# A Statistically Based, Highly Accurate Text-line Segmentation Method

Jisheng Liang Ihsin T. Phillips† Robert M. Haralick

Department of Electrical Engineering
University of Washington
Seattle, WA 98195, U.S.A.

†Department of Computer Science
Seattle University
Seattle, WA 98122, U.S.A.

{jliang, yun, haralick}@george.ee.washington.edu

## Abstract

*This paper describes a text-line identification and segmentation technique that is probability based, where all probabilities are estimated from an extensive training set of various kind of measurements of distances between the terminal and non-terminal entities and between the text-line and the text-block entities with which the algorithm works. The off-line probabilities estimated in the training then drive all decisions in the on-line segmentation algorithm.*

*On the UW-III database of some 1600 scanned document image pages, having some 105,020 text lines, the algorithm identifies and segments 104,773 correctly, an accuracy of 99.76%.*

## 1. Introduction

Given a document image, the end result of a document segmentation algorithm, in general, produces a hierarchical structure that captures the physical layout and the logical meaning of the input document page. The top of the hierarchical structure presents the entire page, and the bottom of the structure includes all glyphs on the document. Entities in the hierarchy are associated with a set of attributes describing the nature of the entities. Most known page segmentation algorithms [1]-[5] construct the document hierarchy from level to level, up and down within the hierarchy, until the hierarchical structures are built and the segmentation criteria are satisfied. Within this model, the page segmentation problem may be considered as a series of level-construction operations. That is, given a set of entities at a certain level of hierarchy, say source_level, the goal of the level-construction operation is to construct a set of entities for another level, say target_level.

This paper formulates the document structure extraction problem, and describes the design of a text-line segmentation method which achieves an accuracy of 99.76% on the UW-III document image database. Our method consists of two major modules: (1) the off-line statistical training and, (2) the on-line text-line segmentation. We conducted an extensive training to estimate the required probabilities of vertical and horizontal distances between the terminal and non-terminal entities, and between text-lines and text-blocks. The off-line probabilities estimated in the training then drive all decisions in the on-line segmentation algorithm.

The paper is organized as follows. In Section 2, we give the statistical formulation for the document segmentation problem in general, as well as for the text-line segmentation problem. In Section 3, we give a description of our text-line segmentation algorithm. The experimental results are given in Section 4.

## 2. Document Segmentation Problem

Let $\mathcal{A}$ be the set of entities at the source_level. Let $\Pi$ be a partition of $\mathcal{A}$ and each element of the partition is an entity on target_level. Let $L$ be a set of labels that can be assigned to elements of the partition. Function $f : \Pi \rightarrow L$ associates each element of $\Pi$ with a label. $V : \wp(\mathcal{A}) \rightarrow \Lambda$ specifies measurement made on subset of $\mathcal{A}$, where $\Lambda$ is the measurement space. The segmentation problem can be formulated as follows: given initial set $\mathcal{A}$, find a partition $\Pi$ of $\mathcal{A}$, and a labeling function $f : \Pi \rightarrow L$, that maximizes the probability

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

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 de-

compose the above probability into

$$\prod_{\tau \in \Pi} P(V(\tau)|f(\tau))P(f|\Pi, \mathcal{A})P(\Pi|\mathcal{A}).$$

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

The above proposed formulation can be uniformly applied to the construction of the document hierarchy at any level, e.g., text-word, text-line, and text-block extractions, just to name a few. For example, as for text-line extraction, given a set of glyphs, the goal of the text-line extraction is to partition the glyphs into a set of text-lines, each text-line having homogeneous properties, and the text-lines' properties within the same region being similar. The text-lines' properties include the deviation of glyphs from the baseline, direction of the baseline, text-lines' height and width, etc.

Given an initial set $\mathcal{A}$, we first construct the read order of the elements of $\mathcal{A}$. Let $A = (A_1, A_2, \cdots, A_M)$ be a linearly ordered set (chain in $\mathcal{A}$) of input entities. Let $\mathcal{G} = \{Y, N\}$ be the set of grouping labels. Let $A^P$ denote a set of element pairs, such that $A^p \subset A \times A$ and $A^p = \{(A_i, A_j)|A_i, A_j \in A \text{ and } j = i+1\}$. Function $g : A^p \rightarrow \mathcal{G}$, associates each pair of adjacent elements of $A$ with a grouping label, where $g(i) = g(A_i, A_{i+1})$. Then, the partition probability $P(\Pi|A)$ can be computed as follows,

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

Therefore, the joint probability is further decomposed as

$$\prod_{\tau \in \Pi} P(V(\tau)|f(\tau))P(f|\Pi, A) \prod_{i=1}^{N-1} P(g(i)|A_i, A_{i+1}).$$

We have developed an iterative, relaxation-like method to find an optimal partition. An initial grouping is determined given the observed local spatial relationships between each pair of adjacent entities. Then, we adjust the grouping by monotonically maximizing the above probability until no improvement can be made. An implementation of this approach on the text-line segmentation is described next.

## 3. Text-line Segmentation Algorithm

Figure 1 gives an overview of the text-line segmentation algorithm.

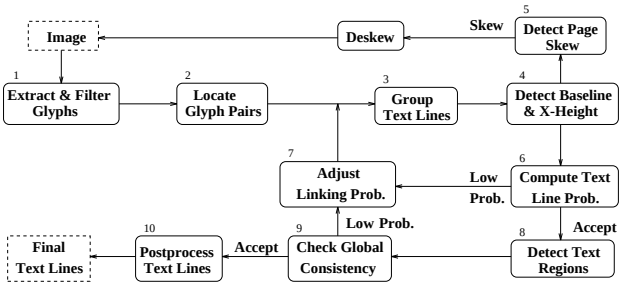


**Figure 1. Illustrates the processing steps of the text-line segmentation algorithm.**

### 3.1. Locate glyph pairs

Let $G = \{g_1, g_2, \cdots, g_M\}$ be the set of glyphs. Each glyph $g_i \in G$ is represented by a bounding box $(x, y, w, h)$. The spatial relations, $(h_a, w_a, h_b, w_b, d(a,b), o(a,b))$, between two adjacent boxes are shown in Figure 2. We

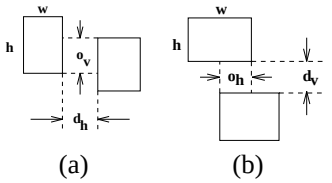


**Figure 2. Illustrates the spatial relations between two bounding boxes.**

define $g_b$ as the adjacent right neighbor of $g_a$ if $b = \arg_i \min(d_h(a,i)|i \neq a, x_i > x_a, o_v(a,i) > 0)$.

Let $g_a$ and $g_b$ be a pair of adjacent glyphs, given the observations of their heights and widths, and the distance and the overlaps between the pair: $h_a$, $w_a$, $h_b$, $w_b$, $d(a,b)$, $o(a,b)$, we compute the probability that $g_a$ and $g_b$ belong to the same text-line as:

$$P(sameline(a,b)|h_a, w_a, h_b, w_b, d(a,b), o(a,b)).$$

For each linked pair, $g_i$ and $g_{i+1}$, we associate with their link with the probability, $P(g(i))$, that indicates how probable they belong to the same text-line. At the initial grouping stage, a linking probability is equal to the "sameline" probability.

### 3.2. Base-line, x-height, and skew angle

Given a text-line $t_j = (g_1, \cdots, g_N)$, its baseline is estimated using a robust estimator. We want to fit a straight line $y(x; a, b) = a + bx$ through a set of data points, which are

the bottom-right corner of glyph boxes. The merit function to be minimized is $\sum_{i=1}^{N} |y_i - a - bx_i|$. Given a set of baseline directions $\{\theta_1, \theta_2, \cdots, \theta_P\}$, the skew angle of page is estimated as $\theta_{page} = \text{median}\{\theta_1, \theta_2, \cdots, \theta_P\}$. If skew angle $\theta_{page}$ is larger than a threshold, the page will be rotated by $-\theta_{page}$. Then, the process is repeated from the beginning.

For each given text-line $t_j$ and the estimated baseline $(a, b)$, we compute the absolute deviation of glyphs from the estimated baseline $\sigma(t_j, a, b) = \sum_{i=1}^{N} |y_i - a - bx_i|$. The x-height of a text-line is estimated by taking the median of the distance from the top-left corner of each glyph box to the baseline $xh(t_j) = \text{median}\{d(x_i, y_i, a, b) | 1 \le i \le N\}$.

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

$$P(xh(t_j), \sigma(t_j, a, b)) | \text{textline}(t_j)).$$

And this probability is used to update the linking probability between each pair of adjacent glyphs, $g_i, g_{i+1} \in t_j$,

$$\begin{aligned} P(g(i)) \propto\ & P(\text{sameline}(i, i+1) | g_i, g_{i+1}) \\ \times\ & P(xh(t_j), \sigma(t_j, a, b)) | \text{textline}(t_j)). \end{aligned}$$

### 3.3. Text-block formation

Given a set of text-line bounding boxes $T = \{t_1, t_2, \cdots, t_P\}$, our goal is to group them into a set of horizontal text-regions $R = \{R_1, R_2, \cdots, R_Q\}$. Let $(x, y, w, h)$ represent the bounding box of the text-line $t_j \in T$. The horizontal projection of $t_j$ is defined as

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

The vertical projections of the left, center, and right edge of $t_j$ are defined as:

$$\begin{aligned} C_{left}[k] &= C_{left}[k] + 1, k = x \\ C_{center}[k] &= C_{center}[k] + 1, k = x + w/2 \\ C_{right}[k] &= C_{right}[k] + 1, k = x + w \end{aligned}$$

We group text-lines into text-blocks by finding the dominant edge clusters from the projection profiles.

### 3.4. Text-line splitting and merging

Given the observations on a text-line $t = (g_1, g_2, \cdots, g_N)$ and its neighbors $N(t)$ within the same block, we compute the probability that $t$ is vertically consistent, or needs to be merged or split:

$$P(\text{v-consistent}(t) | h(t), h_N(t), h_l(g), h_N(g)),$$

where $h(t)$ is the height of $t$, $h_N(t)$ is the median of text-line height in $N(t)$, $h_t(g)$ is the median height of glyphs in $t$, and $h_N(g)$ is the median height of glyphs in $N(t)$. Then, we can update the probability that a pair of adjacent glyphs, $g_i$ and $g_{i+1}$, belong to the same line:

$$P(g(i)) \propto P(\text{sameline}(i, i+1) | g_i, g_{i+1}) P(\text{v-consistent}(t_j)),$$

where $g_i, g_{i+1} \in t_j$.

Given a pair of adjacent text-lines $t_j$ and $t_k$ within a same block or different blocks, we can update the linking probability between a pair of glyphs $g_i, g_{i+1} \in t_j \cup t_k$:

$$\begin{aligned} & P(g(i)) \\ &= P(\text{sameline}(i, i+1) | g_i, g_{i+1}, \text{sameblock}(i, i+1)) \\ &\propto P(g_i, g_{i+1} | \text{sameline}(i, i+1)) \\ &\quad \times P(\text{sameblock}(i, i+1) | \text{sameline}(i, i+1)) \end{aligned} \quad (1)$$

## 4. Experimental Results

Discrete lookup tables are used to represent the estimated joint and conditional probabilities used at each of the algorithm decision steps. We first quantize the value of each variable into a finite number of mutually exclusive states. If $A$ is a variable with states $a_1, \cdots, a_n$, then $P(A)$ is a probability distribution over these states: $P(A) = (x_1, \cdots, x_n)$ where $x_i \ge 0$ and $\sum_{i=1}^{n} x_i = 1$. Here, $x_i$ is the probability of $A$ being in state $a_i$. If the variable $B$ has states $b_1, \cdots, b_m$, then $P(A|B)$ is an $n \times m$ table containing numbers $P(a_i|b_j)$. $P(A, B)$, the joint probability for the variables $A$ and $B$, is also an $n \times m$ table. It consists of a probability for each configuration $(a_i, b_j)$.

We applied our text-line extraction algorithm to the total of 1600 images from the UW-III Document Image Database [6]. The numbers and percentages of miss, false, correct, splitting, merging and spurious detections are shown in Table 1. Of the 105,020 ground truth text-lines, $99.76\%$ of them are correctly detected, and $0.08\%$ and $0.07\%$ of lines are split or merged, respectively. Most of the missing errors are due to the rotated text.

**Table 1. Performance of the text-line extraction algorithm.**

| Total | Correct | Split | Merge | Miss | Spurious |
|---|---|---|---|---|---|
| 105020 | 104773 | 80 | 78 | 79 | 10 |
| GT | 99.76% | 0.08 | 0.07 | 0.08 | 0.01 |
| 105019 | 104773 | 172 | 37 | 25 | 12 |
| DT | 99.77% | 0.16 | 0.04 | 0.02 | 0.01 |

The extracted initial text line segments by merging pairs of connected components are illustrated in Figure 3. We no-

tice some text lines are split while some are merged across different text columns. Figure 4 shows the extracted text regions by grouping the edges of text segments. Finally, the corrected text lines given the observations on text regions are shown in Figure 5.


**Figure 3. Illustrates bounding boxes of the initial text lines.**


**Figure 4. Illustrates bounding boxes of the text regions.**


**Figure 5. Illustrates bounding boxes of the corrected text lines.**

## References


[1] D.J. Ittner and H.S. Baird, Language-Free Layout analysis, *ICDAR'93*, pp. 336-340, October 1993, Tsukuba, Japan.

[2] T. Pavlidis and J. Zhou, Page Segmentation and Classification, *CVGIP, Graphical Models and Image Processing*, Vol. 54, pp. 484-496, November 1992.

[3] G. Nagy and S. Seth, Hierarchical Representation of Optically Scanned Documents, *ICPR'84*, pp. 347-349, July 1984, Montreal, Canada.

[4] J. Ha, R.M. Haralick and I. Phillips, Document Page Decomposition by the Bounding-Box Projection, *ICDAR'95*, pp. 1119-1122, August 1995, Montreal, Canada.

[5] S. Chen, R.M. Haralick and I. Phillips, Extraction of Text Lines and Text Blocks on Document Images Based on Statistical Modeling, *International Journal of Imaging Systems and Technology*, Vol. 7, No. 4, pp. 343-356, Winter, 1996.

[6] I. Phillips, *Users' Reference Manual*, CD-ROM, UW-III Document Image Database-III, 1995.