Pattern Analysis & Applications (2000)3:196–208
© 2000 Springer-Verlag London Limited

Consistent Partition and Labelling of Text Blocks

J. Liang[1], I. T. Phillips[2] and R. M. Haralick[3]

[1]*MathSoft, Inc., Seattle, WA;* [2]*Department of Computer Science/Software Engineering, Seattle University, Seattle, WA;* [3]*Department of Electrical Engineering, University of Washington, Seattle, WA, USA*

Abstract: This paper presents a text block extraction algorithm that takes as its input a set of text lines of a given document, and partitions the text lines into a set of text blocks, where each text block is associated with a set of homogeneous formatting attributes, e.g. text-alignment, indentation. The text block extraction algorithm described in this paper is probability based. We adopt an engineering approach to systematically characterising the text block structures based on a large document image database, and develop statistical methods to extract the text block structures from the image. All the probabilities are estimated from an extensive training set of various kinds of measurements among the text lines, and among the text blocks in the training data set. The off-line probabilities estimated in the training then drive all decisions in the on-line text block extraction. An iterative, relaxation-like method is used to find the partitioning solution that maximizes the joint probability. To evaluate the performance of our text block extraction algorithm, we used a three-fold validation method and developed a quantitative performance measure. The algorithm was evaluated on the UW-III database of some 1600 scanned document image pages. The text block extraction algorithm identifies and segments 91% of text blocks correctly.

Keywords: Document structure; Hidden Markov Model; Layout analysis; Statistical-based; Text block extraction; UW-III database

1. INTRODUCTION

A document structure analysis system converts a scanned document page or a document encoded by a Page Description Language (PDL), such as PostScript and Portable Document Format (PDF), into a well partitioned hierarchical representation that reliably identifies the basic document components and text blocks. This paper presents a text block extraction algorithm that takes as its input a set of text lines of a given document, and partitions the text lines into a set of text blocks, where each text block has homogeneous formatting attributes, e.g. text-alignment, indentation. The resulting block structures can be easily converted into a document markup language, such as a SGML or HTML file.

Most document structure analysis and understanding systems [1–13] contain a text line extraction and a text block extraction module. The performance of their text line extraction modules are much better than their text block extraction modules. The reason is simple. A text line extraction algorithm needs only to consider very restricted local information, such as text line directions, text line endings and spacings among the characters within the same text line. Whereas for a text block extraction algorithm, not only does it need to consider the local information, but also the global formatting of the document. Most of these algorithms are rule-driven, grammar-based or model-based. All these systems assume some prior knowledge of the typographical and layout conventions of document. The required knowledge about the document properties ranges from very specific and precise to fairly general ideas. One way of knowledge acquisition is to manually generate 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 hand-crafted rules tend to be brittle and only work for a specific kind of document. This means that these algorithms analyse successfully only those documents having previously defined text block formatting properties, and may fail on documents that do not meet the formatting criteria. Moreover, most of the rule or grammar-based algorithms use fixed, global distance thresholds for merging text lines into text blocks. These thresholds are either *ad hoc* given or empirically determined. However, the measurements made on the document entities

Received: 17 May 1999
Received in revised form: 7 February 2000
Accepted: 25 February 2000

may have errors, and the knowledge about the document style may be ambiguous. 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.

To build an error-tolerant and robust text block extraction algorithm, it is necessary to replace heuristics by systematics and rely on mathematical optimisation rather than intuition. Only the algorithm of Chen [5] takes a systematic approach to integrating the evidence sources they use. Their algorithm computes Bayesian estimates of text lines and text blocks based on parametric models. However, their algorithm's underlying models ignore the fact that some of the parameters are dependent on the text's font size. Thus, their algorithm performs poorly for data that contains documents of a wide variety of font types and font sizes. Their algorithm reports a 76% accuracy [14] on the same test data set that our algorithm described in this paper is tested on.

The text block extraction algorithm described in this paper is probability based. We adopt an engineering approach to systematically characterising the text block structures based on a large document image database, and develop statistical methods to extract the text block structures from the image. All the probabilities are estimated from an extensive training set of various kinds of measurements among the text lines and among the text blocks which the algorithm is based on. The off-line probabilities estimated in the training then drive all decisions in the online text block extraction. An iterative, relaxation-like method is used to find the partitioning solution that maximises the joint probability.

To evaluate the performance of our text block extraction algorithm, we used a three-fold cross-validation method, and developed a quantitative performance measure. The algorithm was evaluated on the UW-III database of some 1600 scanned document image pages. The text block extraction algorithm identifies and segments 91% of text blocks correctly.

This paper is organised as follows. In Section 2, a literature review is given. In Section 3, the problem of consistent partition and labelling of text blocks is formulated. In Section 4, our text block extraction algorithm is described. In Section 5, the experimental protocol for the off-line probabilities estimation is outlined. In Section 6, the criteria and measurements for the performance evaluation of our algorithm are presented. In Section 7, the experimental results of the algorithm are reported. Section 8 gives summary of the paper.

2. LITERATURE REVIEW

Most of the current document image analysis algorithms in the literature can be categorised 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 classes by the direction the algorithms work to construct the document hierarchy: bottom-up and top-down. Although there are algorithms that use a mixture of the two (hybrid), 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 on. 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. A classic example of this approach is Nagy's X-Y tree [2].

By the strategies or methods that algorithms use in their decision making during the process of dividing or grouping the document entities, the algorithms can be classified into two major categories: (1) rule-based/grammar-driven; and (2) statistical-based (either parametric or non parametric). 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. The *ad hoc* rules or the syntax of the grammar can also be very domain-specific. As for the statistical-based algorithms, the required free parameters that are used in the process are obtained via an off-line training processes. The estimated parameters are used in the decisions which govern the processes of the dividing or the grouping of the document entities.

The following are a handful of selected algorithms within the above defined categories. Only Chen's text block extraction algorithm takes a similar approach as the algorithm described in this paper:

- Nagy and Seth [2] discuss a prototype of document image analysis system for technical journals. They integrate document layout analysis and commercial OCR to generate information for document browsing system. The specific knowledge of a predetermined layout convention for a specific family of publications are encoded before-hand as block grammars in the form of a tree. Documents that satisfy the postulated layout conventions are parsed into different components.
- Ittner and Baird [3] develop a system for isolating blocks, lines, words and symbols within images of machine-printed textual documents. Their algorithm is independent of language and writing system 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 [4] develop a system for partitioning raster images of business letters into logically labelled area items. It employs various knowledge resources utilising spatial and geometric characteristics about the formal style

of business letters. Due to the unsupervised learning of document model, the system allows the establishment of a specific decision tree classifier to generate the knowledge representation.

- Chen [5] 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 non-text 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 their homogeneity in height, width, leading and justification within text blocks.
- Ha et al [6,7] present an algorithm based on recursive cutting of connected component projection profiles for segmenting binary document images into zones; zones are classified as textual and non-textual; then the text zones are decomposed to text blocks, text lines and words. Empirically determined thresholds are used at each cutting step.
- Srihari et al [8] develop a postal automation system that locates and interprets the destination address blocks on letter mail pieces with a high success rate and 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 labelling hypothesis. The best candidate block is selected by verifying the consistency of labelling hypotheses by using spatial relations.
- Etemad et al [9] 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 multi-class membership assignments. Segmentation is performed by integrating soft local decision vectors to reduce their 'ambiguities'.
- Kise et al [10] 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 used to estimate the intercharacter and interline gaps without the use of domain-specific parameters to select the boundaries.
- Wang and Yagasaki [11] present a page segmentation method called block selection which segments the page image into categorised 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 [12] use the traditional bottom-up approach based on the connected-component extraction to efficiently implement page segmentation and region identification. A document model which preserves top-down generation information is proposed. This method is applicable to documents from various technical journals, and can accommodate moderate amounts of skew and noise.

3. CONSISTENT PARTITIONING AND LABELLING OF TEXT BLOCKS

Given a set of text lines, the problem is to partition the text lines into a set of text blocks, each block having homogeneous leading and text alignment, and the attributes between neighbouring blocks being similar.

Let $\mathcal{A}$ be the set of input text lines. Let Π be a *partition* of $\mathcal{A}$ such that each element in Π is a text block. A system Π of nonempty sets is called a partition of $\mathcal{A}$ if

1. Π is a system of mutually disjoint sets, i.e. if $C \in \Pi$, $D \in \Pi$ and $C \neq D$, then $C \cap D = \emptyset$;
2. the union of the sets in Π is the whole set $\mathcal{A}$, i.e. $\cup_{c \in \Pi} = \mathcal{A}$.

Let L be a set of labels, such as text-alignment, indentation, and so on, that can be assigned to elements of the partition. Function f: $\Pi \rightarrow L$ associates each element of Π with a label. V: $\wp(\mathcal{A}) \rightarrow \Lambda$ specifies measurement made on subset of $\mathcal{A}$, i.e. a group of text lines, where Λ is the measurement space.

The consistent partition and labelling problem can be formulated as follows: *Given the initial set $\mathcal{A}$, find a partition Π of $\mathcal{A}$, and a labelling function f: $\Pi \rightarrow L$ that assigns each text block $\tau \in \Pi$ a label in L, that maximises the probability.*

$$\begin{aligned} &P(V(\tau)\text{: } \tau \in \Pi, f, \Pi|\mathcal{A}) \\ &=P(V(\tau)\text{: } \tau \in \Pi|\mathcal{A},\Pi,f)P(\Pi,f|\mathcal{A}) \qquad (1) \\ &= P(V(\tau)\text{: } \tau \in \Pi|\mathcal{A},\Pi,f)P(f|\Pi,\mathcal{A})P(\Pi|\mathcal{A}) \end{aligned}$$

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 Eq. (1) into

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

A brute-force method for finding the optimal solution for the above equation is to search through all possible partitions with all possible labels, and select the configuration which produces the highest conditional probability in Eq. (2). The number of partitions of an n-set is called a Bell number. Fortunately, the entities on a printed document are usually aligned with certain reading order. For example, text lines within a text block are within a proximity of one another. A text line on the top of the document never needs to be hypothesised as grouped with those text lines on the bottom of the document. Thus, with the ordering constraint, the partitioning and labeling problem can be re-formulated.

Let $A = (A_1, A_2, \ldots, A_N)$ be a partially ordered set where elements in A satisfies a partial ordering relation, *ReadBefore*(a, b) (is to be interpretated as a is read before b). The binary relation *ReadBefore* is a partial ordering relation, since if $A_i \in A$ and $A_j \in A$, then

1. *ReadBefore*(A_i, A_i) is false.
2. If *ReadBefore*$(A_i, A_j) \rightarrow$ *ReadBefore*(A_j, A_i) is false.
3. If *ReadBefore*(A_i, A_j) and *ReadBefore*$(A_j, A_k) \rightarrow$ *ReadBefore*(A_i, A_k).

Let $\mathcal{G} = \{Yes, No\}$ be the set of grouping labels. Let $A^p \subset A \times A$ be a set of adjacent pairs of text lines, such that $A^p = \{(A_i, A_j) | A_i, A_j \in A, j = i + 1$, and *ReadBefore*$(A_i, A_j) = True\}$. Grouping function, g: $A^p \rightarrow \mathcal{G}$, associates each pair of adjacent text lines of A with a grouping label (Y or N), where $g(i) = g(A_i, A_{i+1})$. Note that corresponding to each grouping function is a partition. Then, the partition probability $P(\Pi|A)$ can be reduced to computing the probability, $P(g|A)$, and $P(g|A)$ can be expanded as follows:

$$\begin{aligned} P(\Pi|A) &= P(g|A) \\ &= P(g(1), \ldots, g(N-1)|A_1, \ldots, A_N) \\ &= P(g(1)|A_1, A_2) \times \ldots \times P(g(N-1)|A_{N-1}, A_N) \\ &= \prod_{i \in S} P(g(i)|A_i, A_{i+1}), \end{aligned} \quad (3)$$

where $S = \{j | 1 \le j \le N - 1$, *ReadBefore*$(A_j, A_{j+1}) = True\}$. Therefore, the joint probability in Eq. (1) can be further decomposed as

$$P(V(\tau){:}\ \tau \in \Pi, f, \Pi | A)$$

$$= \prod_{\tau \in \Pi} P(V)(\tau)|f(\tau))P(f|\Pi, A) \prod_{i \in s} P(g(i)|A_i, A_{i+1}) \quad (4)$$

Again, in principle, we want to find the joint pair of functions (g, f) that maximises the conditional probabilities in Eq. (4). Such a search could be done by brute-force. Each of the 2^{N-1} different g functions determines a partition. Once a partition is given, the labeling function f is determined by maximising the probability $\Pi_{\tau \in \Pi} P(V(\tau)|f(\tau))P(f|\Pi, A)$.

To avoid the exponential search in the space of 2^{N-1}, where N is the number of input text lines, we settle for a search that finds a local maximum. The next section describes an iterative search method of order $O(N)$ that finds the consistent partition and labelling by monotonically maximising the joint probability in Eq. (4).

4. TEXT BLOCK EXTRACTION ALGORITHM

An iterative search method is developed to find a consistent partition and labeling solution that maximises the joint probability in Eq. (4). First, the set of input text lines A is arranged to a partially ordered set. Then, for each $A_i \in A$ such that *ReadBefore*(A_i, A_{i+1}) is true, we compute the grouping probability $P(g(i)|A_i, A_{i+1})$ for the pair of text lines A_i and A_{i+1}, by observing the spatial relationship between the pair.

An initial partition is determined based on the conditional grouping probabilities, $P(g(i)|A_i, A_{i+1})$. Then, we adjust the partition and assign labels to the members of the partition, by maximising the labelling and the label context probability $\Pi_{\tau \in \Pi} P(V(\tau)|f(\tau))P(f|\Pi, A)$. At each iteration, the adjustment that produces the maximum improvement of the joint probability in Eq. (4) is selected. The iteration stops when there is no improvement on the joint probability. Given an n-set, the size of the search space is therefore decreased to $t \times (n - 1)$, where t is the number of iterations. Figure 1 gives an overview of the processing steps of the text block extraction algorithm. Algorithm 1 presents a detailed description of the algorithm.

Algorithm 1 *Extract text bock from text lines*

1. Compute local grouping probabilities.
   For each $A_i \in A$, we search for the text line 'right below' it and rearrange elements in A according to the *ReadBefore* partial relation. Given the observed spatial relationships between a pair of text lines (A_i, A_{i+1}), we compute the probability that they are within the same text block:

   $P(g(i)|A_i, A_{i+1})$
2. Group text lines
   Given the linking probability $P(g(i))$ between a pair of adjacent text lines A_i and A_{i+1}, if $P(g(i) = Y) > P(g(i) = N)$, we group the pair into a text block; otherwise, A_i and A_{i+1} are in different blocks. During the initial par-

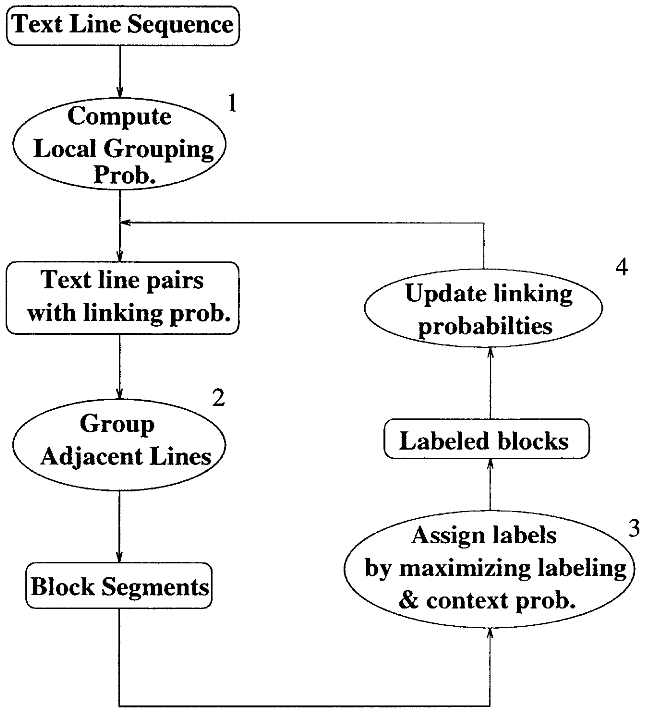

Text Line Sequence
Compute Local Grouping Prob.
1
Text line pairs with linking prob.
Update linking probabilties
4
Group Adjacent Lines
2
Labeled blocks
Assign labels by maximizing labeling & context prob.
3
Block Segments

Fig. 1. The process of text block grouping and labelling.

tition, $P(g(i)) = P(g(i)|A_i, A_{i+1})$, and this step yields our initial text block set, $B = \{B_1, B_2, \ldots, B_K\}$. The details of this step are given in Section 4.1.

3. Label text blocks
In this step, we compute the probability of an extracted block B_k having the homogeneous properties of a text block:
$$P(V(B_k)|f(B_k))$$
where the labelling $f(B_k)$ includes homogeneous leading and the text alignment type. The text blocks within the same neighbourhood usually have similar alignment type. The context constraint $P(f|B, A)$ is modeled with a Markov chain. Given a sequence of extracted text blocks, we determine labelling f by maximising
$$P_{label} = \prod_{B_k \in B} P(V(B_k)|f(B_k))P(f|B, A) \quad (5)$$
A detailed description of computation of the homogeneous leading, the alignment, and the context constraint probability are given in Sections 4.2, 4.3, and 4.4, respectively.
4. Update linking probability and adjust partition
Given the computed labelling probabilities, we update the linking probability $P(g(i))$, between each pair of adjacent text lines:
$$P(g(i)) \propto P(g(i)|A_i, A_{i+1})P_{label} \quad (6)$$
During each iteration, the adjustment which produces the maximum improvement of the linking probability is selected. Then, we continue to Step 2, and adjust the partition according to the updated linking probabilities. If there is no improvement on the linking probability, we stop the iteration and return the extracted text blocks.

Figure 2 shows an example of a set of input text lines on a document image. Extracted text blocks are illustrated in Fig. 3.

4.1. Group Text Blocks

Figure 4 illustrates a pair of adjacent text lines. Leading is the distance between baseline of the top text line and the x-height line of the bottom text line.

For each pair of vertically adjacent text lines A_i and A_{i+1}, where the text line is represented by a bounding box (x, y_b, w, h_x) (y_b is the coordinate of the baseline and h_x is the x-height), we make the following measurements (see Fig. 5):

- x-height: h_i and h_{i+1}
- inter-line spacing: $d(i, i+1)$
- horizontal overlap: $o(i, i+1)$
- left edge offset: $e_l(i, i+1) = x_i - x_{i+1}$
- centre edge offset: $e_c(i, i+1) = x_i - x_{i+1} + (w_i - w_{i+1})/2$
- right edge offset: $e_r(i, i+1) = x_i - x_{i+1} + w_i - w_{i+1}$

The interline spacing is normalised by the text lines' x-height average,
$$d_i = \frac{d(i, i+1)}{\frac{1}{2}(h_i + h_{i+1})}$$
We use ratio of two text lines' x-height to measure the difference between their font size,
$$x_i = \frac{\min(h_i, h_j)}{\max(h_i, h_j)}$$
The horizontal overlap between A_i and A_{i+1} is normalised by h_i and h_{i+1}, respectively,
$$o_i = \frac{o(i, i+1)}{h_i} \quad \text{and} \quad o_{i+1} = \frac{o(i, i+1)}{h_{i+1}}$$
The relative location between the left edges of A_i and A_{i+1} is defined as
$$rl_i = \begin{cases} 0 & \text{if } |e_l(i, i+1)| < \frac{1}{2}(h_i + h_{i+1}) \\ 1 & \text{if } e_l(i, i+1) \geq \frac{1}{2}(h_i + h_{i+1}) \\ -1 & \text{if } e_l(i, i+1) \leq -\frac{1}{2}(h_i + h_{i+1}) \end{cases}$$
The relative location between the centre edges of A_i and A_{i+1} and the relative location between the right edges of A_i and A_{i+1} are defined in a similar way.

Given the above measurements, we compute the probability that A_i and A_{i+1} belong to the same block:
$$P(g(i)|x_i, o_i, o_{i+1}, d_i, rl_i, rc_i, rr_i) \quad (7)$$

4.2. Homogeneous Text Block Properties

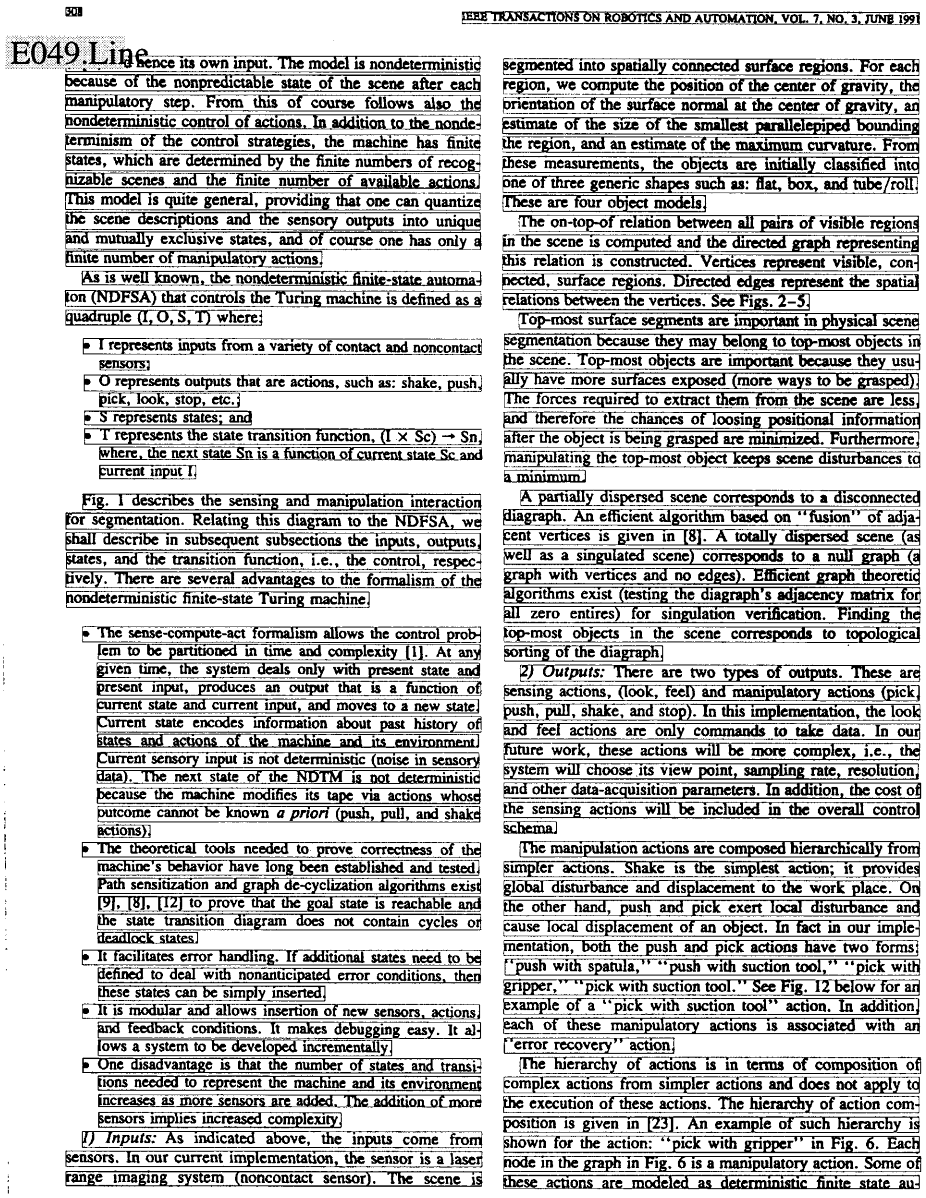
A text block usually has homogeneous inter-line spacing, and certain alignment type (justification, indentation and hanging). Given a detected text block B, we compute the probability that B has homogeneous leading, and certain type of text alignment,
$$P(V(B)|\text{TextBlock}(B)) = P(V(B)|\text{Leading}(B), \text{Alignment}(B))$$
In this section, we describe the estimation of homogeneous leading. The text alignment detection method is given in next section.

Let $B = (A_1, \ldots, A_n)$ be an extracted text block. $D_B = (d(1, 2), \ldots, d(n-1, n))$ is a sequence of inter-line spaces, where $d(j, j+1)$ is the space between A_j and A_{j+1}. We compute the median and the maximum value of the elements of D_B. The probability of a text block B having homogeneous leading is estimated as
$$P(\text{median}(D_B), \max(D_B)|\text{Leading}(B)) \quad (8)$$

4.3. Text Alignment Detection

Figure 6 illustrates four homogeneous types of text block alignment. The various types of indentation are shown in Fig. 7. Other types of alignment, such as the justified-hanging and left-hanging, can also appear in some document images (see Fig. 8).

A justification and indentation label is assigned to each possible block using the model shown in Fig. 9. Given a text block B that consists of a group of text lines $B = (A_1, A_2, \ldots, A_n)$, we determine the text alignment of B by observing the alignment of the text line edges (see Fig. 10).



302 IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, VOL. 7, NO. 3, JUNE 1991

E049.Line

hence its own input. The model is nondeterministic because of the nonpredictable state of the scene after each manipulatory step. From this of course follows also the nondeterministic control of actions. In addition to the nondeterminism of the control strategies, the machine has finite states, which are determined by the finite numbers of recognizable scenes and the finite number of available actions. This model is quite general, providing that one can quantize the scene descriptions and the sensory outputs into unique and mutually exclusive states, and of course one has only a finite number of manipulatory actions.

As is well known, the nondeterministic finite-state automaton (NDFSA) that controls the Turing machine is defined as a quadruple (I, O, S, T) where:

- I represents inputs from a variety of contact and noncontact sensors;
- O represents outputs that are actions, such as: shake, push, pick, look, stop, etc.;
- S represents states; and
- T represents the state transition function, (I × Sc) → Sn, where, the next state Sn is a function of current state Sc and current input I.

Fig. 1 describes the sensing and manipulation interaction for segmentation. Relating this diagram to the NDFSA, we shall describe in subsequent subsections the inputs, outputs, states, and the transition function, i.e., the control, respectively. There are several advantages to the formalism of the nondeterministic finite-state Turing machine.

- The sense-compute-act formalism allows the control problem to be partitioned in time and complexity [1]. At any given time, the system deals only with present state and present input, produces an output that is a function of current state and current input, and moves to a new state. Current state encodes information about past history of states and actions of the machine and its environment. Current sensory input is not deterministic (noise in sensory data). The next state of the NDTM is not deterministic because the machine modifies its tape via actions whose outcome cannot be known *a priori* (push, pull, and shake actions).
- The theoretical tools needed to prove correctness of the machine's behavior have long been established and tested. Path sensitization and graph de-cyclization algorithms exist [9], [8], [12] to prove that the goal state is reachable and the state transition diagram does not contain cycles or deadlock states.
- It facilitates error handling. If additional states need to be defined to deal with nonanticipated error conditions, then these states can be simply inserted.
- It is modular and allows insertion of new sensors, actions, and feedback conditions. It makes debugging easy. It allows a system to be developed incrementally.
- One disadvantage is that the number of states and transitions needed to represent the machine and its environment increases as more sensors are added. The addition of more sensors implies increased complexity.

1) Inputs: As indicated above, the inputs come from sensors. In our current implementation, the sensor is a laser range imaging system (noncontact sensor). The scene is segmented into spatially connected surface regions. For each region, we compute the position of the center of gravity, the orientation of the surface normal at the center of gravity, an estimate of the size of the smallest parallelepiped bounding the region, and an estimate of the maximum curvature. From these measurements, the objects are initially classified into one of three generic shapes such as: flat, box, and tube/roll. These are four object models.

The on-top-of relation between all pairs of visible regions in the scene is computed and the directed graph representing this relation is constructed. Vertices represent visible, connected, surface regions. Directed edges represent the spatial relations between the vertices. See Figs. 2–5.

Top-most surface segments are important in physical scene segmentation because they may belong to top-most objects in the scene. Top-most objects are important because they usually have more surfaces exposed (more ways to be grasped). The forces required to extract them from the scene are less, and therefore the chances of loosing positional information after the object is being grasped are minimized. Furthermore, manipulating the top-most object keeps scene disturbances to a minimum.

A partially dispersed scene corresponds to a disconnected diagraph. An efficient algorithm based on "fusion" of adjacent vertices is given in [8]. A totally dispersed scene (as well as a singulated scene) corresponds to a null graph (a graph with vertices and no edges). Efficient graph theoretic algorithms exist (testing the diagraph's adjacency matrix for all zero entires) for singulation verification. Finding the top-most objects in the scene corresponds to topological sorting of the diagraph.

2) Outputs: There are two types of outputs. These are sensing actions, (look, feel) and manipulatory actions (pick, push, pull, shake, and stop). In this implementation, the look and feel actions are only commands to take data. In our future work, these actions will be more complex, i.e., the system will choose its view point, sampling rate, resolution, and other data-acquisition parameters. In addition, the cost of the sensing actions will be included in the overall control schema.

The manipulation actions are composed hierarchically from simpler actions. Shake is the simplest action; it provides global disturbance and displacement to the work place. On the other hand, push and pick exert local disturbance and cause local displacement of an object. In fact in our implementation, both the push and pick actions have two forms: "push with spatula," "push with suction tool," "pick with gripper," "pick with suction tool." See Fig. 12 below for an example of a "pick with suction tool" action. In addition, each of these manipulatory actions is associated with an "error recovery" action.

The hierarchy of actions is in terms of composition of complex actions from simpler actions and does not apply to the execution of these actions. The hierarchy of action composition is given in [23]. An example of such hierarchy is shown for the action: "pick with gripper" in Fig. 6. Each node in the graph in Fig. 6 is a manipulatory action. Some of these actions are modeled as deterministic finite state au-

Fig. 2. A real document image overlaid with the bounding boxes of input text lines.

Let e_{li} be the left edge of the text line A_i and let e_{ci} and e_{ri} be the centre and right edges of the line box, respectively. Let E_l be the left edges of text line 2 to n, such that $E_l = \{e_{li} | 2 \leq i \leq n\}$. E_c is the centre edges of text line 2 to $n - 1$, and E_r is the right edges of text line 1 to $n - 1$. We first estimate the median of E_l, then compute the absolute deviation D_l of the elements of E_l from its median,

$$D_l = \{d_i | d_i = |e_{li} - \text{median}(E_l)|, \ 2 \leq i \leq n\}$$

Similarly, we estimate the absolute deviation of centre edges and right edges: D_c and D_r. Then, we compute the probability of B being left, centre, right, or both justified by observing the mean absolute deviation of left, centre and right edges,

$$P(\text{mean}(D_l), \text{mean}(D_c), \text{mean}\ (D_r) | J(B)) \tag{9}$$

where $J(B)$ is the justification type assigned to block B.

For each detected text block, we verify the consistency of text alignment, by checking the maximum deviation of text line edges from the corresponding edge of the text block,

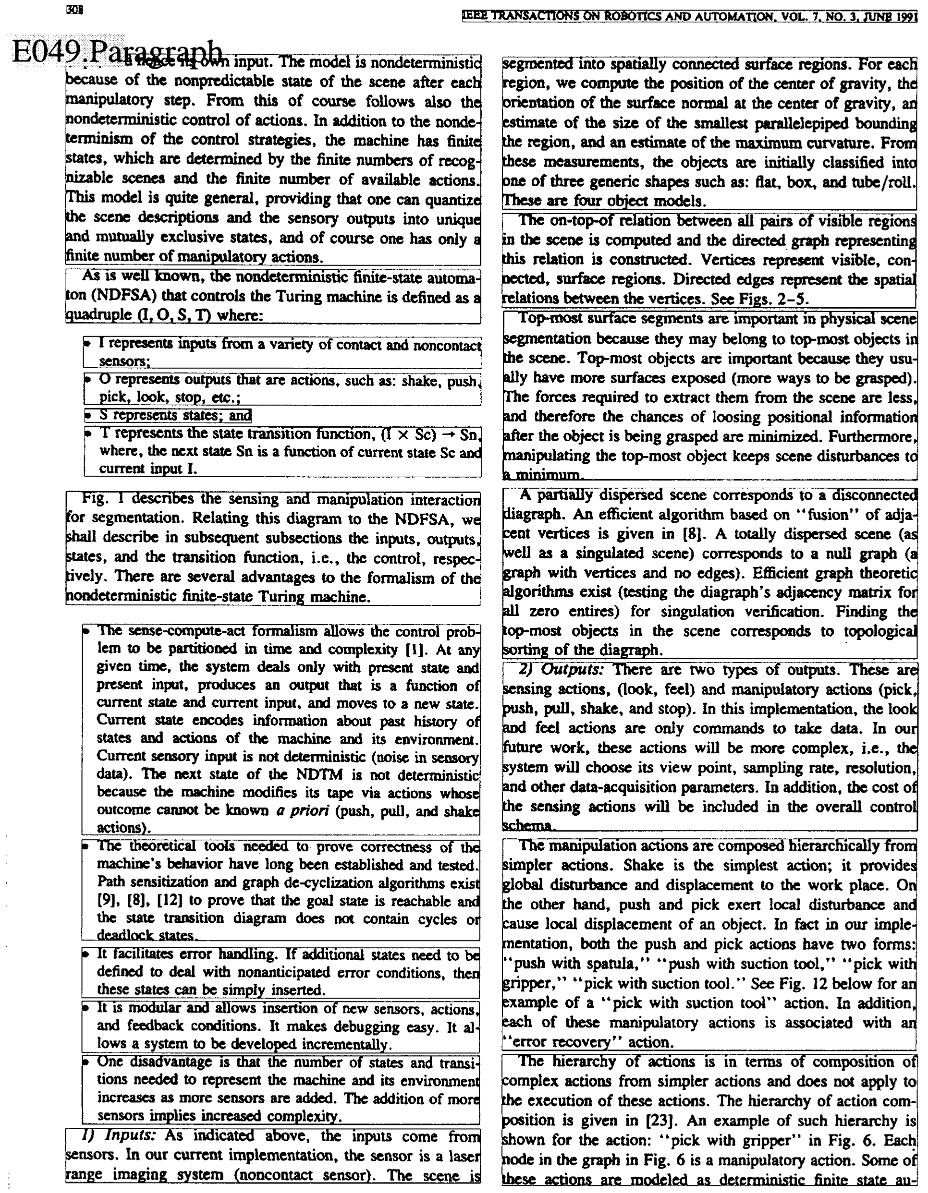
$$P(\max(D_l) | J(B) = \textit{left})$$

$$P(\max(D_c) | J(B) = \textit{centre})$$

$$P(\max(D_r) | J(B) = \textit{right})$$

$$P(\max(D_l, D_r) | J(B) = \textit{both}) \tag{10}$$

Finally, we determine if a left- or both-justified text block


E049.Paragraph

308 IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, VOL. 7, NO. 3, JUNE 1991

[illegible] own input. The model is nondeterministic because of the nonpredictable state of the scene after each manipulatory step. From this of course follows also the nondeterministic control of actions. In addition to the nondeterminism of the control strategies, the machine has finite states, which are determined by the finite numbers of recognizable scenes and the finite number of available actions. This model is quite general, providing that one can quantize the scene descriptions and the sensory outputs into unique and mutually exclusive states, and of course one has only a finite number of manipulatory actions.

As is well known, the nondeterministic finite-state automaton (NDFSA) that controls the Turing machine is defined as a quadruple (I, O, S, T) where:

- I represents inputs from a variety of contact and noncontact sensors;
- O represents outputs that are actions, such as: shake, push, pick, look, stop, etc.;
- S represents states; and
- T represents the state transition function, (I × Sc) → Sn, where, the next state Sn is a function of current state Sc and current input I.

Fig. 1 describes the sensing and manipulation interaction for segmentation. Relating this diagram to the NDFSA, we shall describe in subsequent subsections the inputs, outputs, states, and the transition function, i.e., the control, respectively. There are several advantages to the formalism of the nondeterministic finite-state Turing machine.

- The sense-compute-act formalism allows the control problem to be partitioned in time and complexity [1]. At any given time, the system deals only with present state and present input, produces an output that is a function of current state and current input, and moves to a new state. Current state encodes information about past history of states and actions of the machine and its environment. Current sensory input is not deterministic (noise in sensory data). The next state of the NDTM is not deterministic because the machine modifies its tape via actions whose outcome cannot be known *a priori* (push, pull, and shake actions).
- The theoretical tools needed to prove correctness of the machine's behavior have long been established and tested. Path sensitization and graph de-cyclization algorithms exist [9], [8], [12] to prove that the goal state is reachable and the state transition diagram does not contain cycles or deadlock states.
- It facilitates error handling. If additional states need to be defined to deal with nonanticipated error conditions, then these states can be simply inserted.
- It is modular and allows insertion of new sensors, actions, and feedback conditions. It makes debugging easy. It allows a system to be developed incrementally.
- One disadvantage is that the number of states and transitions needed to represent the machine and its environment increases as more sensors are added. The addition of more sensors implies increased complexity.

1) Inputs: As indicated above, the inputs come from sensors. In our current implementation, the sensor is a laser range imaging system (noncontact sensor). The scene is segmented into spatially connected surface regions. For each region, we compute the position of the center of gravity, the orientation of the surface normal at the center of gravity, an estimate of the size of the smallest parallelepiped bounding the region, and an estimate of the maximum curvature. From these measurements, the objects are initially classified into one of three generic shapes such as: flat, box, and tube/roll. These are four object models.

The on-top-of relation between all pairs of visible regions in the scene is computed and the directed graph representing this relation is constructed. Vertices represent visible, connected, surface regions. Directed edges represent the spatial relations between the vertices. See Figs. 2–5.

Top-most surface segments are important in physical scene segmentation because they may belong to top-most objects in the scene. Top-most objects are important because they usually have more surfaces exposed (more ways to be grasped). The forces required to extract them from the scene are less, and therefore the chances of loosing positional information after the object is being grasped are minimized. Furthermore, manipulating the top-most object keeps scene disturbances to a minimum.

A partially dispersed scene corresponds to a disconnected diagraph. An efficient algorithm based on "fusion" of adjacent vertices is given in [8]. A totally dispersed scene (as well as a singulated scene) corresponds to a null graph (a graph with vertices and no edges). Efficient graph theoretic algorithms exist (testing the diagraph's adjacency matrix for all zero entires) for singulation verification. Finding the top-most objects in the scene corresponds to topological sorting of the diagraph.

2) Outputs: There are two types of outputs. These are sensing actions, (look, feel) and manipulatory actions (pick, push, pull, shake, and stop). In this implementation, the look and feel actions are only commands to take data. In our future work, these actions will be more complex, i.e., the system will choose its view point, sampling rate, resolution, and other data-acquisition parameters. In addition, the cost of the sensing actions will be included in the overall control schema.

The manipulation actions are composed hierarchically from simpler actions. Shake is the simplest action; it provides global disturbance and displacement to the work place. On the other hand, push and pick exert local disturbance and cause local displacement of an object. In fact in our implementation, both the push and pick actions have two forms: "push with spatula," "push with suction tool," "pick with gripper," "pick with suction tool." See Fig. 12 below for an example of a "pick with suction tool" action. In addition, each of these manipulatory actions is associated with an "error recovery" action.

The hierarchy of actions is in terms of composition of complex actions from simpler actions and does not apply to the execution of these actions. The hierarchy of action composition is given in [23]. An example of such hierarchy is shown for the action: "pick with gripper" in Fig. 6. Each node in the graph in Fig. 6 is a manipulatory action. Some of these actions are modeled as deterministic finite state au-

Fig. 3. A real document image overlaid with the bounding boxes of extracted text blocks.

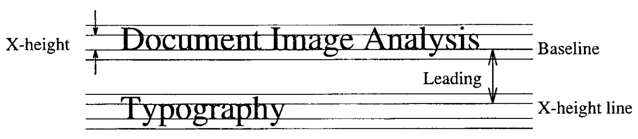


Fig. 4. A pair of adjacent text lines and their spatial relationships.

has left hanging on the first text line. Let D_{l1} be the distance from e_{l1}, the left edge of the first line, to the median of E_l, the hanging probability is

$$P(D_{l1}|H(B)) \tag{11}$$

where $H(B)$ is the hanging label assigned to block B.

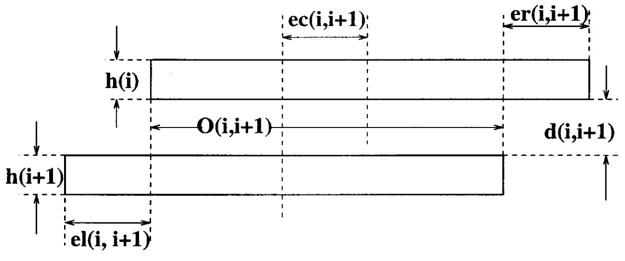


Fig. 5. The measurements made on a pair of adjacent text lines.

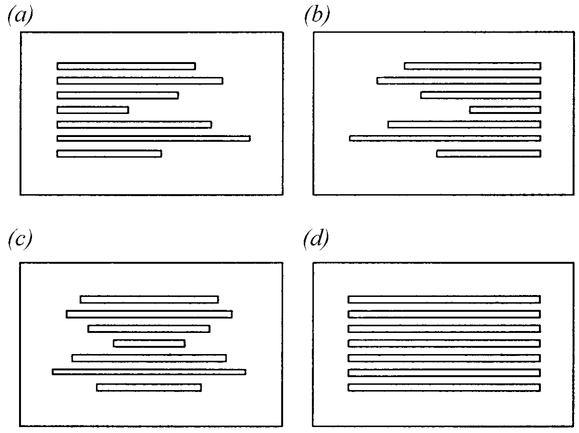


Fig. 6. Four types of text justification. (a) Left justified; (b) right justified; (c) centre justified; and (d) justified.

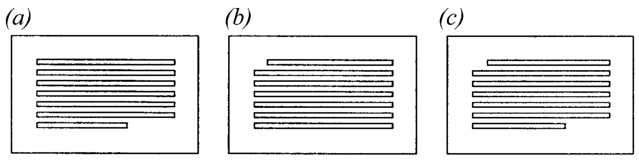


Fig. 7. The various types of indentation. (a) Indentation at the last line; (b) indentation at the first line; and (c) indentation at the first and the last lines.

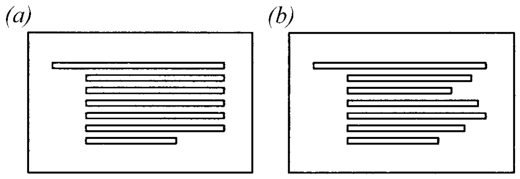


Fig. 8. The hanging type alignment. (a) Justified-hanging; (b) left-hanging.

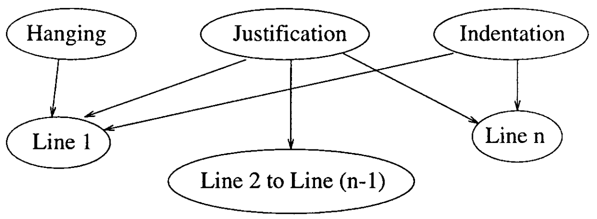


Fig. 9. The process that determines the alignment type of a text block.

4.4. Context Consistency of Text Alignment

Given a sequence of hypothesised text blocks $B = (B_1, B_2, \ldots, B_M)$, we use a Markov chain model to represent the context constraint of the sequence of alignment types $f = f(B_1), f(B_2), \ldots, f(B_M)$. The probability $P(f|B, A)$ requires specification of the alignment of the current block, as well as the alignment types of all the predecessor blocks,

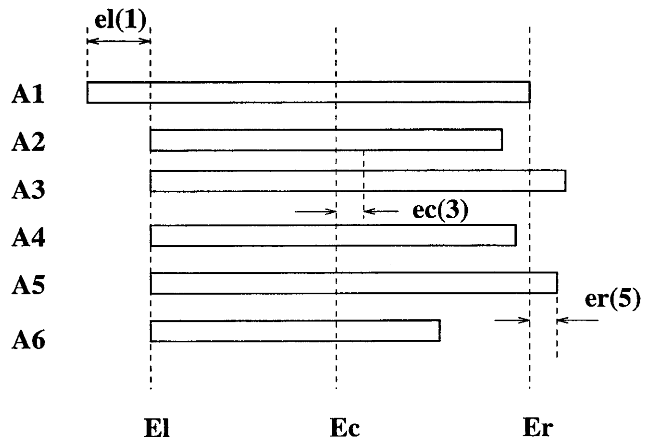


Fig. 10. The measurements made on determining the alignment type of a text block.

$$P(f|B,A) = \prod_{B_t \in B} P(f(B_t)|f(B_{t-1}), f(B_{t-2}), \ldots, f(B_1)) \quad (12)$$

where $P(f(B_t)|f(B_{t-1}), f(B_{t-2}), \ldots, f(B_1))$ is the probability of $f(B_t)$ under the condition that the alignment types of the previous text blocks are $f(B_{t-1}), f(B_{t-2}), \ldots, f(B_1)$. For a first-order Markov chain, the current state is only dependent on the previous state, i.e.

$$P(f(B_t)|f(B_{t-1}), f(B_{t-2}), \ldots, f(B_1)) = P(f(B_t)|f(B_{t-1})) \quad (13)$$

Given this assumption of conditional independence, the probability in Eq. (12) is simplified as

$$P(f|B, A) = \prod_{B_t \in B} P(f(B_t)|f(B_{t-1})) \quad (14)$$

Therefore, the labelling and context consistency probability

$$\prod_{B_t \in B} P(V(B_t)|f(B_t))P(f|B,A) \quad (15)$$

is actually a simple hidden Markov model, shown in Fig. 11. The reader is referred to Rabiner [17] for a tutorial on the Hidden Markov Models (HMM) and Aas et al [18] for the use of HMM in image analysis applications. In this case, the hidden states are the underlying alignment types $f(B)$ which are not observable. $P(f(B_t)|f(B_{t-1}), B_t \in B$, are called *transition probabilities*, and $P(f(B_1))$ are the *initial probabilities*. A sequence of observations $V(B)$ is measured at the text blocks $B_t \in B$, where $P(V(B_t)|f(B_t))$ are observation probabilities.

Given a certain partition of text blocks, finding the

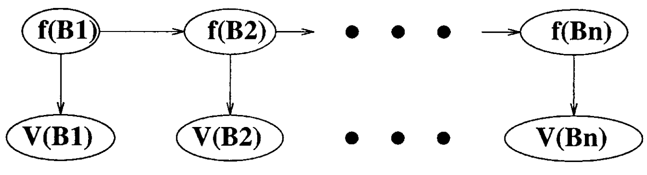


Fig. 11. A model of the text alignment consistency.

sequence of alignment types which maximises the probability (15) can be efficiently computed using the Viterbi algorithm [17]. The recursion formula for computing the highest probability along a single path, which ends at block B_t with the alignment type $f(B_t) = j$, is the following:

$$\delta_t(j) = \tag{16}$$

$$\begin{cases} P(f(B_1) = j)\ P(V(B_1)|f(B_1) = j) & t = 1 \\ [\max_i \delta_{t-1}(i) P(f(B_t) = j|f(B_{t-1}) = i)] P(V(B_t)|f(B_t) = j) & t > 1 \end{cases}$$

where i and j are the possible text alignment types.

The algorithm can be seen as an application of dynamic programming for finding a maximum probability path in a directed graph. We use array $\psi_t(j)$ to keep track of the arguments which maximised (16):

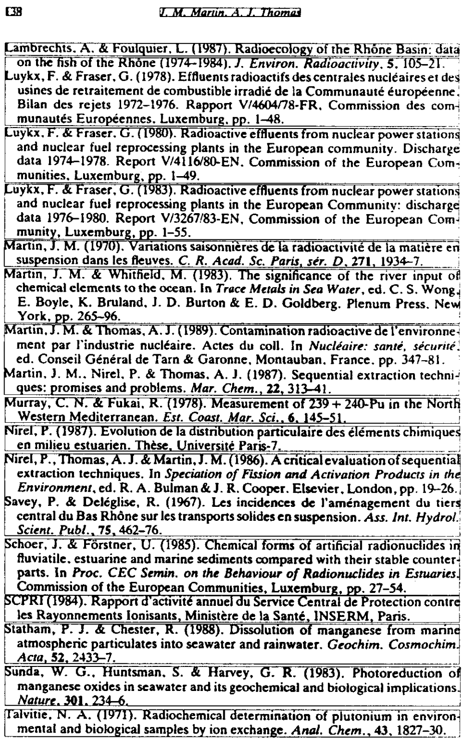
$$\psi_t(j) = \arg\max_i\ [\delta_{t-1}(i) P(f(B_t) = j|f(B_{t-1}) = i)] \tag{17}$$

At the end of the sequence, a backtracking step is performed to retrieve the most probable path of text alignment labels.

Figure 12 illustrates the extracted text blocks after the initial grouping based on local observations. The corrected text blocks by maximising the labelling and context probability are shown in Fig. 13.

5. ESTIMATION OF PROBABILITY DISTRIBUTIONS

The discrete contingency tables are used to represent the joint and conditional probabilities used in the algorithm.



138 J. M. Martin, A. J. Thomas

Lambrechts, A. & Foulquier, L. (1987). Radioecology of the Rhône Basin: data on the fish of the Rhône (1974–1984). *J. Environ. Radioactivity*, **5**, 105–21.

Luykx, F. & Fraser, G. (1978). Effluents radioactifs des centrales nucléaires et des usines de retraitement de combustible irradié de la Communauté éuropéenne. Bilan des rejets 1972–1976. Rapport V/4604/78-FR. Commission des communautés Européennes. Luxemburg, pp. 1–48.

Luykx, F. & Fraser, G. (1980). Radioactive effluents from nuclear power stations and nuclear fuel reprocessing plants in the European community. Discharge data 1974–1978. Report V/4116/80-EN. Commission of the European Communities, Luxemburg, pp. 1–49.

Luykx, F. & Fraser, G. (1983). Radioactive effluents from nuclear power stations and nuclear fuel reprocessing plants in the European Community: discharge data 1976–1980. Report V/3267/83-EN, Commission of the European Community, Luxemburg, pp. 1–55.

Martin, J. M. (1970). Variations saisonnières de la radioactivité de la matière en suspension dans les fleuves. *C. R. Acad. Sc. Paris, sér. D*, **271**, 1934–7.

Martin, J. M. & Whitfield, M. (1983). The significance of the river input of chemical elements to the ocean. In *Trace Metals in Sea Water*, ed. C. S. Wong, E. Boyle, K. Bruland, J. D. Burton & E. D. Goldberg. Plenum Press. New York, pp. 265–96.

Martin, J. M. & Thomas, A. J. (1989). Contamination radioactive de l'environnement par l'industrie nucléaire. Actes du coll. In *Nucléaire: santé, sécurité*, ed. Conseil Général de Tarn & Garonne, Montauban, France, pp. 347–81.

Martin, J. M., Nirel, P. & Thomas, A. J. (1987). Sequential extraction techniques: promises and problems. *Mar. Chem.*, **22**, 313–41.

Murray, C. N. & Fukai, R. (1978). Measurement of 239 + 240-Pu in the North Western Mediterranean. *Est. Coast. Mar. Sci.*, **6**, 145–51.

Nirel, P. (1987). Evolution de la distribution particulaire des éléments chimiques en milieu estuarien. Thèse, Université Paris-7.

Nirel, P., Thomas, A. J. & Martin, J. M. (1986). A critical evaluation of sequential extraction techniques. In *Speciation of Fission and Activation Products in the Environment*, ed. R. A. Bulman & J. R. Cooper. Elsevier, London, pp. 19–26.

Savey, P. & Deléglise, R. (1967). Les incidences de l'aménagement du tiers central du Bas Rhône sur les transports solides en suspension. *Ass. Int. Hydrol. Scient. Publ.*, **75**, 462–76.

Schoer, J. & Förstner, U. (1985). Chemical forms of artificial radionuclides in fluviatile, estuarine and marine sediments compared with their stable counterparts. In *Proc. CEC Semin. on the Behaviour of Radionuclides in Estuaries*. Commission of the European Communities, Luxemburg, pp. 27–54.

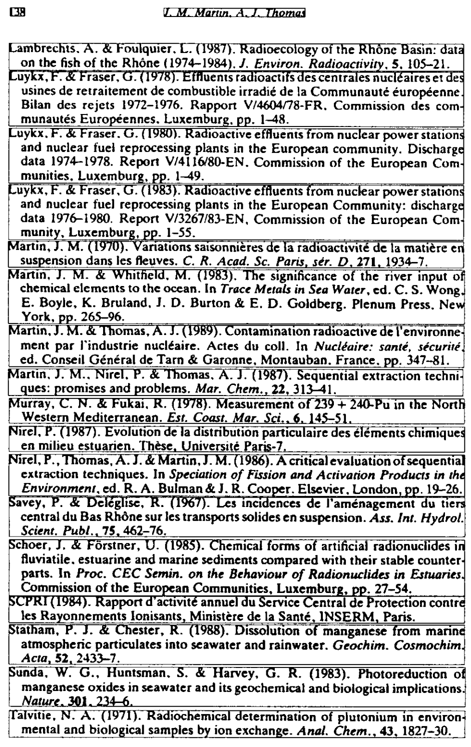
SCPRI (1984). Rapport d'activité annuel du Service Central de Protection contre les Rayonnements Ionisants, Ministère de la Santé, INSERM, Paris.

Statham, P. J. & Chester, R. (1988). Dissolution of manganese from marine atmospheric particulates into seawater and rainwater. *Geochim. Cosmochim. Acta*, **52**, 2433–7.

Sunda, W. G., Huntsman, S. & Harvey, G. R. (1983). Photoreduction of manganese oxides in seawater and its geochemical and biological implications. *Nature*, **301**, 234–6.

Talvitie, N. A. (1971). Radiochemical determination of plutonium in environmental and biological samples by ion exchange. *Anal. Chem.*, **43**, 1827–30.

Fig. 12. A document image overlaid with the bounding boxes of text blocks after initial grouping.



138 J. M. Martin, A. J. Thomas

Lambrechts, A. & Foulquier, L. (1987). Radioecology of the Rhône Basin: data on the fish of the Rhône (1974–1984). *J. Environ. Radioactivity*, **5**, 105–21.

Luykx, F. & Fraser, G. (1978). Effluents radioactifs des centrales nucléaires et des usines de retraitement de combustible irradié de la Communauté éuropéenne. Bilan des rejets 1972–1976. Rapport V/4604/78-FR. Commission des communautés Européennes. Luxemburg, pp. 1–48.

Luykx, F. & Fraser, G. (1980). Radioactive effluents from nuclear power stations and nuclear fuel reprocessing plants in the European community. Discharge data 1974–1978. Report V/4116/80-EN. Commission of the European Communities, Luxemburg, pp. 1–49.

Luykx, F. & Fraser, G. (1983). Radioactive effluents from nuclear power stations and nuclear fuel reprocessing plants in the European Community: discharge data 1976–1980. Report V/3267/83-EN, Commission of the European Community, Luxemburg, pp. 1–55.

Martin, J. M. (1970). Variations saisonnières de la radioactivité de la matière en suspension dans les fleuves. *C. R. Acad. Sc. Paris, sér. D*, **271**, 1934–7.

Martin, J. M. & Whitfield, M. (1983). The significance of the river input of chemical elements to the ocean. In *Trace Metals in Sea Water*, ed. C. S. Wong, E. Boyle, K. Bruland, J. D. Burton & E. D. Goldberg. Plenum Press. New York, pp. 265–96.

Martin, J. M. & Thomas, A. J. (1989). Contamination radioactive de l'environnement par l'industrie nucléaire. Actes du coll. In *Nucléaire: santé, sécurité*, ed. Conseil Général de Tarn & Garonne, Montauban, France, pp. 347–81.

Martin, J. M., Nirel, P. & Thomas, A. J. (1987). Sequential extraction techniques: promises and problems. *Mar. Chem.*, **22**, 313–41.

Murray, C. N. & Fukai, R. (1978). Measurement of 239 + 240-Pu in the North Western Mediterranean. *Est. Coast. Mar. Sci.*, **6**, 145–51.

Nirel, P. (1987). Evolution de la distribution particulaire des éléments chimiques en milieu estuarien. Thèse, Université Paris-7.

Nirel, P., Thomas, A. J. & Martin, J. M. (1986). A critical evaluation of sequential extraction techniques. In *Speciation of Fission and Activation Products in the Environment*, ed. R. A. Bulman & J. R. Cooper. Elsevier, London, pp. 19–26.

Savey, P. & Deléglise, R. (1967). Les incidences de l'aménagement du tiers central du Bas Rhône sur les transports solides en suspension. *Ass. Int. Hydrol. Scient. Publ.*, **75**, 462–76.

Schoer, J. & Förstner, U. (1985). Chemical forms of artificial radionuclides in fluviatile, estuarine and marine sediments compared with their stable counterparts. In *Proc. CEC Semin. on the Behaviour of Radionuclides in Estuaries*. Commission of the European Communities, Luxemburg, pp. 27–54.

SCPRI (1984). Rapport d'activité annuel du Service Central de Protection contre les Rayonnements Ionisants, Ministère de la Santé, INSERM, Paris.

Statham, P. J. & Chester, R. (1988). Dissolution of manganese from marine atmospheric particulates into seawater and rainwater. *Geochim. Cosmochim. Acta*, **52**, 2433–7.

Sunda, W. G., Huntsman, S. & Harvey, G. R. (1983). Photoreduction of manganese oxides in seawater and its geochemical and biological implications. *Nature*, **301**, 234–6.

Talvitie, N. A. (1971). Radiochemical determination of plutonium in environmental and biological samples by ion exchange. *Anal. Chem.*, **43**, 1827–30.

Fig. 13. A document image overlaid with the bounding boxes of text blocks after consistent labelling and partition adjustment.

Each variable of the table has a finite number of mutually exclusive states. If A is a variable with states $a_1, \ldots, a_n$, then $P(A)$ is a probability distribution over these states:

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

where x_i is the probability of A being in state a_i.

Rather than entering the value of each variable for each individual in the sample, cell count records, for each possible combination of values of the measured variables, how many members of the sample have exactly that combinations of values. A cell count is simply the number of units in the sample that have a given fixed set of values for the variables. The joint probability table can be computed directly from the cell count.

The steps for estimating the conditional and joint probability tables are as follows:

1. Determine the variables to observe.
2. Collect and record the data observations.
3. Study graphics and summaries of the collected data to reveal low-dimensional relationships between variables.
4. Choose a model describing the important relationships seen or hypothesised in the data.
5. Quantise the value of each variable into a finite number of mutually exclusive states.
6. Compute the cell count table from the data.

A tree structure quantisation is used to partition the value

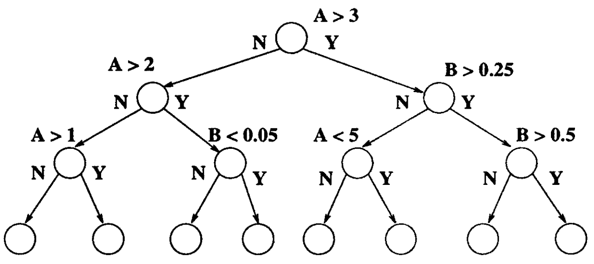


Fig. 14. A constructed tree-based model used for quantisation.

of each continuous 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 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. To construct the quantised table from the tree, one follows the path from the root to the leaves within 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 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 is shown in Fig. 14. 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 quantised into five bins and four bins, respectively. The constructed table with 20 cells is shown in Table 1.

The data set we selected to train and evaluate our text block extraction algorithm is the University of Washington English Document Image Database-III [15,16] (UW-III). The UW-III database contains 1600 skewcorrected English journal document images that come with manually edited ground-truth data. The ground-truth data includes bounding boxes for text blocks and the bounding boxes for text lines within the text blocks. Each text block in the data set is also associated with a set of formatting attributes, such as justification and hanging.

We conduct a series of experiments to empirically determine the probability distributions that we use to extract

Table 1. A cell count table, where the quantisation of variables is determined using the tree structure shown in Fig. 14

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

text blocks. The reader is referred to Liang [19] for the histogram and quantisation of each variable computed from the selected samples in the UW-III data set.

6. PERFORMANCE MEASURE

In general, the performance evaluation of any algorithm can be done by testing the algorithm on a selected testing data set and then comparing its results against the corresponding ground-truth of the test 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. Performance metrics need to be defined to measure the quantitative performance of the algorithm.

There are two problems in making the evaluation. The first is one of correspondence: which entities of the ground-truth set correspond to which entities of the automatically produced set. Once this correspondence is determined then a comparison of detected entities with the ground-truth entities can proceed.

Suppose we are given two sets $\mathcal{G} = \{G_1, G_2, \ldots, G_M\}$ for the ground-truthed entities and $\mathcal{D} = \{D_1, D_2, \ldots, D_N\}$ for the detected entities. The comparison of $\mathcal{G}$ and $\mathcal{D}$ can be made in terms of the following two kinds of measures [14]:

$$\sigma_{ij} = \frac{\text{Area}(G_i \cap D_j)}{\text{Area}(G_i)} \quad \text{and} \quad \tau_{ij} = \frac{\text{Area}(G_i \cap D_j)}{\text{Area}(D_j)} \tag{18}$$

where $1 \le i \le M$, $1 \le j \le N$, and 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 analyse 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 ($\sigma_{ij} < 1$ for all j, and $\Sigma_{j=1}^{N} \sigma_{ij} \approx 1$);
- many-to-one match ($\tau_{ij} < 1$ for all i, and $\Sigma_{i=1}^{M} \tau_{ij} \approx 1$);
- many-to-many match (others).

An example of matching between a set of ground truth entities and the detected entities is illustrated in Fig. 15. By computing their area overlap, we construct two matrices, $\Sigma = (\sigma_{ij})$ and $T = (\tau_{ij})$, shown in Table 2. 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).

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

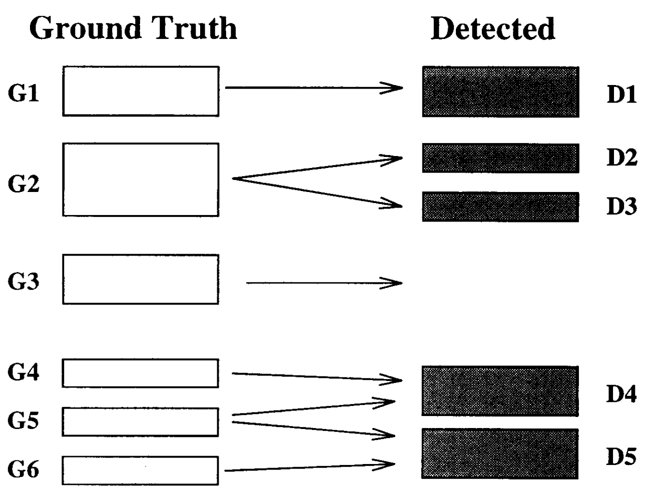


Fig. 15. Correspondence between ground truth and detected structures.

Table 2. The area overlap matrices computed from the example in Fig. 15

	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

$T = (\tau_{ij})$

not detected by the segmentation (misdetection), and vice-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.

7. EXPERIMENTAL RESULTS

The text block extraction algorithm described in this paper is evaluated on a total of 1600 images from the UW-III Document Image Database. 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 and a different part is used for testing at each time. Finally, the performance measures from three parts are combined as the overall performance of the algorithm on the data set.

Within the 1600 document pages in the UW-III database, there are total of 21,788 ground truth text blocks. Each text block is associated with its bounding box information and a text alignment label.

First, we apply the text alignment detection algorithm to the ground-truth blocks, and estimate its performance. The classification contingency table is shown in Table 3. Note that the classification algorithm rejects a text block when there are less than three text lines within the block. The total number of tested text blocks is 11,753. Of them, 11,348 are correctedly classified, and the number of misclassifications is 405. The detection rate of our text alignment detection algorithm is estimated as 96.55%.

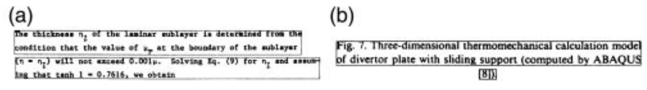
Then, we apply the text block extraction algorithm on the ground-truth text lines in the UW-III database. The numbers and percentages of miss, false, correct, splitting, merging and spurious detections are shown in Table 4(a). Of the 21,788 ground truth text blocks, 91% of them are correctly detected, and 2.57% and 5.74% of blocks are split or merged, respectively.

Finally, the text block extraction algorithm is applied to the text lines generated by our text line extraction algorithm [19]. Table 4(b) shows that 88.91% of text blocks are correctly identified and segmented.

We did a careful examination of all the document pages on which our text block algorithm had made errors. We discovered that most of the pages that our algorithm had failed are in these categories: (1) pages containing nested-list items; (2) pages containing poorly formatted list items; (3) pages containing 'pseudo-codes'; or (4) pages containing two adjacent text blocks that are not possible to separate without knowing the text content. Figure 16 illustrates some of the examples where the text block extraction algorithm failed to identify text blocks correctly.

8. SUMMARY

In this paper, we formulate the text block segmentation as a partitioning problem. The goal of the problem is to find an optimal solution to partition the set of input text lines into a set of text blocks (with a text alignment label) that preserves the formatting property of the original input document. A Bayesian framework is used to assign and


(a)
The thickness η_l of the laminar sublayer is determined from the condition that the value of ν_T at the boundary of the sublayer
($\eta = \eta_l$) will not exceed 0.001μ. Solving Eq. (9) for η_l and assuming that tanh 1 = 0.7616, we obtain

(b)
Fig. 7. Three-dimensional thermomechanical calculation model of divertor plate with sliding support (computed by ABAQUS
[8])

Fig. 16. Examples where the text block extraction algorithm failed to identify text blocks correctly. (a) Splitting error; (b) splitting error; (c) merging error; (d) merging error.

Table 3. The text alignment classification results on the ground truth text blocks from the UW-III database. Each text block has at least three text lines. The mis-classification rate is 3.45%

	Justified	Left	Centre	Right	Justified-hanging	Left-hanging
Justified	8254	129	7	2	5	0
Left	19	747	0	0	1	5
Centre	20	2	122	1	1	0
Right	1	0	1	18	0	0
Justified-hanging	22	1	0	0	1983	128
Left-hanging	4	5	0	0	51	224

Table 4. Performance of the text block grouping algorithm applied on (a) the ground truth text lines, and (b) the detected text lines

	Total	Correct	Splitting	Merging	Mis-False	Spurious
Ground Truth	21788	19828 (91.00%)	560 (2.57%)	1250 (5.74%)	1 (0.01%)	149 (0.68%)
Detected	21709	19828 (91.34%)	1219 (5.62%)	501 (2.31%)	0 (0.00%)	161 (0.74%)

(a)

	Total	Correct	Splitting	Merging	Mis-False	Spurious
Ground Truth	21788	19373 (88.91%)	826 (3.80%)	1298 (5.96%)	9 (0.04%)	282 (1.30%)
Detected	22180	19373 (87.34%)	1999 (9.02%)	521 (2.35%)	1 (0.00%)	286 (1.29%)

(b)

update the probabilities during the text block grouping and labeling process. An iterative, relaxation-like method is used to find a partitioning solution that maximises the joint probability.

The probabilities used within the algorithm are estimated from an extensive training set of various kinds of measurements of distances between the terminal and non-terminal entities with which the algorithm works. The off-line probabilities estimated in the training then drive all decisions in the on-line segmentation module.

The text-alignment detection algorithm and the text block extraction algorithm were tested on the 1600 pages of technical documents within the UW-III database. Of the total of 21,788 text blocks within these pages, the text-alignment detection algorithm yields a 96.55% accuracy rate, and the text block partition algorithm exhibits a 91% accuracy rate.

References

1. Haralick RM. Document image understanding: geometric and logical layout. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1994: 385–390
2. Nagy G, Seth S. Hierarchical representation of optically scanned documents. Proceedings of the Seventh International Conference on Pattern Recognition. Montreal, Canada, 1984: 347–349
3. Ittner DJ, Baird HS. Language-free layout analysis. Proceedings of the Second International Conference on Document Analysis and Recognition. Tsukuba, Japan, 1993: 336–340
4. Dengel A, Dubiel F. Clustering and classification of document structure: a machine learning approach. Proceedings of the Third International Conference on Document Analysis and Recognition. Montreal, Canada, 1995: 587–591
5. Chen S. Document layout analysis using recursive morphological transforms. PhD Thesis, University of Washington, 1995
6. Ha J, Haralick RM, Phillips IT. Document page decomposition using bounding boxes of connected components of black pixels. In: Vincent LM, Baird HS (Eds). Document Recognition II. 1995: 140–151
7. Liang J, Ha J, Haralick RM, Phillips IT. Document layout structure extraction using bounding boxes of different entities. Proceedings of the Third IEEE Workshop on Applications of Computer Vision. 1996: 278–283
8. Palumbo PW, Srihari SN, Soh J, Sridhar R, Demjanenko V. Postal address block location in real time. IEEE Computer 1992: 34–42
9. Etemad K, Doermann D, Chellappa R. Multiscale segmentation

of unstructured document pages using soft decision integration. IEEE Transactions on Pattern Analysis and Machine Intelligence 1997; 19(1):92–96

10. Kise K, Sato A, Iwata M. Segmentation of page images using the area Voronoi diagram. Computer Vision and Image Understanding 1998; 70(3):370–382
11. Wang SY, Yagasaki T. Block selection: a method for segmenting page image of various editing styles. Proceedings of the Third International Conference on Document Analysis and Recognition. Montreal, Canada, 1995: 128–135
12. Jain AK, Yu B. Document representation and its application to page decomposition. IEEE Transactions on Pattern Analysis and Machine Intelligence 1998; 20(3):294–308
13. Watanabe T, Luo Q, Sugie N. Structure recognition methods for various types of documents. Machine Vision and Applications 1993; 6:163–176
14. Liang J, Phillips IT, Haralick RM. Performance evaluation of document layout analysis on the UW data set. Proceedings of the SPIE, Document Recognition IV. San Jose, CA, 1997: 149–160
15. Phillips IT, Chen S, Haralick RM. English document database standard. Proceedings of the Second International Conference on Document Analysis and Recognition. Japan, 1993: 478–483
16. Phillips IT. User's Reference Manual for the UW English/Technical Document Image Database III. UW-III English/Technical Document Image Database Manual, 1996
17. Rabiner LR. A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE 1989; 77(2):257–285
18. Aas K, Eikvil L, Huseby RB. Applications of hidden Markov chains in image analysis. Pattern Recognition 1999; 32: 703–713
19. Liang J. Document structure analysis and performance evaluation. PhD Thesis, University of Washington, 1999

Jisheng Liang received his BS degree from Tianjin University, Tianjin, China, in 1992, and PhD degree from the University of Washington, Seattle, in 1999, both in electrical engineering. His research interests include image processing, pattern recognition, document imaging and multimedia information retrieval. Dr Liang has published a dozen papers in the area of document image analysis. He helped design and implement the University of Washington Document Image Databases. Dr Liang is currently working as a research scientist at the Data Analysis Products Division of MathSoft, Inc. in Seattle, WA. He has been involved in several projects concerning character recognition, medical imaging, and video processing.

Ihsin T. Phillips received her BS, MS and PhD in 1979, 1981, and 1984, all in computer science, from the University of Maryland, College Park, MD. In 1984, she joined the Department of Computer Science at the University of Maryland as an Assistant Professor. In 1985, she joined the Department of Computer Science and Software Engineering at Seattle University, Seattle, WA, where she was promoted to Associate Professor in 1991 and Professor in 1997. She is currently the holder of the Thomas J. Bannan Endowed Chair in Engineering from the School of Science and Engineering at Seattle University. Dr Phillips 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. Dr Phillips' 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. Since their creation, these three databases have been used by researchers and developers in the field from all over the world for testing and benchmarking their systems. Dr Phillips served as program committee member for several IEEE and IAPR conferences and workshop. 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 chairwoman of the IAPR technical committee on performance evaluation. She is a member of IEEE and the IEEE Computer Society.

Robert M Haralick occupies the Boeing Clairmont Egtvedt Professorship in the Department of Electrical Engineering at the University of Washington. He was responsible for developing the grey scale co-occurrence texture analysis technique and the facet model technique for image processing. He has worked on robust methods for photogrammetry and developed fast algorithms for solving the consistent labelling problem. He has developed shape analysis and extraction techniques using mathematical morphology, he developed the theory for the morphological sampling theorem, and fast recursive morphology algorithms. In the area of document image understanding, Professor Haralick, along with Professor Ihsin Phillips, developed a comprehensive ground-truthed set of some 1600 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, word and text line bounding box delineation. Professor Haralick is a Fellow of the IEEE for his contributions in computer vision and image processing, and a Fellow of the IAPR for his contributions in pattern recognition, image processing, and for service to IAPR. He has published over 500 papers and has just completed his term as the president of the IAPR.

Correspondence and offprint requests to: Dr IT Phillips, Department of Computer Science, Software Engineering, Seattle University, Seattle, WA 98122, USA. Email: yun@george.ee.washington.edu