Stochastic Language Models for Style-Directed Layout Analysis of Document Images

Tapas Kanungo, *Senior Member, IEEE*, and Song Mao, *Member, IEEE*

***Abstract*—Image segmentation is an important component of any document image analysis system. While many segmentation algorithms exist in the literature, very few i) allow users to specify the physical style, and ii) incorporate user-specified style information into the algorithm's objective function that is to be minimized. We describe a segmentation algorithm that models a document's physical structure as a hierarchical structure where each node describes a region of the document using a stochastic regular grammar. The exact form of the hierarchy and the stochastic language is specified by the user, while the probabilities associated with the transitions are estimated from groundtruth data. We demonstrate the segmentation algorithm on images of bilingual dictionaries.**

***Index Terms*—Bilingual dictionaries, duration hidden Markov models, physical layout analysis, stochastic regular grammar, style-directed analysis.**

I. Introduction

OUR OBJECTIVE is to develop a generic algorithm for segmenting scanned images of printed bilingual dictionaries formatted in various styles, and in various language pairs. The need for such an algorithm arose in a project in which we are developing an end-to-end system that can rapidly create cross-language information retrieval systems for low-density languages (languages for which online text is not readily available). Bilingual dictionaries have translation of words, which is a crucial resource for building cross-language retrieval systems. Furthermore, bilingual dictionaries are also very valuable for creating speech recognition systems for any new language since dictionaries typically have pronunciations of words.

While many segmentation algorithms have been proposed in the past, very few algorithms either i) allow users to specify the physical style of the input documents or ii) use the user-specified style information for segmenting document images to optimize some criterion. A style-directed segmentation algorithm could arguably give a better performance on the class of documents represented by the style than a generic algorithm that is designed for all types of document styles. In this paper we describe a probabilistic physical layout model for representing the physical style of documents. We then use this model to design an algorithm for extracting the physical structure of the document from a given image.

This paper is organized as follows. In Section II, we provide a survey of related work. In Section III, we introduce a generative stochastic document model and describe the probabilistic physical layout model component in detail. In Section IV, we give the statement of problem and propose a document physical layout analysis algorithm based on our proposed model. A five-step performance evaluation methodology for training and evaluating physical layout analysis algorithms is presented in Section V. In Section VI, an experimental protocol is described for conducting training and evaluating experiments. In Section VII, we present our experimental results and provide a detailed discussion.

II. Literature Survey

There are many generic segmentation algorithms. Wahl *et al.* [29] proposed an algorithm that first smears the black pixels in the x and y directions and then uses intersection of the two smeared images to mark out segments. Fletcher and Kasturi [6] described a system that used rules based on collinearity, proximity, and connected component shape distributions to group text into words and phrases. Baird *et al.* [2] based their algorithm on the observation that segmenting the foreground is a dual of segmenting the background and thus detected columns of white pixels. O'Gorman [23] described a bottom-up algorithm that starts with connected components and progressively groups them into word-level and line-level tokens using proximity and angle information. Kise *et al.* [16] used a computational geometry approach. They constructed Voronoi regions for the image and associated Voronoi regions with text regions. Small regions, which are typically associated with noise or words, were pruned to have line and zone level regions. None of the above algorithms create hierarchical descriptions or allow users to specify document structure information. Furthermore, they do not provide methods for estimating threshold parameters from groundtruth data. A rigorous empirical comparison of these algorithms can be found in Mao and Kanungo [20].

Language models have been successfully used in many areas. Formal languages represented by the grammatical rules have been used for pattern recognition [7]. N-gram and Hidden Markov Models (HMMs) are very popular language models used in speech recognition [24]. Language models such as finite state automaton have been used for text recognition at the text line level [17], [3]. Other language models such as attributed context-free grammar has been proposed for

Manuscript received July 31, 2001; revised December 11, 2002. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Aleksandra Mojsilović.

T. Kanungo is with IBM Almaden Research Center, San Jose, CA 95120 USA (e-mail: kanungo@almaden.ibm.com).

S. Mao is with the Communications Engineering Branch, U.S. National Library of Medicine, Bethesda, MD 20894 (e-mail: maosong@cfar.umd.edu; maos@mail.nlm.nih.gov).

Digital Object Identifier 10.1109/TIP.2003.811487

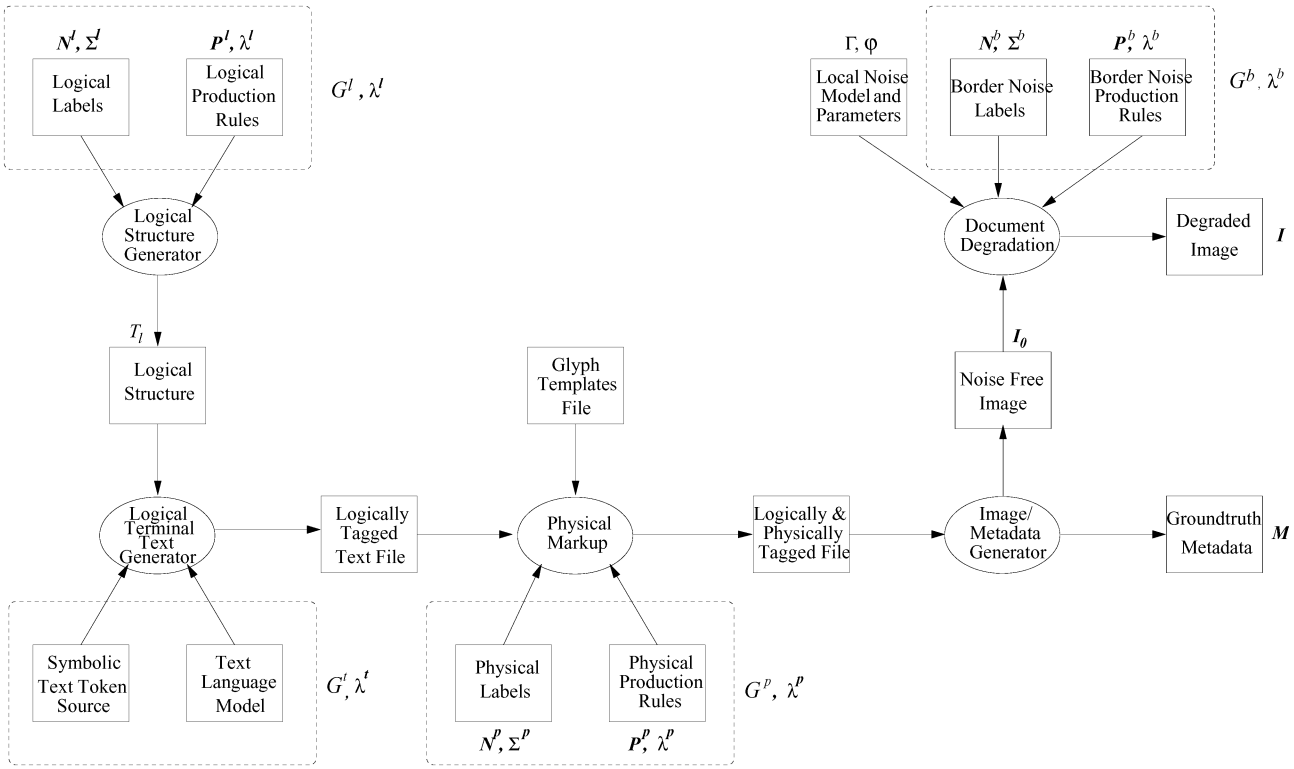


Fig. 1. Generative stochastic document model. This model simulates the generation process of document images. Document images with different physical layout styles, logical structures, and degradation levels can be obtained by varying the parameters of the model.

recognizing two-dimensional (2-D) mathematics in [4] and implemented in [9].

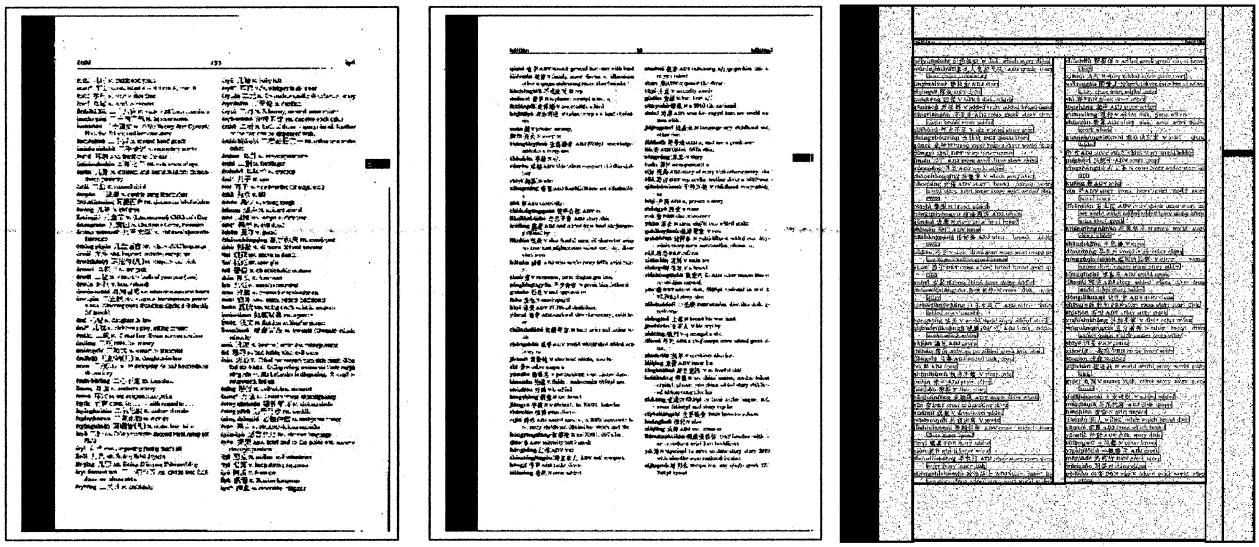
The notion of style-directed recognition, to our knowledge, has been addressed by very few researchers. Kopec and Chou [17] describe an algorithm for segmenting textlines for a column of text that is modeled using a probabilistic finite state grammar. However, their algorithm i) assumes that it is given the templates for symbols in the language, which is not the case in our problem since we need not have the character templates in a bilingual dictionary for a new language pair, ii) assumes that the columns are segmented by some other procedure, and iii) does not provide any estimation procedure for the model parameters. Tokuyasu and Chou [28] recently proposed a communication theory approach to 2-D page segmentation. They model the ideal input field by vertical and horizontal fields and use the Turbo decoding approach to estimate the 2–D field from the observations. However, the theory does not allow users to specify the width and height of lines and columns. Furthermore, the article contains very limited experimental verification of their algorithm. Both algorithms mentioned above maximize the probability of a message given the image. Krishnamoorthy *et al.* [18]. described a hierarchical segmentation algorithm that constructs a tree in which each node represents an axis-parallel region in the image. Users could specify block grammars for individual blocks. However, in the presence of noise the parsing algorithm can fail, and no parameter estimation algorithm is provided. Spitz [26] recently reported a system for style-directed recognition. While the user can specify the style interactively, the algorithm itself is a rule-based system. No objective function is minimized in either [18] or [26].

III. Generative Stochastic Document Model

Models and quantitative metrics are crucial for designing good algorithms. In particular, generative models allow an algorithm designer to perform scientific experiments that allow us to evaluate and characterize the performance of the algorithm. In this section, we describe a generative stochastic document model that is used in Section IV for designing a segmentation algorithm. Evaluation metrics and experimental protocol, which is based on this model, is described in Section V.

A. Overview of the Generative Stochastic Document Model

We model the document image generation process as a five step process: 1) First a logical structure T_l is created according to a logical structure model G^l. The logical structure of document images specifies semantic relations among logical components. The semantic relations can include the reading order and the hierarchical nesting of logical components. 2) Next, each logical component is filled with text according to a text language model G^t. 3) Physical style markup is performed according to a physical layout model G^p to specify the physical appearance and spatial relation of the logical components on a physical medium. In other words, physical style markup specifies a physical layout structure T_p. 4) A typesetting software converts the symbolic file into a noise-free image I_0 and its groundtruth metadata $\mathcal{M}$. 5) Finally, the noise-free image is degraded using a document border noise model G^b and a local noise model Γ to generate a noisy image I. These degradation steps model the noise introduced during printing, photocopying, faxing, microfilming, etc.



(a) (b) (c)

Fig. 2. (a) Real Chinese–English dictionary page, (b) a synthetically generated noise-free dictionary page, and (c) a synthetically generated noisy dictionary page with associated groundtruth using our model.

Thus our proposed generative stochastic document model M is a five-tuple $\mathcal{M} = (G^l, G^t, G^p, G^b, \Gamma)$ and associated model parameter is denoted as $\theta_{\mathcal{M}} = \{\lambda^l, \lambda^t, \lambda^p, \lambda^b, \varphi\}$. An overview of the model is illustrated in Fig. 1. In the following subsections, we describe the physical layout structure model G^p in detail and briefly introduce the degradation model Γ.

A generative stochastic document model based on stochastic attribute grammars was proposed in [5]. Their model is similar to ours in that they also use stochastic grammars to model the logical structure of documents. However, they did not have a explicit physical layout model for specifying physical layout styles of the documents. A simple channel model is used for simulating document degradation. Moreover, there is little experimental verification of effectiveness of their model.

B. Physical Layout Structure Model

Language models can be used to efficiently model syntactic or structural information. Language models are typically represented by formal grammars [1], which compactly encode structural relations in the given data. Language models can be used as both generators and recognizers of languages. Language models of different descriptive power can be used to represent syntactic structures of different complexity. Depending on the language model used, efficient parsing techniques have been used to recognize the structure of given data.

Deterministic language models can result in ambiguous parsing results when the input is probabilistic or when the grammar is ambiguous. In real applications, some grammatical rules are used more often than others. In deterministic language models, we can not learn the relative significance of grammatical rules from a given training dataset. Stochastic grammars and the associated parsing and learning techniques can be used to address the above issues. In stochastic parsing, the best parsing result is considered as the parse with the highest probability [10], [27].

The physical layout of the document image specifies the physical appearance and spatial relation of the document's physical components. While there are formal languages like regular language, context-free, and context sensitive languages [1], each of which having different levels of descriptive power, we found that for the problem of expressing the varieties of physical regions in dictionaries, regular languages are sufficient. The grammar G^p for representing the 2-D hierarchical arraignment of physical regions are described in detail in [30]. In this paper, we use flat form of the grammar. For a given region on a document image and a regular grammar G and its parameter λ, the physical layout structure of the region can be recognized by parsing an observed sequence of tokens using the given grammar. We use stochastic regular grammar $G = (N, \Sigma, P, S)$ to model the physical layout structures of document regions, where N is a set of nonterminal symbols, Σ is a set of terminal symbols, $P = \{A \rightarrow xB$ or $A \rightarrow x|A, B \in N, x \in (\Sigma)^*\}$ is a set of production rules, and S is a special start symbol which is a nonterminal symbol. For each production rule in G, we assign a probability measure. We use model parameter λ to represent probabilities of all production rules where $\lambda = \{P[A \rightarrow xB$ or $A \rightarrow x|A, B \in N, x \in (\Sigma)^*]\}$ is a set of production rule probabilities, where $\sum_{x,B} P[A \rightarrow xB$ or $A \rightarrow x|A, B \in N, x \in (\Sigma)^*] = 1$. Terminal symbols are physical components that can not be further divided, and nonterminal symbols are groups of terminal symbols. For instance, in the application of physical layout analysis of a double column journal title page, *header block, footer block, column block* are terminal symbols if they are the most basic physical components that users are interested in, whereas *body* is nonterminal symbol since it consists of two columns.

TABLE I
STOCHASTIC REGULAR GRAMMARS AND THEIR PARAMETERS FOR REPRESENTING THE PHYSICAL LAYOUT STYLES OF DICTIONARY PAGES

Level	Production Rules	Nonterminal Symbol	Terminal Symbol		Production Rule Parameters	
1	$S \rightarrow t\ h\ g\ B\ u$	B: body	t: top margin	h: header	$P_r(S)$: grammar probability	h_B, w_B: body height and width
			g: header-body gap	u: bottom margin	h_t: top margin height	h_h: header height
					h_g: header-body gap'height	h_u: bottom margin height.
2	$B \rightarrow l\ C\ g\ C\ r$	C: column	l: left margin	g: column gap	$P_r(B)$: grammar probability	w_C: column width
			r: right margin		w_l, w_r: left and right margin widths	w_g column gap
3	$C \rightarrow i\ g\ C$	C: column	i: textline	g: textline gap.	$P_r(C)$: grammar probability	h_i, h_g: textline and line gap heights

C. Local Noise Model

Local noise is introduced during printing, scanning, photocopying, faxing, microfilming of noise-free document images. We use the document degradation model proposed and validated by Kanungo *et al.* [12] as our local noise model Γ. Kanungo and Zheng [15] proposed a method for estimating the parameters of the degradation model. This degradation model has six parameters: $\varphi = (\eta, \alpha_0, \alpha, \beta_0, \beta, k)$, where η controls the flipping probability of all pixels, α_0, α controls the flipping probability of foreground pixels, β_0, β controls the flipping probability of background pixels, and k specifies the size of a disk that is used in the morphological operations. By varying these parameters, document images with different degradation levels can be generated.

Certain types of noise can be structured and concentrated at certain locations in a document image. Black streaks at the edges of document image are examples of such document noise. In this paper, we integrate the black streaks at the edges of document images into our grammatical description. Numerous document image deskewing algorithms [19] have been proposed in the past. We assume one of these algorithms has been used to deskew the images in our document datasets before their recognition since our algorithm is sensitive to document skews.

D. Modeling the Physical Layout Style of Dictionary Pages

We now use our proposed generative stochastic document model to model a Chinese-English dictionary page. In Fig. 2(a), a scanned real dictionary page from *ABC Chinese–English Dictionary* is shown, a synthetically generated clean dictionary page and a synthetically generated noisy dictionary page with associated groundtruth using our model are shown in Fig. 2(b) and (c).

The synthetically generated dictionary pages are typeset into a two-column layout and have a header on each page. The possible physical entities in a dictionary page include top margin t, bottom margin u, left margin l, right margin r, header h, body B, column C, line i and gap g. We use a grammar G^p to represent the physical layout styles of dictionary page. The description of logical model components can be found in [21]. Table I shows the grammar and associated parameters used for representing physical layout styles of dictionary pages.

In the following section, we present a recognition algorithm for deriving the physical layout structure using the stochastic regular grammar model described in this section.

IV. PHYSICAL LAYOUT STRUCTURE ANALYSIS ALGORITHM

We pose the physical layout structure analysis problem as an optimization problem. Our algorithm is based on the generative model described in Section III. We formulate the problem of physical layout analysis of document images as follows: For a given document image I, a physical layout model G^p and its estimated model parameter $\hat{\lambda}^p$, find a physical layout structure T_p^* such that

$$T_p^* = \arg\max_{T_p} P(T_p|I, G^p, \hat{\lambda}^p). \quad (1)$$

Kopec and Chou [17] proposed and investigated a similar optimization framework based on template matching and a simple channel model.

A. The Algorithm

We use a weighted finite state automaton to represent the production rule used at each level of document physical layout tree. Since each symbol in a production rule is mapped to a physical component of a document image, we assign each state in the weighted finite state automaton to a symbol in the production rule. The observations of each state are made on its corresponding physical component and are probabilistic. We compute an observation distribution for each state. We model the physical features of document physical components by state duration densities. For instance, if a physical component is large in size, the duration in the state corresponding to this physical component will be longer. State transitions signify the boundaries of physical components.

We now describe a model that represents the language using a state transition probability matrix A, an initial state distribution π, and an observation model represented by a state observation distribution matrix B. The language model is a weighted finite state automaton that is suitable for representing stochastic regular grammar. The Viterbi algorithm is used to search for the best state sequence for a given observation sequence and model parameters. While this model can be used to segment and label one-dimensional (1-D) signal simultaneously, it does not use the explicit state duration densities, which in our application represent the physical features of document physical components.

We call this model model-I. We augment the above model by a set of state duration densities. Duration Hidden Markov Models and associated estimation and recognition algorithms have been studied in the speech context [25]. We provide a detailed derivation for finding the best state and state duration for a given observation sequence and duration HMM model parameters in Appendix. We call the new model model-II. We compare the performance of the two models in Section VII.

We now formally describe the model-II algorithm. Let $\mathbf{q} = \{q_i, i = 1, 2, \ldots, N\}$ be a sequence of states, each of which corresponds to a terminal or nonterminal symbol on the right-hand-side of the production rule at current level. The terminal or nonterminal symbols in the production rule denote a document's physical components like header, body, column, etc. at current tree level. Let $\mathbf{o} = \{o_k \in \mathcal{Z}^+; 1 \leq o_k \leq M, k = 1, 2, \ldots, T\}$ be a discrete observation symbol sequence of length T. We model the physical extents of components on a document image by state duration distributions. Let $\hat{\lambda}^p = \{T, N, M, D, A, B, C, \pi\}$ be the parameter of the model-II, where T is the length of input, N is the number of states, M is the maximum value of a state observation, D is the maximum length of a state duration, $A = \{a_{ij} | a_{ii} = 0, 1 \leq i \leq N, 1 \leq j \leq N\}$ is the state transition probability matrix, $B = \{b_{im} | 1 \leq i \leq N, 1 \leq m \leq M\}$ is the state observation distribution matrix, $C = \{c_{id} | 1 \leq i \leq N, 1 \leq d \leq D\}$ is the state duration distribution matrix, π is the initial state distribution. Note that since state duration is considered explicitly within a state, the state transition probabilities to the same state, $a_{ii}, i = 1, 2, \ldots, N$ are set to 0.

The problem of finding the best segmentation is equivalent to finding the best state sequence (and state duration lengths) using model-II. That is

$$\mathbf{q}^* = \arg\max_{\mathbf{q}} P(\mathbf{q} | \hat{\lambda}^p, \mathbf{o}) \tag{2}$$

where $\mathbf{q}^* = \{q_1^*, q_2^*, \ldots, q_T^*\}$. Now define the quantity

$$\delta_t(j) = \max_{q_1 \cdots q_{t-1}} P(o_1, \ldots, o_t, q_1, q_2, \ldots, q_{t-1}, q_t = j, q_{t+1} \neq j | \hat{\lambda}^p) \tag{3}$$

which is the highest probability of a path producing observation sequence $o_1, \ldots, o_t$ and $q_1, \ldots, q_{t-1}, q_t = j$, and terminating in state j at time t. We can rewrite $\delta_t(j)$ as

$$\delta_t(j) = \max_{r, s_1, \ldots, s_{r-1}, d_1, \ldots, d_r} P\Big(o_1, \ldots, o_t, q_1 = \cdots = q_{d_1} = s_1, q_{d_1+1} = \cdots = q_{d_1+d_2} = s_2, \ldots, q_{1+\sum_{m=1}^{r-1} d_m} = \cdots = q_{\sum_{m=1}^{r} d_m} = s_r = j | \hat{\lambda}^p\Big) \tag{4}$$

where the maximum is taken subject to the constraints $\sum_{m=1}^{r} d_m = t$, $d_m \in \{1, 2, 3, ...\}$, $s_m \in \{1, 2, \ldots, N\}$, $m = 1, \ldots, r$, $s_k \neq s_{k+1}$, $k = 1, \ldots, r-1$. We use $\hat{\lambda}^p$ to denote the model parameters of the model-II. We can express $\delta_t(j)$ recursively as follows (see Appendix for proof):

$$\delta_t(j) = \max_{d \leq \min(t, D), i \neq j} \delta_{t-d}(i) \cdot \beta_{ij} \cdot c_{jd} \cdot \left[\prod_{s=t-d+1}^{t} b_{js} \right]. \tag{5}$$

In order to avoid machine precision underflow, we use the log version [25] of the recursive relation as follows:

$$\log(\delta_t(j)) = \max_{d \leq \min(t, D), i \neq j} \log(\delta_{t-d}(i)) + \log(\beta_{ij}) + \log(c_{jd}) + \left[\sum_{s=t-d+1}^{t} \log(b_{js}) \right]. \tag{6}$$

We can see that

$$\begin{aligned} P^* &= \max_{1 \leq i \leq N} [\tilde{\delta}_T(i)] = \max_{\mathbf{q}} P(\mathbf{q}, \mathbf{o} | \hat{\lambda}^p) \\ &= \max_{\mathbf{q}} P(\mathbf{q} | \mathbf{o}, \hat{\lambda}^p) \cdot P(\mathbf{o} | \hat{\lambda}^p), \\ \mathbf{q}^* &= \arg\max_{\mathbf{q}} P(\mathbf{q}, \mathbf{o} | \hat{\lambda}^p) \\ &= \arg\max_{\mathbf{q}} P(\mathbf{q} | \mathbf{o}, \hat{\lambda}^p) \cdot P(\mathbf{o} | \hat{\lambda}^p) \\ &= \arg\max_{\mathbf{q}} P(\mathbf{q} | \mathbf{o}, \hat{\lambda}^p). \end{aligned} \tag{7}$$

The last step in the derivation is justified since $P(\mathbf{o} | \hat{\lambda}^p)$ is a constant with respect to $\mathbf{q}$. This derivation result is what was required in (3). Since $\mathbf{q}^*$ determines a unique segmentation result that corresponds to a unique physical layout structure T_p for the given model $\hat{\lambda}^p$, we can rewrite (7), $T_p^* = \arg\max_{T_p} P(T_p | \mathbf{o}, \hat{\lambda}^p) = \arg\max_{T_p} P(T_p | I, G^p, \hat{\lambda}^p)$.

B. Application: Physical Layout Analysis of Dictionary Pages

The physical layout structure of the dictionary page can be represented by a grammar shown in Table I. We denote tree level 1 as page level, denote tree level 2 as column level, and denote tree level 3 as textline level. Note that we are optimizing 1-D segmentation at each level of our model separately. At each level, the segmentation is performed on either X or Y black pixel projection profile depending on the production rule used at that level. When the segmentation is completed on all levels, a hierarchical segmentation of the given document image is achieved.

By performing segmentation recursively on X or Y projection profile, we reduce a 2-D segmentation problem to a 1-D problem. We use our proposed algorithm to segment and label 1-D projection profile *simultaneously*. Our approach is similar to the method used in Krishnamoorthy *et al.* [18] in that both methods use grammatical models to analyze X or Y projection profiles of document images, and both methods therefore assume Manhattan layout of documents. In our method, we pose the analysis procedure as an optimization problem and find the optimal parsing result using stochastic grammars. In [18], the authors use deterministic grammars in their analysis and hence do not produce an optimal result. If a new representation for documents with non-Manhattan layouts is available, our modeling, analysis, and recognition methodology can be applied to them.

Let $\mathbf{h} = \{h_j | j = 1, 2, \ldots, J\}$ be the current projection profile where the value of each h_j is the black pixel count along the projection location j and J is the length of $\mathbf{h}$. We partition $\mathbf{h}$ into T strips. We compute the ratio of black pixel count in each strip and the area of the strip and quantize the ratio into observation symbols with M discrete levels. We then construct a weighted finite state automata for each tree level as shown in

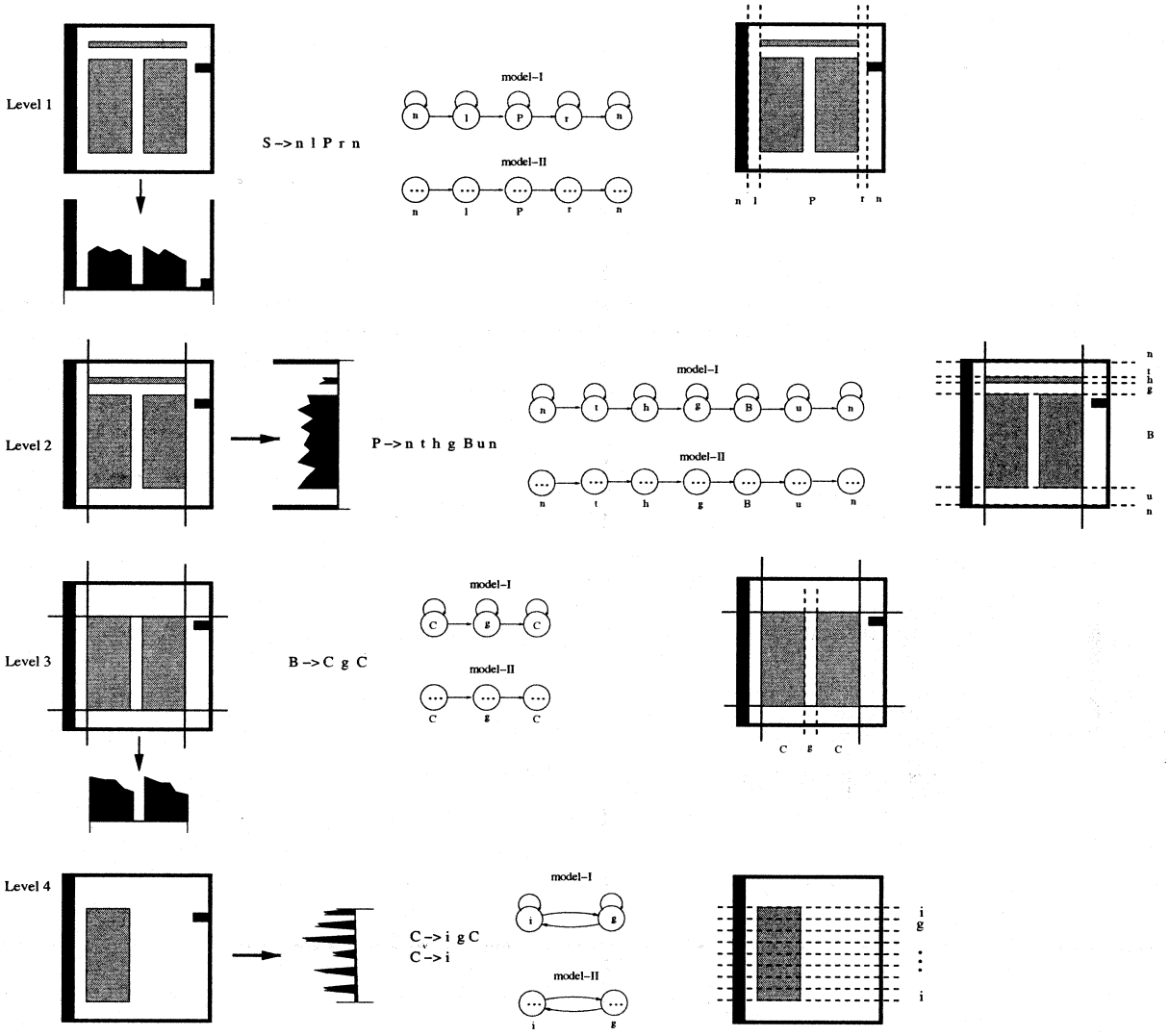


Fig. 3. One-dimensional segmentations on projection profiles using the model-I and model-II approaches at page level (level 1), main body of text level (level 2), column level (level 3) and textline (level 4). At each level, we show the document region from which projection profile is obtained, the projection profile direction and histogram, production rule and its model-I and model-II representations, and segmentation result. Segmentation takes place at state transitions in the finite state automata. Note that we incorporate noise streaks at the edges of the image into our grammars.

Fig. 3. Both models, model-I and model-II, are used to find the optimal segmentation. Each distinctive state in either one of the two models corresponds to a physical component, state transitions signify the boundaries of physical components. Therefore, 1-D segmentation can be achieved by finding state transitions in the optimal state sequence generated by the Viterbi algorithm. In the next section we compute the performance of both models.

V. Performance Evaluation Methodology

An algorithm is typically designed to perform a particular task under certain assumptions specific to the application domain. The algorithm should have satisfactory performance on the application domain and may fail on other domains where the algorithm assumptions are not valid. In this paper, we use our proposed generative stochastic document model to perform controlled experiments that allow us to 1) characterize the behavior of the algorithms, 2) quantitatively compare the performances of the algorithms, and 3) and identify their break-down points. In this section, we describe a performance evaluation methodology adapted from the method described in [20], for performing controlled experiments.

A. Methodology

We use our generative stochastic document model to automatically create large scale synthetic datasets with precise groundtruth and controlled degradation levels. Let $\mathcal{D}$ be a synthetically generated dataset using our proposed model $\mathcal{M}$. The dataset $\mathcal{D}$ contains document image and groundtruth pairs (I_i^d, G_i) where i denotes the image index and d denotes the image degradation level. The steps of our experimental methodology for characterizing the performance of document structure analysis algorithms are as follows.

1) Use the generative model $\mathcal{M}$ with a set of parameters $\theta_{\mathcal{M}}$ to generate a dataset $\mathcal{D}$.
2) Randomly partition the dataset $\mathcal{D}$ into a mutually exclusive training dataset $\mathcal{T}$ and test dataset $\mathcal{S}$. Thus $\mathcal{D} = \mathcal{T} \cup \mathcal{S}$

and $\mathcal{T} \cap \mathcal{S} = \phi$, where ϕ is the empty dataset. Each element in the datasets is an image-groundtruth pair (I_i^d, G_i) where i denotes the image index and $d \in \theta_{\mathcal{M}}$ denotes the degradation level of the image. We can control the dataset generation by adjusting the model parameter set $\theta_{\mathcal{M}}$. Hence, we can perform controlled experiments by generating datasets with different physical layout styles, logical structures and degradation levels.

3) Define a meaningful and computable performance metric $\rho(I_i^d, G_i, R_i^d)$ where I_i^d is an document image with index i and degradation level d, G_i is the groundtruth of I_i^d, and R_i^d is the structure analysis result on I_i^d.
4) For a document structure analysis algorithm A, estimate its parameter set $\hat{\mathbf{p}}_t^A$ using the training dataset T_t with degradation level t where $T_t \in \mathcal{T}$. We estimate the algorithm parameter vector $\mathbf{p}$ on different degradation level t. These estimated models are used in step 5 to study the robustness of the algorithm to document noise.
5) Evaluate the algorithm A with the estimated parameters $\hat{\mathbf{p}}_t^A$ using the test dataset $\mathcal{S}$ over different degradation levels d. Let $\Phi(d, t) = \Phi(\{\rho(G_i^d, Seg_A(I_i^d, \hat{\mathbf{p}}_t^A))|(I_i^d, G_i) \in \mathcal{S}\})$ where $\Phi(d, t)$ is a function of the estimated parameter $\hat{\mathbf{p}}_t^A$ and the performance metric ρ on each document image and groundtruth pair (I_i^d, G_i) in the test dataset $\mathcal{S}$, and $Seg_A(\cdot, \cdot)$ is the structure analysis function corresponding to A. The function Φ is defined by the user. In our case,

$$\Phi(d, t) = \frac{1}{\#\{(I_i^d, G_i) \in \mathcal{S}\}} \sum_{(I_i^d, G_i) \in \mathcal{S}} \rho(G_i, Seg_A(I_i^d, \hat{\mathbf{p}}_t^A))$$

which is the average of the performance metric $\rho(G_i^d, Seg_A(I_i^d, \hat{\mathbf{p}}_t^A))$ on each document image and groundtruth pair (I_i^d, G_i) with degradation level d in the test dataset $\mathcal{S}$.
6) Perform error analysis in different error categories over different degradation levels on both the training and test datasets to identify/hypothesize why the algorithms perform at the respective levels.

B. Performance Metric and Error Measurements

In this section, we provide the definitions of a performance metric and a set of error measures based on set theory and mathematical morphology [8]. Currently, the performance metric and the error measures are based on textlines, i.e., they evaluate the document structure analysis result only at the textline level.

Let $T_X, T_Y \in \mathcal{Z}^+ \cup \{0\}$ be the two horizontal and vertical length thresholds in number of pixels that determine if the overlap between a groundtruth textline and a segmented line is significant or not. T_X and T_Y are defined as $T_X = \min\{\texttt{HPIX}, (100 - \texttt{HTOL}) \cdot h/100\}$ and $T_Y = \min\{\texttt{VPIX}, (100 - \texttt{VTOL}) \cdot v/100\}$, where HPIX and VPIX are two thresholds in pixels, HTOL and VTOL are two thresholds in percentage, and h, v are the height and width of the groundtruth textline. Let T_V be a threshold in percentage that determines if a groundtruth textline is segmented with excessive vertical margin or not. In our experiments, we set the thresholds as $\texttt{HPIX} = 15$, $\texttt{VPIX} = 9$, $\texttt{HTOL} = 85$, $\texttt{VTOL} = 75$, and $T_V = 20$. Let $\{l^G \in \mathcal{L}\}$ be a set of groundtruth textlines. Let $E(T_X, T_Y, l^G) = \{e \in Z^2 | -T_X \le X(e) \le T_X, -T_Y \le Y(e) \le T_Y\}$ be a rectangle centered at $(0, 0)$ with a width of $2 \cdot T_X + 1$ pixels, and a height of $2 \cdot T_Y + 1$ pixels where $X(\cdot)$ and $Y(\cdot)$ denote the X and Y coordinates of the argument, respectively.

We now define two morphological operations: dilation and erosion [8]. Let $A, B \subseteq Z^2$. Morphological *dilation* of A by B is denoted by $A \oplus B$ and is defined as $A \oplus B = \{c \in Z^2 | c = a + b \text{ for some } a \in A, b \in B\}$. Morphological *erosion* of A by B is denoted by $A \ominus B$ and is defined as $A \ominus B = \{c \in Z^2 | c + b \in A \text{ for every } b \in B\}$. Let $D(\cdot)$ define the domain of its argument, let $\{l^R \in L(R)\}$ be a set of segmented lines, we now define five types of textline errors and a performance metric (textline detection accuracy):

1) Groundtruth textlines that are mis-detected:

$$S_L = \left\{ l^G \in \mathcal{L} | (D(l^G) \ominus E(T_X, T_Y, l^G)) \subseteq (\cup_{l^R \in L(R)} D(l^R))^c \right\}.$$

2) Groundtruth textlines whose bounding boxes are cut:

$$C_L = \left\{ l^G \in \mathcal{L} | (D(l^G) \ominus E(T_X, T_Y, l^G)) \cap D(l^R) \neq \phi, (D(l^G) \ominus E(T_X, T_Y, l^G)) \cap (D(l^R))^c \neq \phi, \text{ for some } l^R \in L(R) \right\}.$$

3) Groundtruth textlines that are merged:

$$M_L = \left\{ l^G \in \mathcal{L} | \exists l^R \in L(R), r^G \in \mathcal{L} \text{ and } r^G \neq l^G, \text{ such that } (D(l^G) \ominus E(T_X, T_Y, l^G)) \cap D(l^R) \neq \phi, (D(r^G) \ominus E(T_X, T_Y, r^G)) \cap D(l^R) \neq \phi \right\}.$$

4) Noise lines that are falsely detected (false alarm):

$$F_L = \left\{ l^R \in L(R) | D(l^R) \subseteq (\cup_{l^G \in \mathcal{L}} (D(l^G) \ominus E(T_X, T_Y, l^G)))^c \right\}.$$

5) Groundtruth textlines that are segmented with excessive vertical margins (vertical margin):

$$V_L = \left\{ l^G \in \mathcal{L} | H(l^R) - H(l^G) > T_V * H(l^G)/100, D(l^G) \in ((D(l^R) \oplus E(T_X, T_Y, l^G)) \right\}.$$

Let the number of groundtruth error textlines be $\#\{S_L \cup C_L \cup M_L \cup V_L\}$ (mis-detected, cut, or merged), and let the total number of groundtruth textlines be $\#\mathcal{L}$. We define the performance metric (textline detection accuracy) $\rho(I, G, R)$ as

$$\rho(I, G, R) = \frac{\#\mathcal{L} - \#\{S_L \cup C_L \cup M_L \cup V_L\}}{\#\mathcal{L}}.$$

A more general metric definition can be found in [20]. We define mis-detection error, cut error, merge error, false-alarm error, and vertical margin error as follows:

$$E_S = \frac{\#S_L}{\#\mathcal{L}}, \quad E_C = \frac{\#C_L}{\#\{\mathcal{L} - (M_L - (C_L \cap M_L))\}},$$
$$E_M = \frac{\#M_L}{\#\mathcal{L}}, \quad E_F = \frac{\#F_L}{\#\mathcal{L}}, \quad E_V = \frac{\#V_L}{\#\mathcal{L}}.$$

TABLE II
STOCHASTIC GENERATIVE DOCUMENT MODEL AND MODEL PARAMETERS VALUES FOR REPRESENTING THE PHYSICAL LAYOUT STYLES OF DICTIONARY PAGES IN THE SYNTHETICALLY GENERATED TRAINING AND TEST DATASETS. THE VALUES OF MODEL PARAMETERS ARE MEASURED FROM REAL DICTIONARY IMAGES. NOTE THAT THE NUMBER OF LEVELS HERE IS DIFFERENT FROM THAT IN FIG. 3 SINCE BORDER NOISE STREAKS ARE INCORPORATED IN THE GRAMMAR IN FIG. 3

Level	Production Rules	Nonterminal Symbol	Terminal Symbol		Production Rule Parameter Values	
1	$S \rightarrow t\ h\ g\ B\ u$	S: page, B: body	t: top margin	h: header	$P_r(S) = 1, h_B = 8.75in,$	$w_B = 6in,$
			g: header-body gap	u: bottom margin	$h_t = 0.625in, h_h = 0.5in,$	$h_g = 0.2in, h_u = 0.925in.$
2	$B \rightarrow l\ C\ g\ C\ r$	B: body; C: column	l: left margin	g: column gap,	$P_r(B) = 1, w_C = 2.875in,$	$w_l = 1in, w_g = 0.25in, w_r = 1in.$
			r: right margin			
3	$C \rightarrow i\ g\ C, C \rightarrow i$	C: column	i: textline	g: textline gap	$P_r(C \rightarrow i\ g\ C) = 0.995,$	$P_r(C \rightarrow i) = 0.005,$
					$h_i = 10pt, h_g = normal$	

VI. EXPERIMENTAL PROTOCOL

We use our proposed stochastic generative document model $\mathcal{M}$ to randomly generate a dataset of 160 noise-free synthetic dictionary pages with groundtruth at 300 dpi. The symbolic text source of the database is Optilex [22], a large (600K entries) machine-readable version of a Chinese-English dictionary. We randomly select 50 pages from the dataset as the training set, and consider the remaining 110 pages as the test set. We then resample the dataset and its groundtruth at 200 dpi and 400 dpi. Therefore, we create three datasets with same content but different resolutions. The parameters of the physical layout model are shown in Table II. Kanungo *et al.* [13], [14] have advocated model-based performance evaluation and break-down point identification since 1990. We use a modified performance evaluation methodology to identify break-down points of the algorithm.

The groundtruth of the training and test datasets are at the textline level. We modified the DVI2TIF software to generate textline groundtruth. Each page in the training dataset is degraded into three degradation levels using a document degradation model [12]. The pages in the test dataset are partitioned into 10 groups and pages in each group are degraded using one of 10 degradation levels.

We implemented our software with the C programming language. The compiler used is gcc-2.96. The platform is a 333 MHz PC running Linux 7.0 operating system. The model-I and model-II algorithm implementation is based on [11].

A. Algorithm Training

Each algorithm is trained on the training dataset at three degradation levels [noise-free images, images with degradation $\varphi = (0.05, 1.0, 2.0, 1.0, 1.0, 3)$, and images with degradation $\varphi = (0.09, 1.0, 2.0, 1.0, 1.0, 3)$] and at three resolutions (200 dpi, 300 dpi, and 400 dpi). Therefore, each algorithm has nine estimated model parameter sets. The algorithm parameter estimation uses the groundtruth of the dictionary pages in the training dataset, and the simple event occurrence frequency counting method. For the model-I algorithm, the parameters to be estimated are state transition probability matrix A, state observation distribution matrix B and initial state distribution π. For the model-II algorithm, in addition to the parameters of the model-I algorithm, we need to estimate state duration length distribution matrix C. We set the maximum number of observation value M to 100, and the width of observation strip W to 6 pixels at 300 dpi for level 1, 2, and 3 in Fig. 3. We set the maximum number of observation value M to 100, and the width of observation strip W to 3 pixels at 300 dpi for level 4 in Fig. 3. The parameter values of the images sampled at 200 dpi and 400 dpi are computed proportional to those for the images sampled at 300 dpi.

B. Algorithm Performance Evaluation

We evaluate the model-I algorithm and the model-II algorithm on three test datasets, each of which is sampled at a different resolution (200 dpi, 300 dpi, or 400 dpi) and each of which has 110 document images degraded at 10 different levels [$\varphi = (\eta, 1.0, 2.0, 1.0, 1.0, 3)$ with $\eta = \langle 0.01, 0.02, \ldots, 0.10 \rangle$]. In order to compare the effect of training dataset of different degradation levels and of different resolutions on the evaluation result, we use nine estimated model parameters for each algorithm in the testing procedure. Therefore, for each algorithm, we have nine evaluation results based on nine estimated algorithm model parameters. We can then compare the robustness of the model-I and model-II algorithms with respect to the image degradation levels and the image resolutions.

VII. EXPERIMENTAL RESULTS AND DISCUSSIONS

In this section we empirically characterize the training and recognition algorithms using the experimental protocol described in Section VI. We compare the model-I and the model-II physical layout analysis algorithms and analyze their errors in controlled experiments.

A. Algorithm Training Results

In Fig. 4 we show the estimated observation distribution matrix B of both algorithms, and the estimated state duration distribution matrix C of the model-II algorithm at the textline grammar level and at 300 dpi resolution.

We can see that with the increase of noise level, the B curve moves toward to the right. This is because the observation measurements for noisy images have a larger value than the same observation measurement for clean images. For the C curve, the state duration distribution corresponding to textline height originally has three peaks. The three peaks corresponds to the "x" textlines without ascenders nor descenders, the "b"

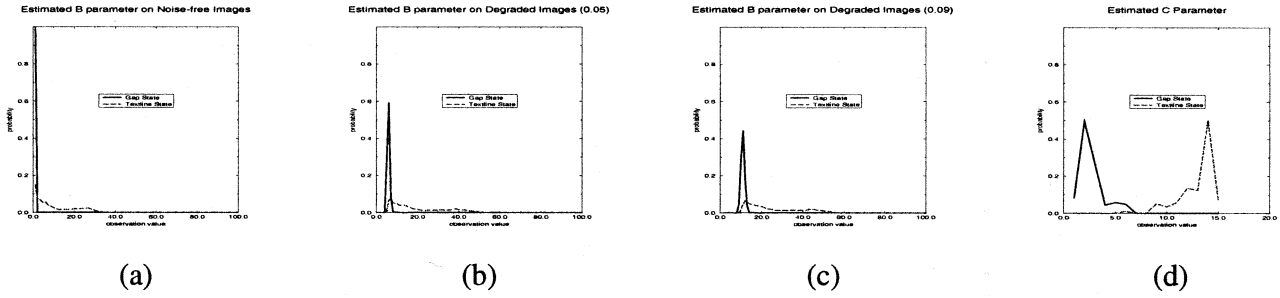


Fig. 4. (a) Estimated observation distribution matrix B on noise-free images, (b) on degraded images at degradation level 0.05, and (c) on degraded images at degradation level 0.09. State duration distribution matrix C is shown in (d). Note the B matrix is the same for the model-I and model-II algorithms, and C is for model-II only and is the same for all degradation levels. The training datasets have a resolution of 300 dpi.

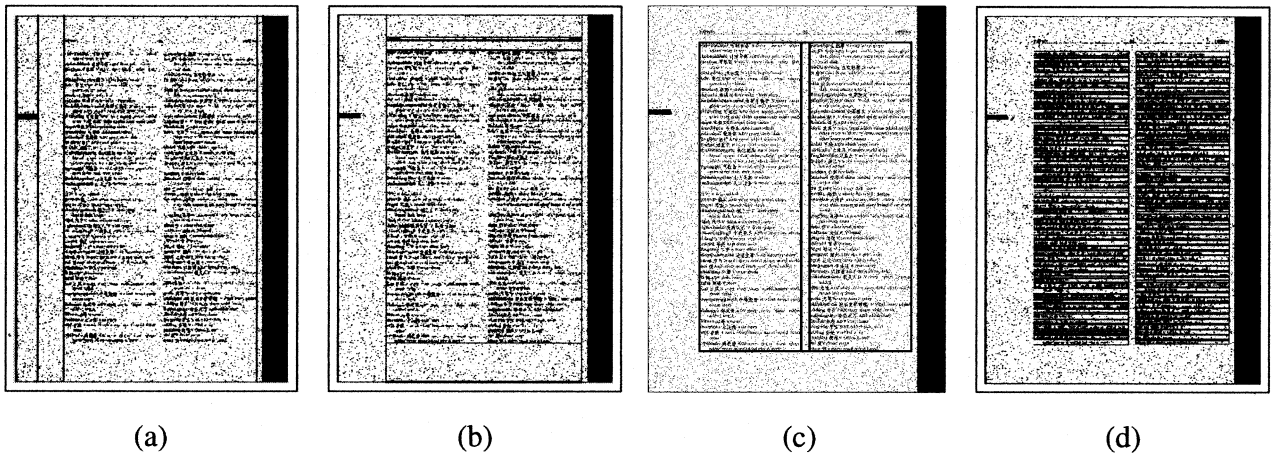


Fig. 5.(a) Hierarchical segmentation result on page level, (b) main body of text level, (c) column level, and textline level (d) using the model-II algorithm. The degradation level of the noisy image is $\varphi = (0.05,\ 1.0,\ 2.0,\ 1.0,\ 1.0,\ 3)$. The algorithm parameters are estimated on the training dataset with the same degradation level.

and "j" textlines with only ascenders or descenders, and the "bj" textlines with both ascenders and descenders.

B. Algorithm Performance Evaluation Results

An algorithm's performance depends on the class of document images used for training. If the test images are drawn from a set outside of this class, the performance of the algorithm typically deteriorates. In order to study this break-down effect, we design our test dataset to contain document images that belong to the training class as well as document images that do *not* belong to the training class. Furthermore, we train and evaluate our algorithms on images with different resolutions in order to study the sensitivity of our algorithms to image resolutions. Using nine sets of estimated algorithm parameters, we evaluate the model-I algorithm and the model-II algorithm on three test datasets each of which is sampled at a certain resolution and each of which has 110 images degraded over 10 degradation levels. There are 10 images at each degradation level and at each resolution. We report average textline detection accuracy as the performance metric for each algorithm. We also report average algorithm timing for each algorithm. Fig. 5 shows a sample of the hierarchical segmentation at four levels (page level, main body of text level, column level, and textline level) using the model-II algorithm. Fig. 6 shows the evaluation results of the model-I and model-II algorithms in terms of performance metric ρ (average textline detection accuracy).

We can see that the performance of the model-II algorithm is significantly better than that of the model-I algorithm in all cases. This is mainly due to the fact that the state duration distributions are explicitly used in the model-II algorithm. Since the state duration distributions corresponds to the physical dimensions of physical components such as header height, column width and height, textline height and gap, etc., the performance of the model-II algorithm is more robust than that of the model-I algorithm in the presence of document noise. We can also see that the two algorithms achieves optimal performance at the noise level used for algorithm training. The performance of the two algorithms deteriorates rapidly beyond the training degradation level since the algorithms begin to have inaccurate segmentation at each of the four segmentation levels.

For noise-free training images, observations tend to have smaller values and the estimated observation distribution matrix B for both models are biased toward smaller observation values. Hence, when these estimated model parameters are used to segment dictionary pages with more background noise, the both algorithms will consider some observation measurements are made from text regions even though they are actually made from gap regions, and tend to merge the text regions. Since the model-II uses explicit state duration densities, it can

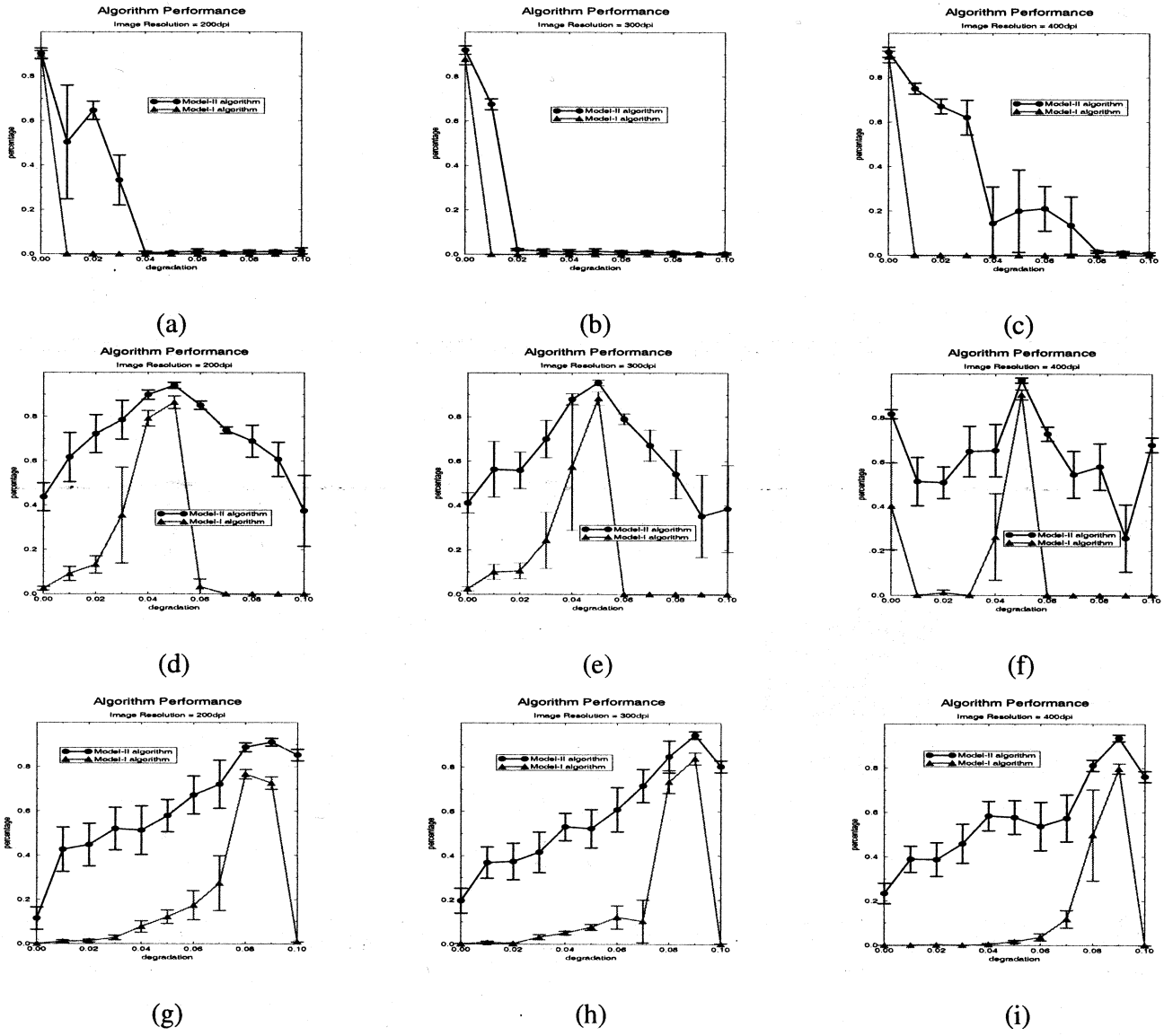


Fig. 6. Performances of the model-I and the model-II algorithms using the model parameters estimated on noise-free training images at image resolution (a) 200 dpi, (b) 300 dpi, and (c) 400 dpi, respectively; on degraded images with degradation parameter $\varphi = (0.05,\ 1.0,\ 2.0,\ 1.0,\ 1.0,\ 3)$ at image resolution (d) 200 dpi, (e) 300 dpi, and (f) 400 dpi, respectively; and on degraded images with degradation parameter $\varphi = (0.09,\ 1.0,\ 2.0,\ 1.0,\ 1.0,\ 3)$ at image resolution (g) 200 dpi, (h) 300 dpi, and (i) 400 dpi, respectively. Error bars represent 95% confidence intervals.

adjust the segmentation favorably whereas the performance of the model-I algorithm deteriorates quickly due to merging of many textlines. However, when the degradation level is too far beyond the training degradation level, the positive effect of state duration densities in model-II is overcome by erroneous estimation of observation distribution matrix B and textline detection accuracy begins to deteriorate.

When the two algorithms are trained on the noisy images, the reverse effect takes place, i.e., the algorithm parameters are estimated with a bias toward larger observation values. Hence, when these estimated model parameters are used to segment dictionary pages with less background noise, both algorithms will consider some observation measurements that are made from gap regions even though they are actually made from text regions, and tend to split the text regions. Similarly, when the images in the test dataset are too clean, the effect of state duration densities in model-II is overcome by erroneous estimation of observation distribution matrix B and textline detection accuracy begins to deteriorate.

When our algorithm is evaluated on datasets with different resolutions, the performances across different resolutions are relatively stable, which demonstrates that our algorithms are relatively insensitive to image resolutions.

In Fig. 7, we provide empirical timing analysis of the two algorithms. The timing performance of model-II is worse than that of model-I. This is because we search for an optimal segmentation over a 2-D space (state and duration) in model-II algorithm, whereas we search for an optimal segmentation over a 1-D space (state) in model-I algorithm. The algorithm timing increases with image resolution level since it takes our algorithms more time to read high resolution images than low resolution images.

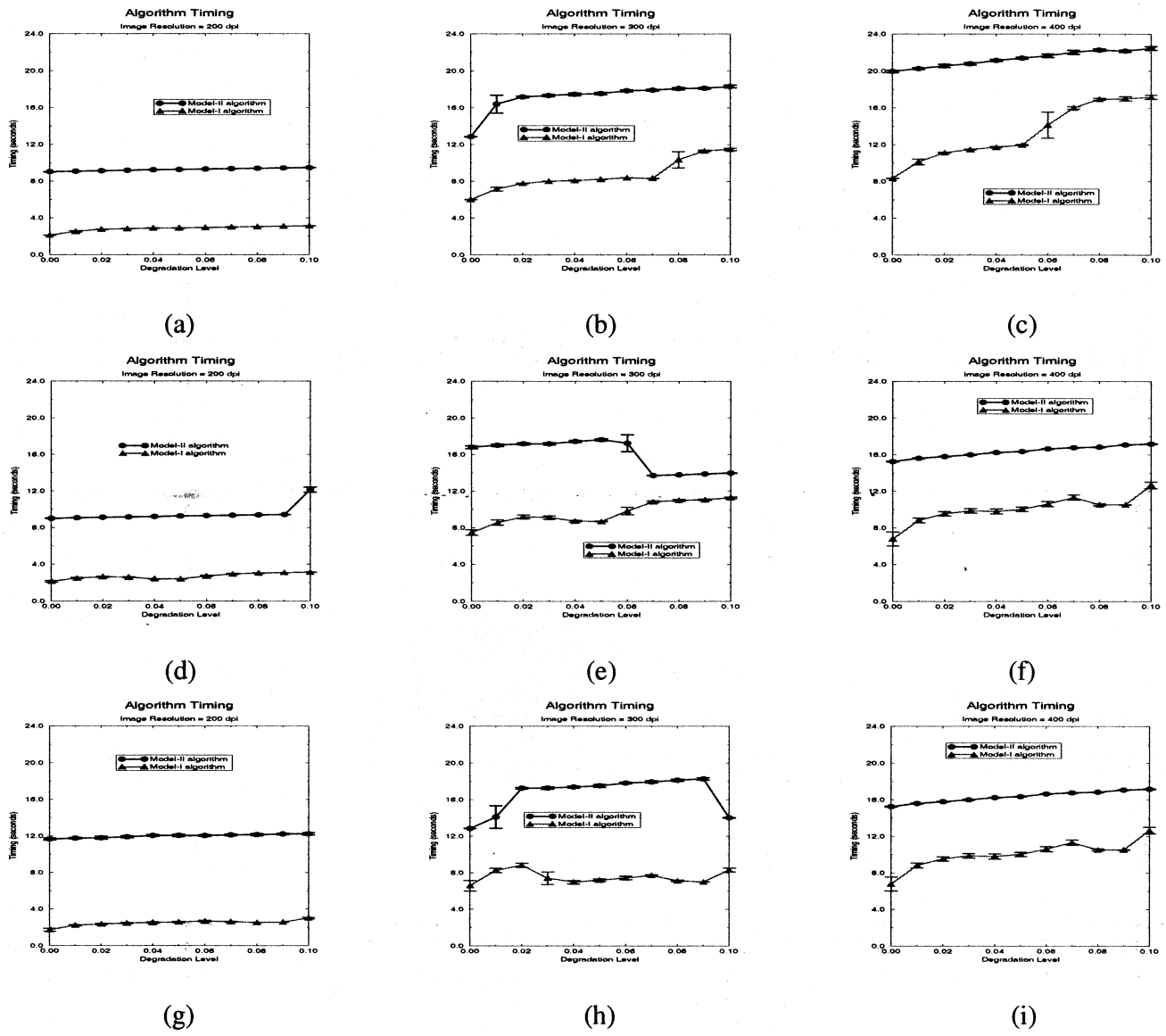


Fig. 7. Algorithm timing of the model-I and the model-II algorithms using the model parameters estimated on noise-free training images at image resolution (a) 200 dpi, (b) 300 dpi, and (c) 400 dpi, respectively; on degraded images with degradation parameter $\varphi = (0.05,\ 1.0,\ 2.0,\ 1.0,\ 1.0,\ 3)$ at image resolution (d) 200 dpi, (e) 300 dpi, and (f) 400 dpi, respectively; and on degraded images with degradation parameter $\varphi = (0.09,\ 1.0,\ 2.0,\ 1.0,\ 1.0,\ 3)$ at image resolution (g) 200 dpi, (h) 300 dpi, and (i) 400 dpi, respectively. Error bars represent 95% confidence intervals.

C. Error Analysis

In this section, we analyze the following five textline-based error categories: groundtruth textline merge error rate E_M, groundtruth textline mis-detection error rate E_S, groundtruth textline cut error rate E_C, falsely detected noisy line error rate E_F, and excessively vertical margin error rate E_V. Due to space limitation, we report the error analysis results for images at 300 dpi resolution. Fig. 8 shows the error characteristics of the two algorithms evaluated on test datasets at different degradation levels and sampled at 300 dpi.

In the case that the two algorithms are trained on noise-free training dataset, when the degradation level of the test dataset increases, the groundtruth textline merge error rate of the model-I algorithm increases drastically. Since the estimation is biased toward noise-free images, some observation measurements made on the noisy images are considered as text region even though they actually arise from gap regions. As a result the algorithm fails at higher level of the hierarchy and many text lines are merged. Moreover, at degradation level 0.01, since one of document margins is segmented as one of the columns, many false-alarm textlines are created. The fact that many groundtruth textlines are merged also results in high vertical margin error rate. When the degradation level of the test dataset increases, the model-II algorithm still has much better segmentation at higher levels of the hierarchy than the model-I algorithm. However, model-II starts to have inaccurate textline level segmentation at higher degradation levels, which results in some textlines being split and parts of some textlines being merged with adjacent textlines. Therefore, there is a drastic

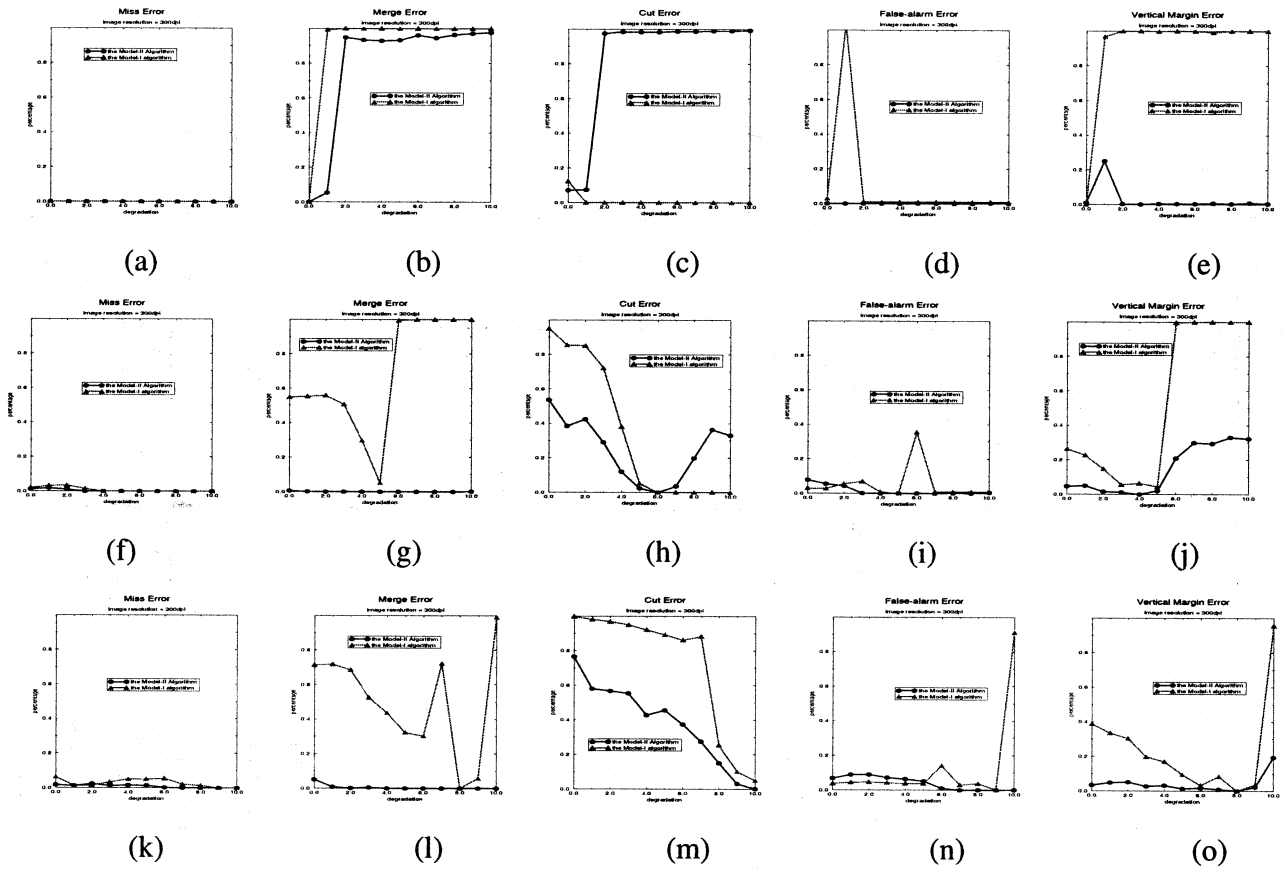


Fig. 8. Five types of the segmentation errors for each algorithm using three estimated model parameters and at image resolution of 300 dpi. The five types of error are: groundtruth textline mis-detection error rate, groundtruth textline merge error rate, groundtruth textline cut error rate, falsely detected noise lines error rate, and excessive vertical margin error rate. The errors in (a), (b), (c), (d), and (e) are the algorithm segmentation errors when algorithm parameters are estimated on noise-free training dataset. The errors in (f), (g), (h), (i), and (j) are the algorithm segmentation errors when algorithm parameters are estimated on degraded training dataset using degradation parameter $\varphi = (0.05,\ 1.0,\ 2.0,\ 1.0,\ 1.0,\ 3)$. The errors in (k), (l), (m), (n), and (o) are the algorithm segmentation errors when algorithm parameters are estimated on degraded training dataset using degradation parameter $\varphi = (0.09,\ 1.0,\ 2.0,\ 1.0,\ 1.0,\ 3)$.

increase in both merge and cut error rate. Since model-II uses explicit state duration densities, the merge error rates at smaller degradation levels are lower than those of model-I algorithm. Since many groundtruth textlines are vertically split, vertical margin error rate is very low. There is no mis-detection error for both model-I and model-II algorithms.

In the case that the two algorithms are trained on training dataset with the degradation level 0.05, the error rate results on and beyond degradation level 0.05 resemble those in the last case. This is again because algorithm parameters are trained with a bias toward the cleaner images, which, in this case, are the images with degradation level of 0.05. When degradation level is in the range of no degradation to the degradation level of 0.04, cut error rate for both algorithms are much larger at lower degradation levels than at higher degradation levels. This is because the model estimation is biased toward noisier images, which results in some observation measurement made on clean images being considered as gap regions even though they actually come from text regions. Therefore, the text regions with relatively small observation measurement values tend to be split. The model-I algorithm also has relatively large merge errors since inaccurate page level segmentation which splits parts of the two columns and also merges parts of the two columns.

In the case that the two algorithms are trained on training dataset with the degradation level 0.09, the error rate results on and beyond degradation level 0.09 are similar to the results in the first case for the same reason that is mentioned in the first case paragraph. When degradation level is in the range of no degradation to the degradation level of 0.08, the error rate results resemble those from no degradation to the degradation level of 0.04 in the second case for the similar reasons mentioned in the second case paragraph.

VIII. Summary

We have presented an end-to-end framework for analyzing the physical layout structure of document images. In this framework, we first proposed a generative document model using stochastic language models to represent document physical layout styles and document logical structures. We represented document's physical layout structure by stochastic regular grammars, and presented a new segmentation algorithm based on a probabilistic finite state automaton. In the proposed physical segmentation algorithm (called model-II algorithm), we incorporated information about the physical style layout parameters by estimating a set of state duration distributions. We found the best physical segmentation result by finding the best state sequence and state duration length sequence in the probabilistic

finite state automaton. We compared our algorithm with a baseline physical segmentation algorithm (called model-I algorithm) and found that the model II algorithm performs significantly better than the model I algorithm.

The proposed generative document model was used in designing controlled experiments in which a training dataset and a test dataset with groundtruth and of different degradation levels were synthetically created. The proposed model-II and a baseline model-I algorithm were trained on the training dataset and then evaluated and compared on the test dataset. We found the model-II algorithm is more robust to document noise then the model-I algorithm due to its consideration of state duration densities.

In future we plan to i) use better feature models for modeling state observation distributions, ii) extend the model to non-Manhattan layouts, and iii) incorporate model-based logical structure analysis of document page images.

APPENDIX
VITERBI PROCEDURE OF MODEL-II ALGORITHM

Let $\delta_t(j) = \max_{q_1 \cdots q_{t-1}} P(o_1, \ldots, o_t, q_1, \ldots, q_{t-1}, q_t = j, q_{t+1} \neq j|\hat{\lambda}^p)$, be the highest probability of a path producing observation sequence $o_1, \ldots, o_t$ and state sequence $q_1, \ldots, q_{t-1}, q_t = j$, and terminating in state j at time t. We can show that

$$\delta_t(j) = \max_{s_1,\ldots,s_{r-1},d_1,\ldots,d_r} P\Big(o_1, \ldots, o_t, q_1 = \cdots = q_{d_1} = s_1, q_{d_1+1} = \cdots = q_{d_1+d_2} = s_2, \ldots, q_{1+\sum_{m=1}^{r-1} d_m} = \cdots = q_{\sum_{m=1}^{r} d_m} = s_r = j|\hat{\lambda}^p\Big) \quad (8)$$

where the maximum is taken subject to the constraints

$$\sum_{m=1}^{r} d_m = t, \quad d_m \in \{1, 2, 3, \ldots\}, \quad s_m \in \{1, 2, \ldots, N\}, \quad m = 1, \ldots, r,\ s_k \neq s_{k+1},\ k = 1, \ldots, r-1.$$

We use $\hat{\lambda}^p$ to denote the model parameters of the model-II. We now provide a detailed derivation of a recursive relation of the quantity defined in (8). From now on, we omit qs in the following derivation for simplicity. We can rewrite (8) as

$$\delta_t(j) = \max_{d_r, i\neq j, s_1,\ldots,s_{r-2},d_1,\ldots,d_{r-1}} P(o_1, \ldots, o_t, s_1, \ldots, s_{r-1} = i, s_r = j, d_1, \ldots, d_r|\hat{\lambda}^p).$$

By Bayes' rule, we get

$$= \max_{d_r, i\neq j, s_1,\ldots,s_{r-2},d_1,\ldots,d_{r-1}} P(o_1, \ldots, o_t, s_1, \ldots, s_{r-2}, s_r = j, d_1, \ldots, d_r|s_{r-1} = i, \hat{\lambda}^p) \cdot P(s_{r-1} = i|\hat{\lambda}^p)$$

$$= \max_{d_r, i\neq j, s_1,\ldots,s_{r-2},d_1,\ldots,d_{r-1}} P(o_1, \ldots, o_{t-d_r}, s_1, \ldots, s_{r-2}, d_1, \ldots, d_{r-1}|s_{r-1} = i, \hat{\lambda}^p) \cdot P(o_{t-d_r+1}, \ldots, o_t, s_r = j, d_r|s_{r-1} = i, \hat{\lambda}^p) \cdot P(s_{r-1} = i|\hat{\lambda}^p).$$

By the "Markovity" assumption, we get

$$= \max_{d_r, i\neq j, s_1,\ldots,s_{r-2},d_1,\ldots,d_{r-1}} P(o_1, \ldots, o_{t-d_r}, s_1, \ldots, s_{r-2}, d_1, \ldots, d_{r-1}|s_{r-1} = i, \hat{\lambda}^p) \cdot P(o_{t-d_r+1}, \ldots, o_t, s_r = j, d_r|s_{r-1} = i, \hat{\lambda}^p) \cdot P(s_{r-1} = i|\hat{\lambda}^p).$$

By combining the first and last terms, and using Bayes' rule on the second term, we get

$$= \max_{d_r, i\neq j} \left(\max_{s_1,\ldots,s_{r-2},d_1,\ldots,d_{r-1}} P(o_1, \ldots, o_{t-d_r}, s_1, \ldots, s_{r-2}, s_{r-1} = i, d_1, \ldots, d_{r-1}|\hat{\lambda}^p) \cdot P(o_{t-d_r+1}, \ldots, o_t, d_r|s_r = j, s_{r-1} = i, \hat{\lambda}^p) \cdot P(s_r = j|s_{r-1} = i, \hat{\lambda}^p) \right).$$

Assume that the duration distribution of a state is independent from observations of the state. Since the observations are independent from each other given their states, we get

$$= \max_{d_r, i\neq j} \delta_{t-d_r}(i) \cdot P(s_r = j|s_{r-1} = i, \hat{\lambda}^p) \cdot P(d_r|s_r = j, \hat{\lambda}^p) \cdot \left(\prod_{s=t-d_r+1}^{t} P(o_s|s_r = j, \hat{\lambda}^p) \right)$$

$$= \max_{d_r, i\neq j} \delta_{t-d_r}(i) \cdot \beta_{ij} \cdot c_{jd_r} \cdot \left[\prod_{s=t-d_r+1}^{t} b_{js} \right].$$

For simplicity we denote d_r by d and get the following recursive relationship:

$$\delta_t(j) = \max_{d, i\neq j} \delta_{t-d}(i) \cdot \beta_{ij} \cdot c_{jd} \cdot \left[\prod_{s=t-d+1}^{t} b_{js} \right] \quad (9)$$

where $\beta_{ij} = a_{ij}$ if $d < t$, $\beta_{ij} = \pi_j$ if $d = t$ or $t = 1$, and $\delta_0(j) = 1$ for all j.

ACKNOWLEDGMENT

T. Kanungo would like to thank participants of DAS 2000 for discussions on style-directed layout recognition; P. Chou for providing relevant references; and D. Oard, P. Resnik, S. Khudanpur, and D. Yarowsky for discussions on the problem of dictionary parsing for translingual information access.

REFERENCES

[1] A. V. Aho and J. D. Ullman, *The Theory of Parsing, Translation, and Compiling*. Englewood Cliffs, NJ: Prentice-Hall, 1972.

[2] H. S. Baird, S. E. Jones, and S. J. Fortune, "Image segmentation by shape-directed covers," in *Proc. Int. Conf. Pattern Recognition*, Atlantic City, NJ, June 1990, pp. 820–825.

[3] I. Bazzi, R. Schwartz, and J. Makhoul, "An omnifont open-vocabulary OCR system for English and Arabic," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, pp. 495–504, June 1999.

[4] P. A. Chou and G. E. Kopec, "A stochastic attribute grammar model of document production and its use in document image decoding," in *Proc. Int. Soc. Optical Engineering Document Recognition*, San Jose, CA, Feb. 1995, pp. 66–73.

[5] ——, "A stochastic attribute grammar model of document production and its use in document image decoding," in *Proc. SPIE Conf. Document Recognition II*, San Jose, CA, Jan. 1995, pp. 66–73.

[6] L. A. Fletcher and R. Kasturi, "A robust algorithm for text string separation from mixed text/graphics images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, pp. 910–918, 1988.

[7] K. S. Fu, *Syntactic Methods in Pattern Recognition*. New York: Academic, 1974.

[8] R. M. Haralick, S. R. Sternberg, and X. Zhuang, "Image analysis using mathematical morphology," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, pp. 523–550, 1987.
[9] J. F. Hull, "Recognition of mathematics using a two-dimensional trainable context-free grammar," M.S. thesis, Mass. Inst. Technol., Cambridge, 1996.
[10] Y. A. Ivanov and A. F. Bobick, "Recognition of visual activities and interactions by stochastic parsing," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, pp. 852–872, 2000.
[11] T. Kanungo, "UMDHMM: A hidden Markov model toolkit," in *Extended Finite State Models of Language*, A. Kornai, Ed. Cambridge, U.K.: Cambridge Univ. Press, 1999. [Online] Available: http://www.cfar.umd.edu/~kanungo/software/software.html.
[12] T. Kanungo, R. M. Haralick, H. S. Baird, W. Stuezle, and D. Madigan, "Statistical, nonparametric methodology for document degradation model validation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, pp. 1209–1223, 2000.
[13] T. Kanungo, M. Y. Jaisimha, R. M. Haralick, and J. Palmer, "A methodology for characterization of a line detection algorithm," in *Proc. SPIE Symp. Advances in Intelligent Systems*, Boston, MA, Nov. 1990.
[14] T. Kanungo, M. Y. Jaisimha, J. Palmer, and R. M. Haralick, "A methodology for quantitative performance evaluation of detection algorithms," *IEEE Trans. Image Processing*, vol. 4, pp. 1667–1674, 1995.
[15] T. Kanungo and Q. Zheng, "Estimation of morphological degradation model parameters," in *Int. Conf. Acoustics, Speech, and Signal Processing*, Salt Lake City, UT, May 2001, pp. 1961–1964.
[16] K. Kise, A. Sato, and M. Iwata, "Segmentation of page images using the area Voronoi diagram," *Comput. Vis. Image Understand.*, vol. 70, pp. 370–382, 1998.
[17] G. E. Kopec and P. A. Chou, "Document image decoding using Markov source models," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, pp. 602–617, June 1994.
[18] M. Krishnamoorthy, G. Nagy, S. Seth, and M. Viswanathan, "Syntactic segmentation and labeling of digitized pages from technical journals," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 737–747, July 1993.
[19] D. X. Le, G. R. Thoma, and H. Weschler, "Automated page orientation and skew angle detection for binary document images," *Pattern Recognit.*, vol. 27, pp. 1325–1344, 1994.
[20] S. Mao and T. Kanungo, "Empirical performance evaluation methodology and its application to page segmentation algorithms," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 23, pp. 242–256, 2001.
[21] ——, "Stochastic language models for automatic acquisition of lexicons from printed bilingual dictionaries," in *Proc. Document Layout Interpretation and Its Applications*, Seattle, WA, Sept. 2001.
[22] MRM Corp., "Optilex,", 1999.
[23] L. O'Gorman, "The document spectrum for page layout analysis," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 1162–1173, 1993.
[24] L. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
[25] L. R. Rabiner, B. H. Juang, S. E. Levinson, and M. M. Sondhi, "Recognition of isolated digits using hidden Markov models with continuous mixture densities," *AT&T Tech. J.*, vol. 64, pp. 1211–1234, 1985.
[26] A. L. Spitz, "Style-directed document segmentation," in *Proc. 1995 Symp. Document Image Understanding Technology*, Baltimore, MD, Apr. 2001.
[27] A. Stolcke, "An efficient probabilistic context-free parsing algorithm that computes prefix probabilities," *Comput. Ling.*, vol. 21, pp. 165–201, 1995.
[28] T. A. Tokuyasu and P. A. Chou, "Turbo recognition: a statistical approach to layout analysis," *Proc. SPIE*, Jan. 2001.
[29] F. Wahl, K. Wong, and R. Casey, "Block segmentation and text extraction in mixed text/image documents," *Graph. Mod. Image Process.*, vol. 20, pp. 375–390, 1982.
[30] S. Mao, A. Rosenfeld, and T. Kanungo, "Stochastic attributed K-D tree modeling of technical paper title pages," in *IEEE Int. Conf. Image Processing*, Barcelona, Spain, Sept. 2003, to be published.

Tapas Kanungo (SM'01) received the M.S. and Ph.D. degrees in electrical engineering from the University of Washington, Seattle, in 1990 and 1996 respectively.

He is a Research Staff Member at the IBM Almaden Research Center, San Jose, CA. Prior to joining IBM, he served as a Co-Director of the Language and Media Processing Lab at the University of Maryland, College Park, where he conducted research in the areas of document image analysis, OCR-based cross-language information retrieval, pattern recognition and computer vision. From March 1996 to October 1997, he worked at Caere Corporation, Los Gatos, CA, on their OmniPage OCR product. During the summer of 1994 he worked at Bell Labs, Murray Hill, NJ, and during the summer of 1993 he worked at the IBM Almaden Research Center, San Jose, CA. Prior to that, from 1986 to 1988, he worked on speech coding and online handwriting analysis in the Computer Science group at Tata Institute for Fundamental Research, Bombay, India. He Co-Chaired the 2001 and 2002 SPIE Conferences on Document Recognition and Retrieval and the 1999 IAPR Workshop on Multilingual OCR. He was a Co-Guest Editor of the *International Journal of Document Analysis and Recognition, Special Issue on Performance Evaluation*, and has been program committee member for several conferences.

Song Mao (M'02) received the B.E. and M.E. degrees from the Department of Precision Instrument Engineering, Tianjin University, Tianjin, China, in 1993 and 1996, respectively, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland, College Park in 1999 and 2002, respectively.

He is now a Postdoctoral Fellow at the Lister Hill National Center for Biomedical Communications, an R&D Division of the National Library of Medicine, the National Institute of Health, Bethesda, MD. He worked as a co-op student at the IBM Almaden Research Center during the spring and summer of 2001. His research interests include document image analysis, information extraction, machine learning, pattern recognition, performance evaluation, and computer vision.