Computer Vision and Image Understanding **84,** 144–159 (2001)
doi:10.1006/cviu.2001.0933, available online at http://www.idealibrary.com on IDEAL®

# Performance Evaluation of Document Structure Extraction Algorithms

Jisheng Liang

*Insightful Corporation, 1700 Westlake Avenue N. Suite 500, Seattle, Washington 98109*
E-mail: jliang@insightful.com

Ihsin T. Phillips

*Department of Computer Science, Queens College, City University of New York, 65-30 Kissena Boulevard, Flushing, New York 11367*
E-mail: yun@image.cs.qc.edu

and

Robert M. Haralick

*Department of Computer Science, Graduate Center, City University of New York, 365 Fifth Avenue, New York, New York 10016*
E-mail: haralick@gc.cuny.edu

Received September 15, 1999; accepted August 8, 2001

This paper presents a performance metric for the document structure extraction algorithms by finding the correspondences between detected entities and ground truth. We describe a method for determining an algorithm's optimal tuning parameters. We evaluate a group of document layout analysis algorithms on 1600 images from the UW-III Document Image Database, and the quantitative performance measures in terms of the rates of correct, miss, false, merging, splitting, and spurious detections are reported. © 2001 Elsevier Science (USA)

## 1. INTRODUCTION

The 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 provided illustrative results and hardly any had their techniques tested on significant-sized data sets and gave quantitative performance measures [1]. The main reasons are the lack of precise and concise mathematical document model, 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. Since document analysis techniques are moving to the consumer market, they must perform nearly perfectly. This means that they must be proved out on significant-sized data sets and that there must be suitable performance metrics for each kind of information a document analysis technique infers. From the user's point of view, the appropriate measure of a document recognition system is the total cost of document conversion, typically dominated by the cost of correcting residual errors in the output. However, such a measure is not necessarily appropriate for a researcher who seeks to measure progress in developing an algorithm [2]. A developer might use a number of metrics at an early stage of development, and an overall cost to evaluate the final system. Different performance metrics may be important at different stages of system development.

Currently, no standard testing procedures exist for measuring and comparing algorithms within a document structure analysis system. Randriamasy and Vincent [3] 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 that 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 a representation scheme of regions. Quantitative evaluation of segmentation is derived from the cost of incorrect splittings and merging. 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.* [2] proposed a text-based method to evaluate the zone segmentation performance. They compute an edit distance that is based on the number of edit operations (text insertions, deletions, and block moves) required to transform an OCR (Optical Character Recognition) 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 [4] proposed a scoring method that computes the coverage and efficiency of the 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.

This paper introduces quantitative performance metrics for each kind of information a document structure analysis technique infers. 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. In the University of Washington Document Image Database series [8, 10], there are 1600 English document images that come with manually edited ground truth of entity bounding boxes. These bounding boxes enclose text and nontext zones, text lines, and words. We do the performance evaluation by determining the correspondence between the ground truth document structure and the automatically computed document structure, making the comparison between two structures, and reporting the performance measures.

In this paper, we first give a formal definition of the hierarchical document structure in Section 2. In Section 3, a set of quantitative metrics are presented to evaluate the performance of the document analysis algorithms. We also describe a method for automatically selecting an algorithm's parameters. Finally, the performance evaluation results of a group of document layout analysis algorithms on the UW-III database are presented in Section 4.

## 2. HIERARCHICAL DOCUMENT STRUCTURE

Document structure is a hierarchical structure, which includes a set of entities at different levels of the hierarchy. Each entity is represented by a polygonal area on the image and is associated with a list of attributes, and there is a certain reading order among entities on the same hierarchical level. In this section, we present a formal definition of the hierarchical document structure.

A polygonal area on a document page is given by a pair $(\theta, I)$, where $\theta \in \Theta$ specifies the label that designates the physical type of content, i.e., text block, text line, word, table, equation, drawing, half-tone, etc., and $I$ is the area enclosed by boundary of the polygon. A polygon is homogeneous if all of its area is of one physical type and there is a standard reading order for the content within the area. Two polygons are physically adjacent if each has a significant length of a side that near parallels 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. A polygonal structure is associated with the following properties:

1. *Content Type.* $\Theta$ is the set of physical types (text block, text line, word, table, equation, drawing, etc.). $C\colon \mathcal{A} \to \Theta$ associates polygonal areas with their physical types of content. $\Gamma$ is the set of functional types (paragraph, section, word, title, heading, caption, abstract, author, footnote, page number, etc.). $M\colon \mathcal{A} \to \Gamma$ associates polygonal areas with their functional types of content.
2. *Content.* $\Sigma$ is the alphabet consisting of symbols. Let $\Sigma^*$ be the set of all sequences of symbols from $\Sigma$. Let $\mathcal{A}_t \subseteq \mathcal{A}$ be the set of text polygonal areas. $O\colon \mathcal{A}_t \to \Sigma^*$ associates text polygonal areas with their contents.
3. *Format Attribute.* We denote by $\mathcal{F}$ the set of format attributes (font type, font size, font style, justification, indentation, etc.). $S\colon \Gamma \to \mathcal{F}$ specifies the format attributes for each functional type of content.
4. *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.
5. *Spatial Relation.* $\mathcal{D}$ is the set of dividers (white space, ruling, etc.). $T\colon \Gamma \times \Gamma \to \mathcal{D}$ specifies the divider used between different types of content (interline spacing, intercolumn spacing, spacing between figure and caption, etc.).
6. *Reading Order.* Let $A = \{A_1, \ldots, A_K\}$ be a set of polygonal areas. The reading order $R$ of $A$ is a tuple $(r_1, \ldots, r_K)$, which is a permutation of $(1, \ldots, K)$.

$V\colon \wp(\mathcal{A}) \to \Lambda$ specifies measurement made on subset of $\mathcal{A}$, where $\Lambda$ is the measurement space.

The document structure is a hierarchical structure. Let $\Phi$ be the set of all polygonal structures on a given document page. We define the relation $\subseteq_\Phi$ in $\Phi$ as $x \subseteq_\Phi y$ if and only if $x \subseteq y$ and $x, y \in \Phi$. We denote by $\Theta = \{\theta_1, \theta_2, \ldots, \theta_k\}$ the set of content types on the document page. Let $\{\Phi_1, \ldots, \Phi_k\}$ be a partition of $\Phi$, where each $\Phi_i$ is a set of mutually disjoint polygonal structures

$$\Phi_i = \{\phi \mid \phi \in \Phi, C(\phi) = \theta_i\}, \tag{1}$$

and if there exists $\Phi_j$ and $j > i$, then $\forall \phi_p \in \Phi_i$, either $\phi_p \subseteq_\Phi \Phi_j$, or $\phi_p$ and $\Phi_j$ are disjoint.

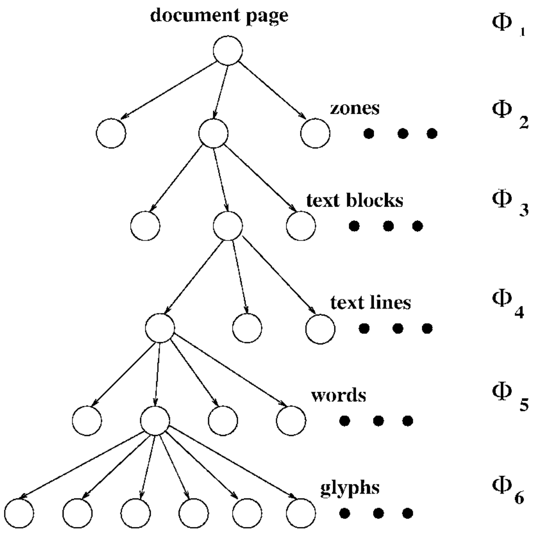


**FIG. 1.** A document hierarchy, where $\Phi_i$ represents a subset of physical document entities on a certain level.

An example of the document hierarchy, where each level consists of a set of physical entities, is shown in Fig. 1. Figure 2 illustrates a document hierarchy where each entity is associated with a logical label. Given a document image, the goal of document structure analysis is to extract a most likely hierarchical structure based on observations from the image.

Document structure analysis usually contains the following subproblems:

- Layout analysis extracts a set $\mathcal{A}$ of homogeneous polygonal areas from a document page.
- Logical analysis $(M, \mathcal{R})$ involves assigning functional labels to each entity of the page, and ordering the entities according to their read order.

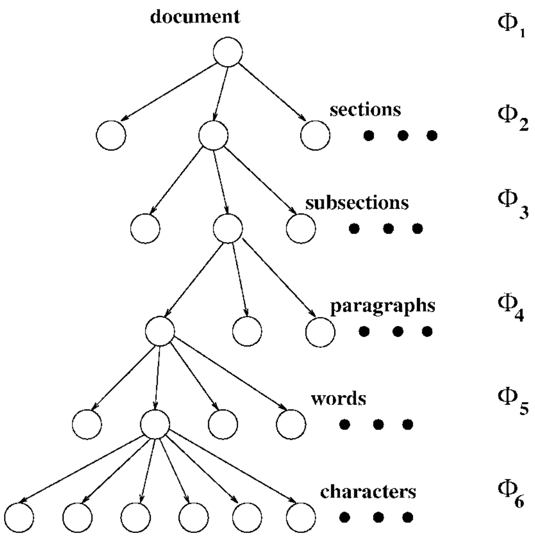


**FIG. 2.** A document hierarchy, where $\Phi_i$ represents a subset of logical document entities on a certain level.

- Content recognition $O$ associates entities with their contents.
- Page style $(S, T, \mathcal{P})$ describes the typographical properties for entities with different content types.

## 3. PERFORMANCE EVALUATION OF DOCUMENT STRUCTURE ANALYSIS

Let $g$ designate the ground truth structure, $d$ the detected structure, and $u(g, d)$ the utility of $(g, d)$. The expected performance of an algorithm is then

$$f = \sum_g \sum_d u(g, d) P(g, d), \tag{2}$$

where $P(g, d)$ is the joint probability of the algorithm, output is $d$, while the ground truth is $g$ [5].

For each structure that we use to describe a document, there is an associated metric that measures the difference between a structure that is automatically produced and the ground truth structure. We present the utility function and performance measure for the document analysis algorithms in this section. Given the ground-truthed and detected structures, the steps of performance evaluation are as follows:

1. Match entities from two structures based on their area overlap.
2. Evaluate segmentation (correct, misdetection, false alarm, splitting, and merging) of entities.
3. For matched entities, evaluate detection of their properties.

Because the assessment tests only observe a finite sample, there is of necessity a difference between the observed performance on the test sample and the long-term performance on the total population. The issue of making the comparison between the specification and the observed performance is addressed by Haralick [6].

### 3.1. Matching of Detected Structure with the Ground Truth

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 ground-truthed entities and $\mathcal{D} = \{D_1, D_2, \ldots, D_N\}$ for detected entities. The comparison of $\mathcal{G}$ and $\mathcal{D}$ can be made in terms of the following two kinds of measures,

$$\sigma_{ij} = \frac{\text{Area}(G_i \cap D_j)}{\text{Area}(G_i)} \quad \text{and} \quad \tau_{ij} = \frac{\text{Area}(G_i \cap D_j)}{\text{Area}(D_j)}, \tag{3}$$

where $1 \leq i \leq M$, $1 \leq j \leq N$, and Area$(A)$ represents the area of $A$. The measures in the above equation constitute two matrices $\Sigma = (\sigma_{ij})$ and $T = (\tau_{ij})$. Note that $\sigma_{ij}$ indicates how much portion of $G_i$ is occupied by $D_j$, and $\tau_{ij}$ indicates how much portion of $D_j$ is occupied by $G_i$. Our strategy of performance evaluation is to analyze these matrices to

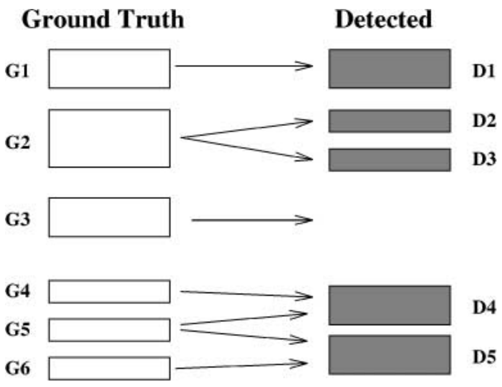


**FIG. 3.** Correspondence between ground truth and detected structures.

determine the correspondence between two sets of entity areas:

- one-to-one match ($\sigma_{ij} \approx 1$ and $\tau_{ij} \approx 1$);
- one-to-zero match ($\sigma_{ij} \approx 0$ for all $1 \leq j \leq N$);
- zero-to-one match ($\tau_{ij} \approx 0$ for all $1 \leq i \leq M$);
- one-to-many match ($\sigma_{ij} < 1$ for all $j$, and $\sum_{j=1}^{N} \sigma_{ij} \approx 1$);
- many-to-one match ($\tau_{ij} < 1$ for all $i$, and $\sum_{i=1}^{M} \tau_{ij} \approx 1$);
- 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 1. 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 $D_5$).

## 3.2. Performance Measure of Layout Analysis

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

**TABLE 1**
**The Area Overlap Matrices Computed from the Example in Fig. 3**

| | $D_1$ | $D_2$ | $D_3$ | $D_4$ | $D_5$ | | $D_1$ | $D_2$ | $D_3$ | $D_4$ | $D_5$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $G_1$ | 0.95 | | | | | $G_1$ | 0.9 | | | | |
| $G_2$ | | 0.45 | 0.51 | | | $G_2$ | | 0.85 | 0.91 | | |
| $G_3$ | | | | | | $G_3$ | | | | | |
| $G_4$ | | | | 0.7 | | $G_4$ | | | | 0.4 | |
| $G_5$ | | | | 0.37 | 0.29 | $G_5$ | | | | 0.25 | 0.3 |
| $G_6$ | | | | 0.8 | | $G_6$ | | | | | 0.45 |
| | | | $\Sigma = (\sigma_{ij})$ | | | | | | $T = (\tau_{ij})$ | | |

**TABLE 2**
**Weights Used for Computing the Performance Measure of Layout Analysis**

| Correct | Merge | Split | Miss | False | Spurious |
|---|---|---|---|---|---|
| 0.0 | 0.5 | 0.5 | 1.0 | 1.0 | 1.0 |

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 ($G^m \subset G$ is identified as $D^m \subset D$ in the sample) as

$$P^m(G, D) = \frac{|G^m| + |D^m|}{|G| + |D|}, \tag{4}$$

where $|A|$ denotes the size of the set $A$. Then the performance measure of a layout analysis process is defined as

$$f_{\text{layout}} = \sum_{m \in M} W^m P^m(G, D), \tag{5}$$

where $M = \{\text{correct, miss, false, merging, splitting, spurious}\}$ is the set of possible matching, and $W^m$ is the weight assigned to each type of matching.

For the example shown in Fig. 3, we determine that $G_1$ is correctly detected; $G_2$ is split into $D_2$ and $D_3$; $G_3$ is missed; and $G_4$, $G_5$, and $G_6$ are merged and split into $D_4$ and $D_5$. In the experiment, we choose the weights as in Table 2. Therefore, the performance measure for the example is

$$f_{\text{layout}} = \frac{0.0 \times (1+1) + 0.5 \times (1+2) + 1.0 \times 1 + 1.0 \times (3+2)}{6+5} = 0.68.$$

Each kind of matching can be further divided into different categories. For example, merging text across columns apparently should be given more penalty than merging two adjacent text blocks within the same column. A many-to-one matching between ground truth zones and detected zones can be divided into the following possible detection errors:

- merge text and nontext,
- merge text across columns,
- merge text zones with different reading directions, and
- merge adjacent homogeneous text.

### 3.3. Performance Measure of Classification

A classification module classifies each extracted structure into one of the predefined categories. Labeling of the physical and functional types, detection of format attributes, and isolated character recognition are typical classification problems. The output of the classification process is compared with the labels from the ground truth in order to evaluate the performance of the algorithm. Let $P(t, a)$ be the probability of observing a unit, in

**TABLE 3**
**An example of a Contingency Table Showing the Classification Results of Text and Nontext Zones**

| | Text | Nontext |
|---|---|---|
| Text | $P(t, t)$ | $P(t, n)$ |
| Nontext | $P(n, t)$ | $P(n, n)$ |

*Note.* Rows represent the true categories and columns represent the assigned categories.

the sample, whose true category is $t$, and whose assigned category is $a$. The expected performance is

$$f_{\text{label}} = \sum_{t\in\Psi}\sum_{a\in\Psi} W(t, a)P(t, a), \tag{6}$$

where $\Psi$ is the set of categories, and $W(t, a)$ is the weight associated with different assignment. A contingency table is computed to indicate the number of entities of a particular class label that are identified as members of another class. The misclassification rate is defined as

$$f_{\text{classification}} = \sum_{t\in\Psi}\sum_{a\in\Psi,\ a\neq t} P(t, a). \tag{7}$$

An example of a contingency table showing the classification results of text and nontext zones is shown in Table 3. We denote by $P(n, t)$ the fractions of entities in the sample that are truly nontext but are assigned text, and so on.

The misclassification rate is the sum of $P(t, n)$ and $P(n, t)$,

$$f_{\text{classification}} = P(n, t) + P(t, n).$$

The misdetection rate of text entity is computed as

$$P(n \mid t) = \frac{P(t, n)}{P(t, t) + P(t, n)},$$

and the false-alarm rate of text entity is computed as

$$P(t \mid n) = \frac{P(n, t)}{P(n, t) + P(n, n)}.$$

## 3.4. Performance Measure of Reading Order Detection

Let $\mathcal{A} = \{A_1, \ldots, A_K\}$ be the set of entities that have been correctly identified. The detected reading order $\hat{R}$ is a tuple $(\hat{r}_1, \ldots, \hat{r}_K)$, which is a permutation of true reading order $R = (r_1, \ldots, r_K)$. The problem of evaluating a reading order determination algorithm can be reduced to computing the minimum number of moves required to obtain $\hat{R}$ from $R$. The problem can be modeled as a sorting problem where a string of $K$ integers ordered in a random manner must be sorted in ascending (or descending) order [7].

### 3.5. Systematic Parameter Selection of Algorithms

Performance evaluation serves two purposes. One is comparing the ground truth and algorithm output and producing a quantitative overall measure of the algorithm. Another is to help the developers analyze the results and their algorithms. In this section, we describe a method for determining the optimal algorithm tuning parameters given the training data. A model-checking method for algorithms based on parametric document models is presented in [14, 19].

In document analysis algorithms, tuning parameters (thresholds) are often used. These parameters have been traditionally chosen by trial-and-error. In this section, we describe a systematic parameter estimation method by optimizing the algorithm performance over the training data. The parameter estimation problem can be stated as follows: Given an algorithm with parameter $X = (X_1, X_1, \ldots, X_N)^{\mathrm{T}}$, search for the parameter vector $X$ that optimizes expected performance $f(X)$. This process is equivalent to fitting the algorithm's underlying model to the data. Here we describe an engineering method based on the given criterion function, instead of the traditional trial-and-error approach.

When a representative sample set of a domain is available, and a quantitative performance metric is defined, we can tune the parameter values of an algorithm and select a set that produces the optimal performance on the input population. Suppose the performance measure $f$ is the cost, then the best tuning parameter vector is $X^*$ after

$$X^* = \arg\min_X f(X) \tag{8}$$

and $X = (X_1, X_2, \ldots, X_N)^{\mathrm{T}}$ is the vector of parameters.

We use an iterative first-order search method to find the optimal parameters,

$$X^q = X^{q-1} + k_q^* S^q. \tag{9}$$

Here $q$ is the iteration number, $S^q$ is the search direction, and $k_q^*$ is a scalar multiplier determining the amount of change in $X$ for the iteration. The search direction is taken as the negative of the gradient of the function,

$$S^q = -\nabla f(X). \tag{10}$$

To compute the gradient of the function, we consider a first-order Taylor series expansion of $f(X)$ about $X^0$

$$f(X) \simeq f(X^0) + \frac{\partial f(X^0)}{\partial X}\Delta X,$$

where

$$\Delta X = X - X^0.$$

By changing only one parameter $X_i$ by $\Delta X_i$,

$$\Delta X = (0, 0, \ldots, \Delta X_i, \ldots, 0, 0)^{\mathrm{T}},$$

we have

$$f(X) \simeq f(X^0) + \frac{\partial f(X^0)}{\partial X_i} \Delta X_i.$$

This leads to

$$\frac{\partial f(X^0)}{\partial X_i} = \frac{f(X) - f(X^0)}{\Delta X_i}.$$

By tuning $N$ parameters one by one and obtaining the corresponding performance, the gradient function $\nabla f(x)$ can be calculated as

$$\frac{\partial f(X^0)}{\partial X} = \left( \frac{\partial f(X^0)}{\partial X_1}, \ldots, \frac{\partial f(X^0)}{\partial X_N} \right). \tag{11}$$

After finding the search direction, we use the golden section algorithm [11, 12] to find the best scalar $k$.

This method is similar to the back-propagation algorithm of neural networks, which is a first-order approximation of the steepest-descent technique in the sense that it depends on the gradient of the instantaneous error surface in weight space [13]. Like any gradient descent search, our parameter tuning method has problems with efficiency and convergence. It runs the risk of being trapped in a local minimum, where every small change in parameters increases the cost function. Usually, however, a reasonably good solution can be found after a small number of iterations. Simulated annealing [11] provides a way of escaping local minimum by taking some random steps instead of going greedily for the quick, nearby solution. In other words, the method sometimes goes uphill as well as downhill in the search space, determined by a control parameter (temperature). However, the assignment of an annealing schedule may require physical insight and/or trial-and-error experiments.

## 4. EXPERIMENTAL RESULTS

In this section, we briefly present a rule-based algorithm that extracts document layout structure using the bounding boxes of different entities [15]. Then we report the performance of each module on the images from the UW-III Document Image Database.

### 4.1. UW Document Image Databases

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. At the University of Washington, we have produced the UW Document Image Database [8, 9] series of ground truth databases on CDROM for the document image analysis and recognition communities. The first two databases in the series, UW-I and UW-II, provide a variety of ground truth information about their constituent documents, including zone and page bounding boxes, attributes, and ASCII text. In order to facilitate the training and evaluation of document layout analysis algorithms, UW-III [10] includes text line bounding boxes

**TABLE 4**
**Ground Truth Information Provided for Each Page in the UW-III Document Image Database**

| Structure | Ground truth |
|---|---|
| Layout | Hierarchy of bounding boxes enclosing page, header, footer, text and nontext zones, text lines, and words for each page. |
| Logical | Label of content type for each zone. Reading order of text and nontext zones, text lines, and words. |
| Content | Text content for each text zone. |
| Style | Page and zone format attributes. |

and word bounding box ground truth, in addition to the ground truth information provided in the previous database releases. Ground truth for nontext structures are also provided in the database, such as chemical formulas, mathematical formulas, and engineering line drawings.

The UW-III contains a total of 1600 English document images randomly selected, copied, and scanned from scientific and technical journals. Each page contains a hierarchy of manually verified page, zone, text line, and word entities with bounding box and in the correct reading order. The text ground truth and zone attributes are attached to each zone entity. The zone attribute includes information about the type of the zone (i.e., text, figure, table, etc.), the dominant font size, the dominant font style, and the justification. A list of ground truth information for each page is provided in Table 4.

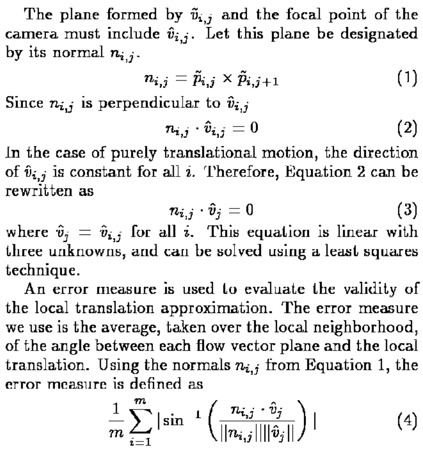
The UW document image databases can be utilized by the document understanding community as a common platform to develop, test, and evaluate their systems. Based on the ground truth data, we can evaluate the performance of document analysis algorithms and build statistical models to characterize various types of document structures. For each structure that we use to describe a document, there is an associated metric that measures the difference between a structure that is automatically produced and the ground truth structure.

## 4.2. Algorithm Description

The input of the algorithm is a set of bounding boxes that enclose the connected components on a binary document image. Figure 4a shows a segment of a document image, and Fig. 4b shows the bounding boxes produced in this step.

The document page segmentation roughly divides the image into a list of zones by analyzing the spatial configuration of bounding boxes. The analysis is done by projecting bounding boxes onto vertical and horizontal lines [16]. A projection profile is a frequency distribution of the bounding boxes on the projection line. Figures 4c and 4d show the horizontal and vertical projection profiles of the bounding boxes in Fig. 4b.

A document image can be segmented using a recursive $X$–$Y$ cut [17] procedure based on the bounding boxes. At each step, the horizontal and vertical projection profiles are calculated. Then a zone division is performed at the most prominent valley in either projection profile. The process is repeated recursively until no sufficiently wide valley is left. The free



The plane formed by $\tilde{v}_{i,j}$ and the focal point of the camera must include $\hat{v}_{i,j}$. Let this plane be designated by its normal $n_{i,j}$.

$$n_{i,j} = \tilde{p}_{i,j} \times \tilde{p}_{i,j+1} \quad (1)$$

Since $n_{i,j}$ is perpendicular to $\hat{v}_{i,j}$

$$n_{i,j} \cdot \hat{v}_{i,j} = 0 \quad (2)$$

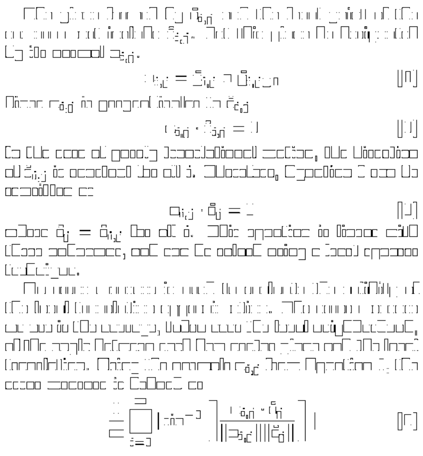
In the case of purely translational motion, the direction of $\hat{v}_{i,j}$ is constant for all $i$. Therefore, Equation 2 can be rewritten as

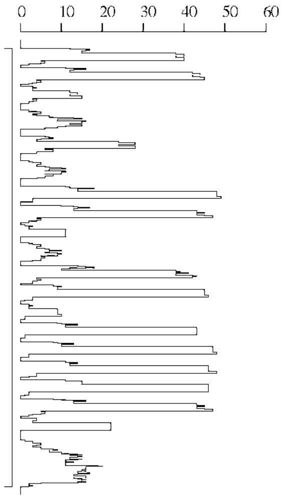
$$n_{i,j} \cdot \hat{v}_j = 0 \quad (3)$$

where $\hat{v}_j = \hat{v}_{i,j}$ for all $i$. This equation is linear with three unknowns, and can be solved using a least squares technique.

An error measure is used to evaluate the validity of the local translation approximation. The error measure we use is the average, taken over the local neighborhood, of the angle between each flow vector plane and the local translation. Using the normals $n_{i,j}$ from Equation 1, the error measure is defined as

$$\frac{1}{m}\sum_{i=1}^{m} |\sin^{-1}\left(\frac{n_{i,j} \cdot \hat{v}_j}{||n_{i,j}|| \, ||\hat{v}_j||}\right)| \quad (4)$$





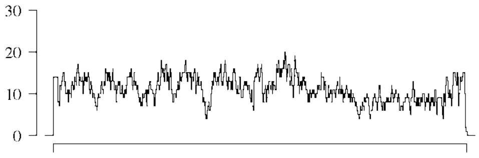


**FIG. 4.** (a) An English document, (b) bounding boxes of connected components of black pixels, (c) a horizontal projection profile, and (d) a vertical projection profile.

parameters used in the page segmentation algorithm are as follows:

- the minimum connected component width and height: $w_1$ and $h_1$.
- the maximum connected component width and height: $w_2$ and $h_2$.
- the maximum height/width aspect ratio of connected component: $r_1$.
- the minimum width of the valley in the vertical projection profile, $w_3$, and in the horizontal profile, $w_4$. If $w_3$ or $w_4$ is smaller than a threshold, no cut will be applied.
- the maximum height/width aspect ratio of the current region: $r_2$. If $r_2$ is larger than a threshold, no further cut will be applied.

Inside each zone generated from the page segmentation process, the spatial configuration of bounding boxes is analyzed to extract text lines. From Figs. 4c and 4d, it is clear that the text lines can be extracted by finding the distinct high peaks and deep valleys at somewhat regular intervals in the horizontal projection profile. The free parameters used in this text line extraction algorithm are listed as follows:

- the minimum height of text line: $h_3$.
- the maximum height of text line: $h_4$.
- the threshold for detecting rule: $r_3$. If a detected text line contains only one component, and the text line's aspect ratio is larger than $r_3$, the text line is considered as a rule.
- the minimum size of connected components accepted for projection: $w_5$ and $h_5$.
- the maximum size of connected components accepted for projection: $w_6$ and $h_6$.
- the maximum aspect ratio of connected components accepted for projection: $r_4$.

Within each textual zone, text lines are merged into text blocks. The beginning of a text block, such as a paragraph, math zone, and section heading, is usually marked either by changing the justification of the current text line, by putting extra space between two text lines, or by changing font size or font style. So when a significant change in intertext line spacing or justification occurs, it is very likely that a new text block begins. The free parameters used in this algorithm are:

- justification threshold: $t_1$.
- interline spacing threshold: $t_2$.

## 4.3. Parameter Tuning

The parameters of each algorithm are decided using the optimization process described in Section 3.5. For the segmentation process, the weights we use for computing the performance are given in Table 2. Therefore, our performance criterion is the cost for converting the detected layout structure to the ground truth.

For each algorithm, we start with their default parameters obtained by observing a small number of images, then we tune the parameters by minimizing the algorithm's cost on the images in the UW-III database, until the improvement of performance is less than a threshold. The algorithms' cost after the parameter tuning, compared to their performance using the default parameters, is shown in Table 5. The average improvement is 17.6% with respect to the original performance.

The numbers and percentages of miss, false, correct, splitting, merging, and spurious detections of each algorithm after the optimization process are presented in the following sections. The parameter values before and after the optimization process are also given.

**TABLE 5**
**The Performance of Algorithms before and after the Optimization Process**

| | Page | Line | Block |
|---|---|---|---|
| Original | 0.143 | 0.022 | 0.141 |
| Optimized | 0.104 | 0.017 | 0.137 |
| Improved by | 27.3% | 22.7% | 2.8% |

**TABLE 6**
**The Default and Optimized Parameters for the Page Segmentation**

| | $w_1$ | $h_1$ | $w_2$ | $h_2$ | $r_1$ | $r_2$ | $w_3$ | $w_4$ |
|---|---|---|---|---|---|---|---|---|
| Default | 3 | 3 | 1800 | 2200 | 50 | 5 | 30 | 30 |
| Optimized | 3.4 | 3.3 | 2615 | 2187 | 52.6 | 3.1 | 39.1 | 34.4 |

## 4.4. Performance of Page Segmentation

Given a set of connected components computed from a document image, page segmentation partitions the components into a set of zones, such that each zone is homogeneous (text zones with homogeneous read order, and nontext zones). The parameters before and after the optimization are given in Table 6. Table 7 illustrates the numbers, and percentages of miss, false, correct, splitting, merging and spurious detections with respect to the ground truth zones as well as the algorithm output. Since the page segmentation finds the coarse homogeneous zones, we do not consider the merging of adjacent text within the same column as error.

This algorithm is restricted to bi-directional $X$–$Y$ cuttable layouts. It is also sensitive to severe page skew. So deskew must be done before applying the projection. To decide whether to apply a cut on a projection profile valley, a threshold on the width of valley is used. Instead of having a global value, the threshold might be adaptively determined by considering the width and depth of the projection profile valley, the size of nearby connected components, and the aspect ratio of current zone.

## 4.5. Performance of Text Line Segmentation

Given a set of homogeneous text zones, and the connected components enclosed by each zone, the text line segmentation partitions the components into a set of text lines. In this experiment, the ground truth text zones are used as the input. The parameters before and after the optimization are given in Table 8. The numbers and percentages of miss, false, correct, splitting, merging, and spurious detections of the text line extraction algorithm are shown in Table 9.

The advantages of this algorithm are that it is very simple and fast, and it produces a very low misdetection rate. Table 9 shows that most of the detection errors are text lines being merged. This algorithm is sensitive to skew (global or local), smear, warping, and noise. If the intertext line spacing is very small and the superscript, subscript, or the ascender and descender of adjacent lines overlap or touch each other, the text lines are usually merged.

## 4.6. Performance of Text Block Extraction

Given a set of text lines within the same column, text block extraction partitions the text lines into a set of text blocks. In this experiment, the ground truth text lines are used as the input and the column information is assumed to be known. The default and optimized parameters are given in Table 10. The performance of the text block extraction algorithm is shown in Table 11. Its total cost is 13.76%.

This algorithm utilizes the paragraph formatting attributes as the cues for text block segmentation. Instead of the global threshold for the interline spacing and the justification, a local threshold can be adaptively determined based on the text font size within a homogeneous region.

**TABLE 7**
**Performance of the *X–Y* Cut Page Segmentation on 1600 Pages from the UW Database**

| | Total | Correct | Splitting | Merging | Miss–False | Spurious |
|---|---|---|---|---|---|---|
| Ground truth | 24216 | 21019 (86.80%) | 462 (1.91%) | 2186 (9.03%) | 245 (1.01%) | 304 (1.25%) |
| Detected | 14848 | 11346 (76.41%) | 1883 (12.68%) | 710 (4.78%) | 592 (3.99%) | 317 (2.14%) |

**TABLE 8**
**The Default and Optimized Parameters for the Text Line Segmentation**

| | $h_3$ | $h_4$ | $r_3$ | $w_5$ | $h_5$ | $r_4$ |
|---|---|---|---|---|---|---|
| Default | 10 | 500 | 5 | 3 | 3 | 50 |
| Optimized | 8.4 | 1000 | 7.4 | 41 | 20 | 44.3 |

**TABLE 9**
**Performance of the Text Line Extraction Algorithm on 1600 Pages from the UW Database**

| | Total | Correct | Splitting | Merging | Miss–False | Spurious |
|---|---|---|---|---|---|---|
| Ground truth | 105439 | 100471 (95.39%) | 124 (0.12%) | 4543 (4.31%) | 157 (0.15%) | 33 (0.03%) |
| Detected | 102494 | 100471 (98.03%) | 383 (0.37%) | 1504 (1.47%) | 4 (0.00%) | 132 (0.13%) |

**TABLE 10**
**The Default and Optimized Parameters for the Text Block Extraction**

| | $t_1$ | $t_2$ |
|---|---|---|
| Default | 20 | 30 |
| Optimized | 17.8 | 26.2 |

**TABLE 11**
**Performance of Text Block Extraction on 1600 Pages from the UW Database**

| | Total | Correct | Splitting | Merging | Miss–False | Spurious |
|---|---|---|---|---|---|---|
| Ground truth | 21738 | 16680 (76.73%) | 1670 (7.68%) | 3014 (13.87%) | 2 (0.01%) | 372 (1.71%) |
| Detected | 23302 | 16680 (71.58%) | 5191 (22.28%) | 1094 (4.69%) | 0 (0.00%) | 337 (1.45%) |

## 5. SUMMARY

This paper introduced quantitative performance metrics for the document structure analysis algorithms. 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. In the University of Washington English Document Image Database-III [10], there are 1600 English document images that come with manually edited ground truth of entity bounding boxes and their properties. These bounding boxes enclose text and nontext zones, text lines, and words. We do the performance evaluation by determining the correspondence between the ground truth document structure and the automatically computed document structure, making the comparison between two structures, and reporting the performance measures.

## REFERENCES

1. R. M. Haralick, Document image understanding: Geometric and logical layout, in *Proceedings 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, June 21–23, 1994*, pp. 385–390.
2. J. Kanai, S. V. Rice, T. A. Nartker, and G. Nagy, Automated evaluation of OCR zoning, *IEEE Trans. Pattern Anal. Mach. Intell.* **17**, 1995, 86–90.
3. S. Randriamasy and L. Vincent, Benchmarking page segmentation algorithms, in *Proceedings 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Seattle, WA, 1994*, pp. 411–416.
4. M. D. Garris, Evaluating spatial correspondence of zones in document recognition systems, in *Proc. Int. Conf. on Image Processing, Washington, DC, Oct. 1995*, Vol. 3, pp. 304–307.
5. R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*, Vol. I, Addison-Wesley, Reading, MA, 1992.
6. R. M. Haralick, Performance assessment of near-perfect machines, *Mach. Vision Appl.* **2**, 1989, 1–16.
7. S. Latifi, How can permutation be used in the evaluation of zoning algorithms, *Internat. J. Pattern Recognit. Artif. Intell.* **10**, 1996, 223–237.
8. I. T. Phillips, S. Chen, and R. M. Haralick, English document database standard, in *Proc. of the Second International Conference on Document Analysis and Recognition, Japan, October 20–22, 1993*, pp. 478–483.
9. I. T. Phillips, S. Chen, J. Ha, and R.M. Haralick, English document database design and implementation methodology, in *Proc. of the Second Annual Symposium on Document Analysis and Information Retrieval, April 26–28, 1993*, pp. 65–104.
10. I. T. Phillips, *User's Reference Manual for the UW English/Technical Document Image Database III*, UW-III English/Technical Document Image Database Manual, 1996.
11. W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*, Cambridge Univ. Press, Cambridge, UK, 1992.
12. G. N. Vanderplaats, *Numerical Optimization Techniques for Engineering Design*, McGraw-Hill, New York, 1984.
13. S. Haykin, *Neural Networks: A Comprehensive Foundation*, Macmillan College, New York, 1994.
14. A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian Data Analysis*, Chapman and Hall, London, 1995.
15. J. Liang, J. Ha, R. M. Haralick, and I. T. Phillips, Document layout structure extraction using bounding boxes of different entities, in *Proceedings Third IEEE Workshop on Applications of Computer Vision, Sarasota, FL, 1996*, pp. 278–283.
16. J. Ha, R. M. Haralick, and I. T. Phillips, Document page decomposition using bounding boxes of connected components of Black Pixels, in *Proceedings Document Recognition II, San Jose, CA, 1995* (L. M. Vincent and H. S. Baird, Eds.), pp. 140–151.
17. G. Nagy and S. Seth, Hierarchical representation of optically scanned documents, in *Proceedings Seventh ICPR, Montreal, Canada, 1984*, pp. 347–349.
18. J. Liang, I. T. Phillips, and R. M. Haralick, Performance evaluation of document layout analysis on the UW data set, in *Proceedings Document Recognition IV, San Jose, CA, 1997*, pp. 149–160.
19. J. Liang, *Document Structure Extraction and Performance Evaluation*, Ph.D. dissertation, University of Washington, 1999.