its facilities. The research for this paper was done while the first author worked at the Missouri Botanical Garden supported by a grant from The Japanese Society for the Promotion of Science. He thanks Peter H. Raven, who served as his official host. Finally, we thank our wives, Misaki Murakami, who did the illustrations, and Cirri K. R. Moran, who helped with the computer-related aspects of the research.
[2] Botanical Gardens, Nikko, University of Tokyo, 1842 Hanaishicho, Nikko, Tochigi 321-14, Japan.
[3] Missouri Botanical Garden, P.O. Box 299, St. Louis, Missouri 63166-0299, U.S.A.

ANN. MISSOURI BOT. GARD. 80: 1–38. 1993

(b)

Fig. 9a. Illustrates a real document image overlaid with the extracted bounding boxes of (a) the connected components and (b) the initial text line segments.



Volume 80
Number 1
1993

Annals
of the
Missouri
Botanical
Garden

MONOGRAPH OF THE
NEOTROPICAL SPECIES OF
ASPLENIUM SECT.
HYMENASPLENIUM
(ASPLENIACEAE)[1]

Noriaki Murakami[2] *and*
Robbin C. Moran[3]

ABSTRACT

Asplenium sect. *Hymenasplenium* is one of the best defined groups of *Asplenium*, being characterized by creeping rhizomes, dorsiventrally symmetrical steles, swollen petiole bases, unique rachis-costae structure, and chromosome base numbers of $x = 38$ or 39. In the Neotropics, the section has ten species and three hybrids. The species are *A. delitescens*, *A. hoffmannii*, *A. laetum*, *A. obtusifolium*, *A. ortegae*, *A. purpurascens*, *A. repandulum*, *A. riparium*, *A. triquetrum*, and *A. volubile*. The hybrids are *A.* ×*papyraceum* (of unknown parentage), *A. delitescens* × *A. laetum*, and *A.* ×*incisoserratum* (= *A. hoffmannii* × *A. laetum*). All the aforementioned species and hybrids are endemic to the Neotropics. Central America and the Andes harbor the most species and endemics. A cladistic analysis was not done because the neotropical species of the section apparently do not form a monophyletic group separable from the paleotropical ones.

Section *Hymenasplenium* is one of the best defined groups within *Asplenium*, distinguished by the following synapomorphies: creeping rhizomes, dorsiventrally symmetrical steles, swollen petiole bases, unique rachis-costae structure and chromosome base numbers of $x = 38$ or 39. All other *Asplenium* species have erect or ascending rhizomes, radially symmetrical steles, nonswollen petiole bases, and $n = 36$ or multiples thereof (rare exceptions differ in only one of these characteristics).

Hymenasplenium was first described by Hayata

[1] We thank the curators of the following herbaria for making loans available: A, AAU, B, BM, BR, COL, CR, F, G, GH, K, L, M, MICH, NY, PORT, QCNE, S, U, UC, Z. We also thank the following botanists who helped one or both of us with fieldwork in Latin America: George Anger, Gerardo Aymard, Mireya Correa, Lisa and Larry Dorr, Michael Grayum, Hiroshi Kidono, Maria Morrello, David Neill, Benjamin Øllgaard, Francisco Ortega, and Pablo Sánchez. We also thank the Smithsonian Tropical Research Institute for the use of its Fortuna Dam facilities when we worked in Panama, and UNELLEZ in Guanare, Venezuela, for the use of its facilities. The research for this paper was done while the first author worked at the Missouri Botanical Garden supported by a grant from The Japanese Society for the Promotion of Science. He thanks Peter H. Raven, who served as his official host. Finally, we thank our wives, Misaki Murakami, who did the illustrations, and Cirri K. R. Moran, who helped with the computer-related aspects of the research.
[2] Botanical Gardens, Nikko, University of Tokyo, 1842 Hanaishicho, Nikko, Tochigi 321-14, Japan.
[3] Missouri Botanical Garden, P.O. Box 299, St. Louis, Missouri 63166-0299, U.S.A.

ANN. MISSOURI BOT. GARD. 80: 1–38. 1993

(c) (d)

Fig. 9b. Illustrates a real document image overlaid with the extracted bounding boxes of (c) the texts zones and (d) the corrected text lines.

f. Remove the text line boxes enclosed by the detect zone from T
g. Repeat until $T = \emptyset$.

If the interzone spacing between two adjacent zones is very small, it may cause the majority of text lines from those two zones to merge. On the other hand, a list-item structure usually has large gaps and this causes splitting errors. In order to detect these two cases, we compute the vertical projection profile of glyphs enclosed by each zone. If there is a zero-height valley in the profile, we compute the probability that a region Z should be split into two zones:

$$P(\text{TwoZone}(Z)|w_{gap}, n(Z), xh(Z), h(Z_l), h(Z_r), w(Z_l), w(Z_r)), \tag{26}$$

where w_{gap} is the width of the profile gap, $n(Z)$ is the total number of text lines within the current region Z, and $xh(Z)$ is the median of text lines' x-height within Z. $h(Z_l)$ and $w(Z_l)$, ($h(Z_r)$ and $w(Z_r)$) are the height and width of the region Z_l (Z_r) on the left (right) side of gap. If the probability is larger than a certain threshold, we split the region at the detected gap.

Given a pair of adjacent zones, Z_l and Z_r, the probability that they are part of the list-item structure is:

$$P(\text{ListItem}(Z_l, Z_r)|w_{gap}, h(Z_l), h(Z_r), w(Z_l), w(Z_r), n(Z_l), n(Z_r)), \tag{27}$$

where $n(Z_l)$ and $n(Z_r)$ are number of text lines within the left and right zones respectively.

7.4 Text Line Splitting and Merging

Given the detected zones, we can determine if a text line is horizontally merged or split, or vertically merged or split.

Given the observations on a text line $T = (A_1, A_2, \cdots, A_m)$ and its neighbors N_T within the same zone Z, we compute the probability that T is vertically consistent, merged, or split:

$$P(\text{v-consistent}(T, N_T)|h(T), h(N_T), h_c(T), h_c(N_T)), \tag{28}$$

where $h(T)$ is height of the text line T, $h(N_T)$ is the median of text line height in zone N_T, $h_c(T)$ is the median height of glyphs in T, and $h_c(N_T)$ is the median height of glyphs in N_T. Then, we can update the linking probability between a pair of adjacent glyphs A_i and A_j:

$$P(g(i,j)) \propto P(\text{SameLine}(i,j)|A_i, A_j)P(\text{v-consistent}(T, N_T)), \tag{29}$$

where $A_i \in T$, and $A_j \in Z$.

TABLE 4
Performance of the Text Line Extraction Algorithm

	Total	Correct	Splitting	Merging	Mis-False	Spurious
Ground Truth	105020	104773 (99.76%)	80 (0.08%)	78 (0.07%)	79 (0.08%)	10 (0.01%)
Detected	105019	104773 (99.77%)	172 (0.16%)	37 (0.04%)	25 (0.02%)	12 (0.01%)

Given a pair of adjacent text lines T_m and T_n within the same zone, we can update the linking probability between a pair of glyphs $A_i \in T_m$ and $A_j \in T_n$:

$$\begin{aligned}&P(g(i,j)) \propto P(\mathrm{SameLine}(i,j)|A_i, A_j, \mathrm{SameZone}(i,j))\\&\propto P(A_i, A_j|\mathrm{SameLine}(i,j))P(\mathrm{SameZone}(i,j)|\mathrm{SameLine}(i,j)).\end{aligned} \tag{30}$$

Similarly, if a text line is across two or more zones, we can update the linking probability for each pair of adjacent glyphs that belong to different zones

$$\begin{aligned}&P(g(i,j)) \propto P(\mathrm{SameLine}(i,j)|A_i, A_j, \mathrm{DiffZone}(i,j))\\&\propto P(A_i, A_j|\mathrm{SameLine}(i,j))P(\mathrm{DiffZone}(i,j)|\mathrm{SameLine}(i,j)).\end{aligned} \tag{31}$$

8 Experimental Results

The text line extraction algorithm described in this paper is evaluated on a total of 1,600 scanned images from the UW-III Document Image Database [23], [24], [19]. A three-fold cross-validation method is used to estimate the algorithm's performance. We partition the data set into three parts, use two parts to do the training and use the third part to test and evaluate the performance. The training and testing procedure is repeated three times using a different part for testing each time. Finally, the performance measures from three parts are combined as the overall performance of the algorithm on the data set.

In the UW-III database, a page is partitioned into a set of zones and each zone is assigned a physical content type, such as, text, math, table, drawing, figure, ruling, etc. For each text zone, the database provides the text ground-truth, the bounding boxes of words, text line, and text zone. However, the database does not provide ground-truth for any of the text entities within nontext zones.

Thus, we first evaluate the performance of our text line algorithm only on the text zone areas of the 1,600 pages, and then, we evaluate the algorithm on the entire pages. These two experiments are given as follows.

8.1 Experiment 1: Text Areas Only

In this experiment, we assume the text areas on a page are given. First, the connected components are extracted from the image. Then, we classify the components into two sets: One includes all components that are enclosed by the text areas; another consists of components belong to other regions, such as table, math, line drawing, etc.

The text line extraction algorithm is applied to the set of connected components that are within the text areas. The numbers and percentages of miss, false, correct, splitting, merging, and spurious detections are shown in Table 4. Of the 105,020 ground truth text lines, 99.76 percent of them are correctly detected, and 0.08 percent and 0.07 percent of lines are split or merged, respectively. Most of the missing errors are due to the rotated text.

A few cases that the algorithm failed are shown in Fig. 10. A vertical merging error was shown in Fig. 10a. Fig. 10b and Fig. 10c illustrate horizontal and vertical splitting errors due to the large spacing. A spurious error caused by warping is shown in Fig. 10d.

8.2 Experiment 2: Text and Nontext Areas

In this experiment, we apply the text line extraction algorithm on the whole page of each image in the test data set. First, the connected component analysis is applied to the input binary image. Then, a few simple measurements are made on each extracted component, i.e., height, width, aspect ratio, black pixel density within the component, etc. Based on these observations, we determine if a component has the attributes of a character. The components with extreme physical properties are removed from the set. And we run the text line extraction algorithm on all remained character-like components (glyphs).

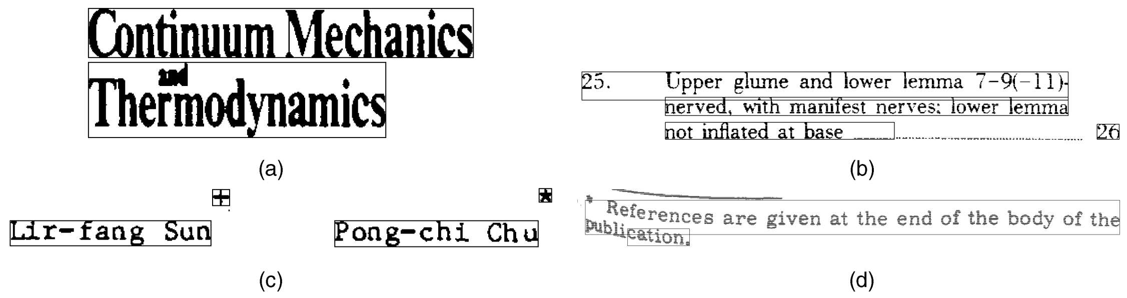
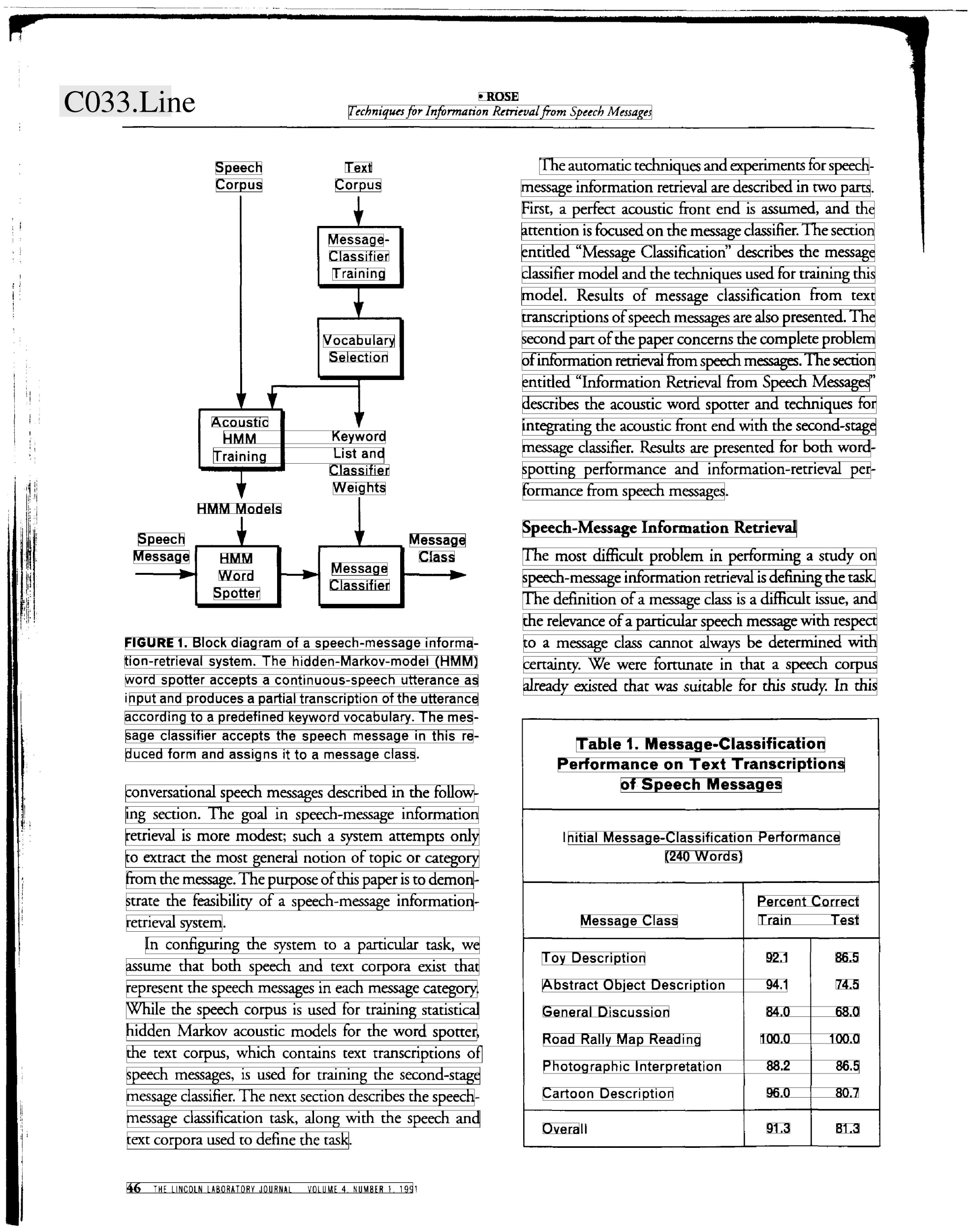

Continuum Mechanics
and
Thermodynamics
(a)
25. Upper glume and lower lemma 7–9(–11)-
nerved, with manifest nerves; lower lemma
not inflated at base 26
(b)
Lir-fang Sun
Pong-chi Chu
(c)
References are given at the end of the body of the
publication.
(d)

Fig. 10. Illustrates examples that the text detection algorithm failed.

TABLE 5
Performance of the Text Line Extraction Algorithm on Normal Sized Connected Components

	Total	Correct	Splitting	Merging	Mis-False	Spurious
Ground Truth	105020	103829 (98.87%)	638 (0.61%)	422 (0.40%)	66 (0.06%)	65 (0.06%)
Detected	106146	103829 (97.81%)	1653 (1.56%)	210 (0.20%)	36 (0.03%)	418 (0.39%)

Since no ground-truth are given to those text entities within the nontext zones in the data set, we are not able to evaluate the extracted text lines within the nontext zones. For the extracted text lines fall within the text areas, we compare their bounding boxes with the bounding boxes of the ground-truth text lines. The numbers and percentages of miss, false, correct, splitting, merging, and spurious detections are shown in Table 5. Fig. 11 illustrates the bounding boxes of the text lines extracted from areas of text, table and drawing. Fig. 12 illustrates examples where connected components in figures are grouped into text lines.


C033.Line
Speech Corpus
Text Corpus
Message-Classifier Training
Vocabulary Selection
Acoustic HMM Training
Keyword List and Classifier Weights
HMM Models
Speech Message
HMM Word Spotter
Message Classifier
Message Class

• ROSE
Techniques for Information Retrieval from Speech Messages

FIGURE 1. Block diagram of a speech-message information-retrieval system. The hidden-Markov-model (HMM) word spotter accepts a continuous-speech utterance as input and produces a partial transcription of the utterance according to a predefined keyword vocabulary. The message classifier accepts the speech message in this reduced form and assigns it to a message class.

conversational speech messages described in the following section. The goal in speech-message information retrieval is more modest; such a system attempts only to extract the most general notion of topic or category from the message. The purpose of this paper is to demonstrate the feasibility of a speech-message information-retrieval system.

In configuring the system to a particular task, we assume that both speech and text corpora exist that represent the speech messages in each message category. While the speech corpus is used for training statistical hidden Markov acoustic models for the word spotter, the text corpus, which contains text transcriptions of speech messages, is used for training the second-stage message classifier. The next section describes the speech-message classification task, along with the speech and text corpora used to define the task.

The automatic techniques and experiments for speech-message information retrieval are described in two parts. First, a perfect acoustic front end is assumed, and the attention is focused on the message classifier. The section entitled "Message Classification" describes the message classifier model and the techniques used for training this model. Results of message classification from text transcriptions of speech messages are also presented. The second part of the paper concerns the complete problem of information retrieval from speech messages. The section entitled "Information Retrieval from Speech Messages" describes the acoustic word spotter and techniques for integrating the acoustic front end with the second-stage message classifier. Results are presented for both word-spotting performance and information-retrieval performance from speech messages.

Speech-Message Information Retrieval

The most difficult problem in performing a study on speech-message information retrieval is defining the task. The definition of a message class is a difficult issue, and the relevance of a particular speech message with respect to a message class cannot always be determined with certainty. We were fortunate in that a speech corpus already existed that was suitable for this study. In this

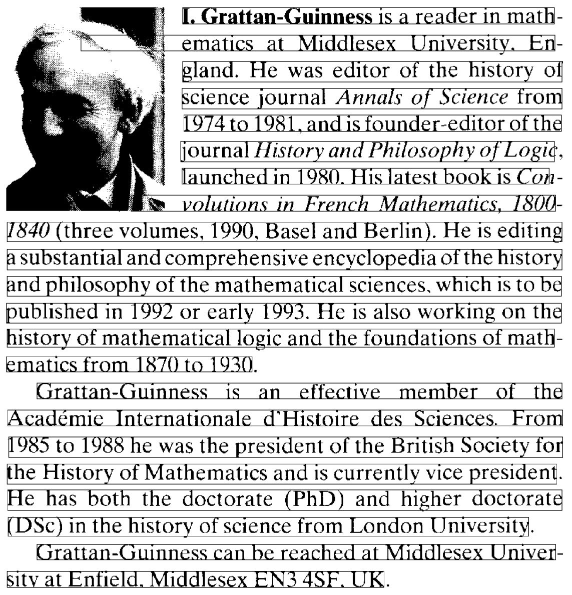
Table 1. Message-Classification Performance on Text Transcriptions of Speech Messages

Initial Message-Classification Performance (240 Words)

Message Class	Percent Correct Train	Percent Correct Test
Toy Description	92.1	86.5
Abstract Object Description	94.1	74.5
General Discussion	84.0	68.0
Road Rally Map Reading	100.0	100.0
Photographic Interpretation	88.2	86.5
Cartoon Description	96.0	80.7
Overall	91.3	81.3

46 THE LINCOLN LABORATORY JOURNAL VOLUME 4, NUMBER 1, 1991

Fig. 11. Illustrates a real document image overlaid with the bounding boxes of the text lines extracted from areas of text, table, and drawing.

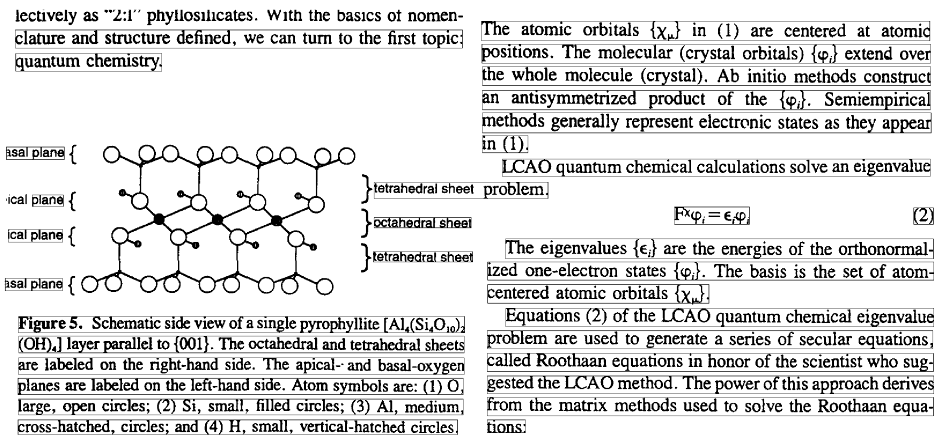


I. Grattan-Guinness is a reader in mathematics at Middlesex University, England. He was editor of the history of science journal *Annals of Science* from 1974 to 1981, and is founder-editor of the journal *History and Philosophy of Logic*, launched in 1980. His latest book is *Convolutions in French Mathematics, 1800-1840* (three volumes, 1990, Basel and Berlin). He is editing a substantial and comprehensive encyclopedia of the history and philosophy of the mathematical sciences, which is to be published in 1992 or early 1993. He is also working on the history of mathematical logic and the foundations of mathematics from 1870 to 1930.

Grattan-Guinness is an effective member of the Académie Internationale d'Histoire des Sciences. From 1985 to 1988 he was the president of the British Society for the History of Mathematics and is currently vice president. He has both the doctorate (PhD) and higher doctorate (DSc) in the history of science from London University.

Grattan-Guinness can be reached at Middlesex University at Enfield, Middlesex EN3 4SF, UK.

(a)



lectively as "2:1" phyllosilicates. With the basics of nomenclature and structure defined, we can turn to the first topic: quantum chemistry.

basal plane
apical plane
apical plane
basal plane
tetrahedral sheet
octahedral sheet
tetrahedral sheet

Figure 5. Schematic side view of a single pyrophyllite $[Al_4(Si_4O_{10})_2(OH)_4]$ layer parallel to {001}. The octahedral and tetrahedral sheets are labeled on the right-hand side. The apical- and basal-oxygen planes are labeled on the left-hand side. Atom symbols are: (1) O, large, open circles; (2) Si, small, filled circles; (3) Al, medium, cross-hatched, circles; and (4) H, small, vertical-hatched circles.

The atomic orbitals $\{\chi_\mu\}$ in (1) are centered at atomic positions. The molecular (crystal orbitals) $\{\varphi_i\}$ extend over the whole molecule (crystal). Ab initio methods construct an antisymmetrized product of the $\{\varphi_i\}$. Semiempirical methods generally represent electronic states as they appear in (1).

LCAO quantum chemical calculations solve an eigenvalue problem.

$$F^\chi\varphi_i = \epsilon_i\varphi_i \quad (2)$$

The eigenvalues $\{\epsilon_i\}$ are the energies of the orthonormalized one-electron states $\{\varphi_i\}$. The basis is the set of atom-centered atomic orbitals $\{\chi_\mu\}$.

Equations (2) of the LCAO quantum chemical eigenvalue problem are used to generate a series of secular equations, called Roothaan equations in honor of the scientist who suggested the LCAO method. The power of this approach derives from the matrix methods used to solve the Roothaan equations.

(b)

Fig. 12. Illustrates examples where connected components in (a) a half-tone and (b) a line-drawing are grouped into text lines.

9 Summary

In this paper, we formulated the document structures extraction as an optimal partitioning problem. The goal was to find an optimal solution partitioning the set of glyphs on a given document image into a hierarchical tree structure where entities within the hierarchy are associated with their physical properties and semantic labels. A Bayesian framework is used to assign and update the probabilities during the structures extraction process. An iterative, relaxation-like method is used to find the partitioning solution that maximizes the probability of the extracted structure given the document image data.

A text line extraction algorithm has been implemented to demonstrate the usage of this framework. This algorithm consists of two major components—offline statistical training and online text line extraction. The probabilities used within this algorithm are estimated from an extensive training set of various kinds of measurements of distances between the terminal and nonterminal entities with which the algorithm works. The offline probabilities estimated in the training then drive all decisions in the online segmentation module. The online segmentation module extracted and filtered the set of connected components of the input image to obtain a set of glyphs. Each glyph is linked to its adjacent neighbor to form glyph pairs. Associated with each link is the pair's linking probability. The entire text line extraction process became an iterative readjustment of the pairs' linking probabilities on the glyph set. The segmentation algorithm terminated when a decision can be made in favor for each link within the final set of text line segments.

The algorithm was tested on the 1,600 pages of technical documents within the UW-III database having a total of 105,020 text lines. The algorithm exhibited a 99.76 percent accuracy rate.

References

[1] R.M. Haralick, "Document Image Understanding: Geometric and Logical Layout," *Proc. IEEE Computer Soc. Conf. Computer Vision and Pattern Recognition*, pp. 385-90, June 1994.

[2] G. Nagy and S. Seth, "Hierarchical Representation of Optically Scanned Documents," *Proc. Seventh Int'l Conf. Pattern Recongnition,* pp 347-349, 1984.
[3] J. Ha, R.M. Haralick, and I.T. Phillips, "Document Page Decomposition Using Bounding Boxes of Connected Components of Black Pixels," *Proc. Document Recognition II,* L.M. Vincent and H.S. Baird, eds., pp. 140-151, 1995.
[4] J. Liang, J. Ha, R.M. Haralick, and I.T. Phillips, "Document Layout Structure Extraction Using Bounding Boxes of Different Entities," *Proc. Third IEEE Workshop Applications of Computer Vision,* pp 278-283, 1996.
[5] D.J. Ittner and H.S. Baird, "Language-Free Layout Analysis," *Proc. Second Int'l Conf. Document Analysis and Recognition,* pp. 336-40, 1993.
[6] A. Dengel and F. Dubiel, "Clustering and Classification of Document Structure: A Machine Learning Approach," *Proc. Third Int'l Conf. Document Analysis and Recognition,* pp. 587-591, 1995.
[7] F. Esposito, D. Malerba, and G. Semeraro, "Automated Acquisition of Rules for Document Understanding," *Proc. Second Int'l Conf. Document Analysis and Recognition,* pp. 650-654, 1993.
[8] 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, pp. 370-82, June 1998.
[9] S.Y. Wang and T. Yagasaki, "Block Selection: A Method for Segmenting Page Image of Various Editing Styles," *Proc. Third Int'l Conf. Document Analysis and Recognition,* pp. 128-135, 1995.
[10] A.K. Jain and B. Yu, "Document Representation and Its Application to Page Decomposition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 3, pp. 294-308, Mar. 1998.
[11] S. Chen, "Document Layout Analysis Using Recursive Morphological Transforms," PhD thesis, Univ. of Washington, 1995.
[12] P.W. Palumbo, S.N. Srihari, J. Soh, R. Sridhar, and V. Demjanenko, "Postal Address Block Location in Real Time," *Computer,* pp. 34-42, July 1992.
[13] K. Etemad, D. Doermann, and R. Chellappa, "Multiscale Segmentation of Unstructured Document Pages Using Soft Decision Integration," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 1, pp. 92-96, Jan. 1997.
[14] F.V. Jensen, *An Introduction to Bayesian Networks.* Springer, 1996.
[15] *S-PLUS Guide to Statistics.* MathSoft, 1997.
[16] J. Kanai, S.V. Rice, T.A. Nartker, and G. Nagy, "Automated Evaluation of OCR Zoning," *IEEE Trans. Pattern Analysis and Machine Intelligence,* pp. 86-90, vol. 17, 1995.
[17] M.D. Garris, "Evaluating Spatial Correspondence of Zones in Document Recognition Systems," *Proc. Int'l Conf. Image Processing,* pp. 304-307, vol. 3, Oct. 1995.
[18] S. Randriamasy and L. Vincent, "Benchmarking Page Segmentation Algorithm," *Proc. IEEE Computer Soc. Conf. Computer Vision and Pattern Recognition,* pp. 411-416, 1994.
[19] I.T. Phillips, "User's Reference Manual for the UW English/Technical Document Image Database III, Image Database Manual," technical document, UW III, 1996.
[20] R.M. Haralick and L.G. Shapiro, *Computer and Robot Vision, Vol. I.* Addison-Wesley, 1992.
[21] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, *Numerical Recipes in C.* Cambridge Univ. Press, 1992.
[22] S. Chen, R.M. Haralick, and I. Phillips, "Automatic Text Skew Estimation in Document Images," *Proc. Third Int'l Conf. Document Analysis and Recognition (ICDAR'95),* pp. 1153-1156, Aug. 1995.
[23] I.T. Phillips, S. Chen, and R.M. Haralick, "CD-ROM Document Database Standard," *Proc. Int'l Conf. Document Analysis and Retrieval (ICDAR 93),* pp. 478-483, Oct. 1993.
[24] I.T. Phillips, J. Ha, R.M. Haralick, and D. Dori, "The Implementation Methodology for the CD-ROM English Document Database," *Proc. Int'l Conf. Document Analysis and Retrieval,* pp. 484-487, Oct. 1993.

Jisheng Liang received the BS degree from Tianjin University, Tianjin, China, in 1992, and the PhD degree from the University of Washington, Seattle, in 1999, both in electrical engineering. His research interests include image processing, pattern recognition, document image analysis, and information retrieval. He has published a dozen papers in the area of document image analysis. Dr. Liang is currently working as a research scientist at Insightful Corporation in Seattle, Washington where he has been involved in projects concerning cross-language document retrieval, character recognition, medical imaging, and video processing. Dr. Liang is a member of the IEEE Computer Society.

Ihsin T. Phillips received the BS, MS, and PhD degrees in 1979, 1981, and 1984, respectively, all in computer science, from the University of Maryland, College Park. In 1984, she joined the Department of Computer Science at the University of Maryland, Baltimore campus, as an assistant professor. In 1985, she joined the Department of Computer Science and Software Engineering at Seattle University, Seattle, Washington, where she was promoted to associate professor in 1991 and to professor in 1997. In 1998, she was awarded as the Thomas J. Bannan Endowed Chair in Engineering from the School of Science and Engineering at Seattle University. She has also been an affiliate faculty with the Department of Electrical Engineering at the University of Washington since 1989, and has served on the graduate faculty there since 1991. Currently, She is the chair of the Department of Computer Science at Queens College, the City University of New York. Her research areas include image processing, pattern recognition, document image understanding, document image database design, and performance evaluation of document image analysis, and recognition systems. Her most significant contribution to the field of document image analysis and recognition has been the leadership role she had in the design and creation of the three sets of document image databases: UW-I, UW-II, and UW-III. She has served as program committee member for several IEEE and IAPR conferences and workshops. She also helped lead the first (1995), the second (1997), and the third (1999) International Graphic Recognition System Contests on Engineering drawings. She is currently the chairpersson of the IAPR technical committee on performance evaluation. She is a senior member of IEEE and a member of the IEEE Computer Society.

Robert M. Haralick is a distinguished professor in the Department of Computer Science, Graduate Center, City University of New York. He is responsible for developing the gray scale cooccurrence texture analysis technique and the facet model technique for image processing. In high level computer vision, he has worked on robust methods for photogrammetry and developed fast algorithms for solving the consistent labeling problem. He has developed shape analysis and extraction techniques using mathematical morphology, the morphological sampling theorem, and fast recursive morphology algorithms. In the area of document image understanding, Dr. Haralick, along with Dr. I. Phillips, developed a comprehensive ground-truthed set of some 1,600 document image pages most in English and some 200 pages in Japanese. He has also developed algorithms for document image skew angle estimation, zone delineation, and word and text line bounding box delineation. His most recent research is in the area of computer vision performance characterization and covariance propagation. He is a fellow of IEEE for his contributions in computer vision and image processing and a fellow of IAPR for his contributions in pattern recognition, image processing, and for service to IAPR. He has published more than 500 papers and recently completed his term as the president of the International Association for Pattern Recognition.