Structural Mixtures for Statistical Layout Analysis

Faisal Shafait[1], Joost van Beusekom[2], Daniel Keysers[1], Thomas M. Breuel[2]

Image Understanding and Pattern Recognition (IUPR) Research Group
[1]German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, Germany
[2]Technical University of Kaiserslautern, Germany
{faisal.shafait, daniel.keysers}@dfki.de, {joost, tmb}@iupr.net

Abstract

A key limitation of current layout analysis methods is that they rely on many hard-coded assumptions about document layouts and can not adapt to new layouts for which the underlying assumptions are not satisfied. Another major drawback of these approaches is that they do not return confidence scores for their outputs. These problems pose major challenges in large scale digitization efforts where a large number of different layouts need to be handled and manual inspection of the results on each individual page is not feasible. This paper presents a novel statistical approach to layout analysis that aims at solving the abovementioned problems for Manhattan layouts. The presented approach models known page layouts as a structural mixture model. A probabilistic matching algorithm is presented that gives multiple interpretations of input layout with associated probabilities. First experiments on documents from the publicly available MARG dataset achieved below 5% error rate for geometric layout analysis.

1 Introduction and Related Work

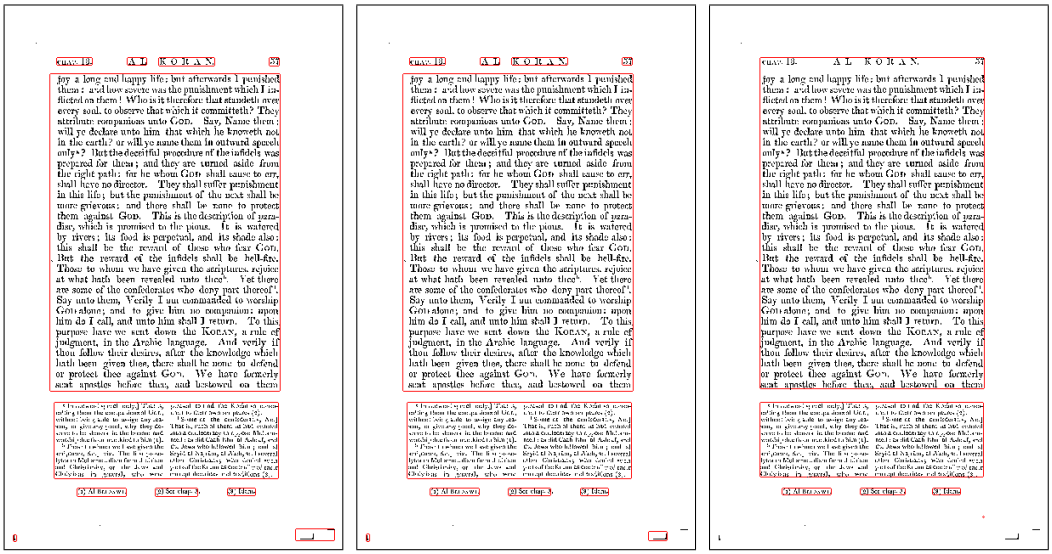
A number of different approaches for geometric layout analysis of scanned documents have been presented in the literature. Some of these approaches have come to widespread use like the X-Y Cut algorithm by Nagy et al. [14], the smearing algorithm by Wong et al. [21], the whitespace analysis algorithm by H. Baird [1], the constrained text-line extraction algorithm by T. Breuel [3], the Docstrum algorithm by O'Gorman [15], and the Voronoi algorithm by K. Kise [11]. These approaches have shown to work quite well on standard datasets like UW-III [16]. Most of these algorithms rely on many hard-coded assumptions about document layouts. For instance, a common assumption is that larger structural divisions inside a document are indicated by larger amounts of whitespace. This assumption holds for many simple document layouts, but it may break for more complex or non-stereotypical layouts.

An example of a non-stereotypical layout from the Google 1000 books dataset is shown in Figure 1. The dataset was released by Google Inc. in September 2007 and contains 1000 scanned books with hOCR-format [4] ground-truth. It contains scans of old books for which copyrights have expired. Therefore, most of the books have simple one-column page layouts. The results of applying the state-of-the-art page segmentation algorithms to this image are shown in Figure 1. Interestingly, none of the algorithms was able to segment the two-column part of the page correctly.

A closer look at the example image reveals that the gap between the two text columns is not larger than the global inter-word spacing of the document. Hence the basic assumption of most of the research algorithms and commercial systems - that larger structural divisions inside a document are indicated by larger amounts of whitespace - does not hold for this layout. This results in incorrect segmentation of the page both by research algorithms and commercial systems.

There are two traditional solutions to this problem. One solution is to manually correct the output of the page segmentation algorithm. However, it does not fit the needs for large scale digitization tasks since the user has to manually fix the results for all incorrectly handled pages, which is not feasible due to the scale of the problem. The second solution is to tune the parameters of the page segmentation algorithm such that it segments the target document correctly. However, parameter tuning is not trivial for most of the page segmentation algorithms especially for an end-user. Additionally, the assumptions made by an algorithm might prohibit it altogether to segment a particular layout correctly. Experiments on automated parameter tuning of generic layout analysis methods for the example layout in Figure 1 are part of ongoing work and will be reported in an upcoming paper.

978-0-7695-3337-7/08 $25.00 © 2008 IEEE
DOI 10.1109/DAS.2008.61



(a) Docstrum (b) Voronoi (c) X-Y Cut

Figure 1. Segmentation results of applying state-of-the-art page segmentation algorithms on an example image from the Google 1000 books dataset. None of the algorithms segmented the page correctly.

Another major issue in large scale digitization projects is to find the documents on which the page segmentation algorithm failed so that these can be presented to the operator for manual correction. The state-of-the-art layout analysis algorithms and commercial software do not give any confidence of their output. Hence the user has no clue when an algorithm fails to segment a page until he takes a look at the segmentation result of the algorithm. Manual inspection of the results of a segmentation algorithm for each scanned image becomes prohibitive in large scale applications where hundreds of thousands of pages are involved.

This paper presents a statistical approach to layout analysis aimed at solving these problems. A statistical layout analysis system is based on statistical modeling of layouts. These layout models can be learned from training data and hence can be adapted to segment non-stereotypical layouts. Secondly, the use of statistical layout models to segment a page allows to get the probability of a performed segmentation. This probability can be used as a confidence value of the output of the algorithm. Hence, the user can look at the segmentation results of only those documents for which the statistical layout analysis algorithm gives a low probability.

Some attempts to build a trainable layout analysis system have been carried out in the past. One of the first attempts in this direction was made by Gary Kopec et al. [9, 12]. They presented a communication theory approach to document recognition and called it "document image decoding". The key idea of their approach is to view document recognition as a decoding problem. The decoder estimates the message, given the observed image, by finding the a posteriori most probable path through the combined source and channel models using a Viterbi-like dynamic programming algorithm. The approach was used for direct recognition of text from scanned single-column parts of telephone yellow pages. However, this approach could not come to widespread use because it can not handle multi-column layouts.

Most of the efforts made by other researchers towards the development of trainable layout analysis systems have focused on the use of probabilistic grammars [10, 18, 20]. The latest development in the domain of grammatical modeling of document layouts is by Shilman et al. [18]. They use a discriminative grammar to model page layout instead of generative grammars as used in previous work. Their work is inspired by the advances in research on grammars, where it is shown that discriminative models are strictly more powerful than the probabilistic context-free grammars. A common limitation of modeling page layouts using stochastic grammars is that optimal geometric parsing is exponential in the number of terminal symbols. Using several geometric constraints to yield $O(n^3)$ complexity, Shilman et al. [18] were able to achieve a parsing time of 30 seconds for the task of grouping text-lines into paragraphs and text-columns (80 terminal symbols) on a 1.7GHz Pentium 4 machine. If we consider the task of grouping connected components into text-lines, where the number of terminal symbols is usually around 4000 for a typical A4 document, the running

time of the algorithm becomes a major bottleneck.

Other attempts for statistical modeling of page layout include the Markov Random Field (MRF) approach by Liang et al. [13], and the generative zone model approach of Gao et al. [8]. Liang et al. modeled the statistical relationships between the locations of characters, lines, and paragraphs as a hierarchical MRF. The model is trained by measuring different kinds of distances between terminal and non-terminal symbols on an extensive training set. Gao et al. present the approach of generating overlapping zone hypothesis by using Voronoi diagrams. Then an optimal maximum a posteriori non-overlapping zone combination is obtained using a learned generative zone model. Both these approaches are capable of learning layout information from training data. However, they require large amounts of labeled training data just to capture coarse document layout structure.

In this work, page layout is represented as a structural mixture model, where each component in the model is a layout that we are interested in. The primary focus of this work are document images with a Manhattan layout that can be represented as an X-Y tree. However, the algorithms presented here can be readily applied to non-Manhattan layouts if a suitable representation is available for them. An individual layout is represented as a hierarchical tree of horizontal or vertical whitespace cuts - that is axis-aligned whitespace rectangles that divide a particular page segment into two parts. We consider only skew corrected documents here since open-source implementations of accurate skew correction algorithms [2, 5] can be used to deskew scanned pages. A parametric model is built to model the geometric variability in position and size of corresponding whitespace cuts across different documents of the same layout. For each layout, the distribution of parameters is estimated from the training set. These learned models are then matched on the input image and the best matching model is returned with the positions of best fit and the associated probability. The details of the layout model and the matching algorithm are described in Section 2. Experimental results are presented in Section 3 followed by a conclusion in Section 4.

2 Statistical Layout Analysis

2.1 Statistical Layout Model

The first problem that needs to be addressed for designing a statistical layout analysis algorithm is the representation of page layout. Although different models for page layouts have been proposed in the literature, each model comes with its own problems. The hierarchical MRF model by Liang et al. and the generative zone model by Gao et al. require large amounts of labeled training data to obtain good results. Stochastic grammars, on the other hand, are not a natural representation of page layouts. Page layouts

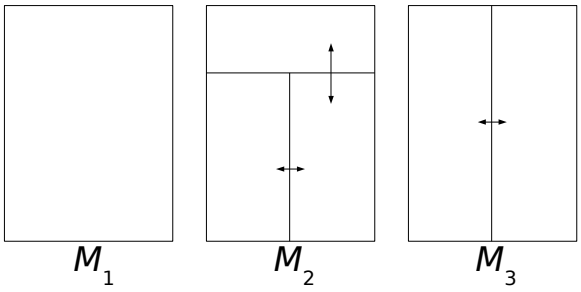


Figure 2. Representation of page layouts as a structural mixture model. This example models page layout as a mixture of three layout components. The geometric variability of these components is visualized by the arrows.

are generated by a number of typesetting rules and hence exhibit a large amount of regularity. However, parsing a page with stochastic grammars might result in page layouts that do not appear in practice.

Instead of trying to model generic page layouts, we take the approach of style-directed layout analysis [6, 10, 19]. A particular advantage of style-directed layout analysis is that it closely resembles the document generation process, hence it can obtain better performance on a specific class of documents. In contrast to previous approaches like [6, 19] that use rule-based systems to model document style, this work represents page layout as a statistical mixture model of layouts. Each layout is represented as a hierarchical X-Y tree of whitespace rectangles. A visualization of the model is shown in Figure 2. A key difference of our hierarchical document model to those published before is that we consider the page frame [17] as the top level entity instead of the complete page image. This allows us to neatly separate variations in positions of whitespaces originating from intrinsic layout variations from those introduced during the scanning process (e.g. page translation and skew).

Let a layout component be modeled as a sequence of rectangles $M = \{m_1, \ldots, m_N\}$, each defined by four parameters describing the center position (x, y), width w, and height h of the corresponding rectangle. These parameters are assumed to have independent Gaussian distributions. The sequence describes the hierarchy of model rectangles. Some important features of this model tree are:

- Each model rectangle divides a page segment into two parts. Due to this property a model rectangle will also be referred as a model cut in the work.

- The parameters of model rectangles are relative to the page segment to which they are applied.

- The first model cut is applied to the page frame. A horizontal cut divides the page frame into upper and lower parts, whereas a vertical cut divides the page frame into left and right parts. As a result of applying the model cut to the page frame, two new page segments are generated.

- The generated page segments are inserted as children of the root node in a pre-defined order. Upper or left page segment is inserted as the left child, whereas lower or right page segment is inserted as the right child.

- If a page segment is not further sub-divided, two dummy nodes are added as its children.

2.2 Statistical Model Matching

The goal of statistical model matching is to find a set of whitespace rectangles in a target document that correspond to the layout model with the highest probability. For this purpose, first a whitespace cover of the page background is extracted using the algorithm described in [3]. Then, each layout component in the structural mixture model is considered as a candidate that can explain the layout of the target document. We are interested in finding the layout model that best explains the target document and then extracting whitespaces corresponding to that best matching layout model. An illustration of this layout matching algorithm is shown in Figure 3.

Consider a layout model $M = \{m_1, \ldots, m_N\}$ consisting of N model rectangles, and a set S of K whitespace rectangles $\{w_1, \ldots, w_K\}$, where $N < K$, that constitute a whitespace cover of page background. We are interested in computing $p(W|M)$, i.e. the likelihood of observing W given M where

- $W = (w_1, \ldots, w_N)$ is an n-tuple with $w_i \in S$ and $w_i \neq w_j \; \forall i, j : \; i \neq j$

- each element of W corresponds to an element of M

Overall, we want to find the most likely subset of whitespaces:

$$\hat{W} = \arg\max_{W} p(W|M) \tag{1}$$

The likelihood of observing whitespace rectangles $W = (w_1, \ldots, w_N)$ given a layout model $M = \{m_1, \ldots, m_N\}$ can be written as

$$\begin{aligned} p(W|M) &= p(w_1, w_2, \cdots, w_N|m_1^N) \\ &= p(w_1|m_1^N)p(w_2, \cdots, w_N|w_1, m_1^N) \\ &= p(w_1|m_1^N)p(w_2|w_1, m_1^N)\cdots \\ &\qquad p(w_N|w_1, \cdots, w_{N-1}, m_1^N) \end{aligned} \tag{2}$$

where w_i is the whitespace cover rectangle that m_i has been matched on. Due to the hierarchical structure of our layout models, the likelihood of observing whitespace w_i does not depend on model cuts that are lower in the hierarchy, i.e. model cuts with indices $i+1$ to N. Hence, the first term on the right hand side of Equation 2 - $p(w_1|m_1^N)$ - can be computed as

$$\begin{aligned} p(w_1|m_1^N) &= p(w_1|m_1, m_2, \cdots, m_N) \\ &= p(w_1|m_1) \\ &= \mathcal{N}(x_1; \mu_{x_1}, \sigma_{x_1})\mathcal{N}(y_1; \mu_{y_1}, \sigma_{y_1}) \\ &\quad \mathcal{N}(w_1; \mu_{w_1}, \sigma_{w_1})\mathcal{N}(h_1; \mu_{h_1}, \sigma_{h_1}) \end{aligned} \tag{3}$$

Similarly, other terms in Equation 2 can be written as:

$$p(w_j|w_1^{j-1}, m_1^N) = p(w_j|w_1^{j-1}, m_1^j) \tag{4}$$

The dependency between a whitespace cut w_j and its ancestors is modeled by the hierarchy of the tree. The ancestors of w_j define the page segment to which the cut w_j is to be applied. Since the coordinates of whitespace cuts are computed relative to the page segment to which they are applied, these need to be recomputed based on the current page segment. This is done by first intersecting the whitespace w_j with the current page segment to trim its part extending beyond that segment, and then normalizing its coordinates with the page segment's width or height (x-center and width are divided by the page segment width, whereas y-center and height are divided by page segment height). The likelihood of the updated whitespace can then be simply computed by using Equation 3.

Using Equations 3 and 4 in Equation 2 gives the likelihood of matching a particular combination of whitespaces to the layout model. The main challenge then is to find the global maximum in Equation 1. This is a combinatorial optimization problem and brute-force search to find the globally optimal solution is not practically possible. In this work, A* search is employed to find the globally optimal combination of whitespaces that best matches the layout model. Using A* search, mean running time of matching one layout model to an image is less than one second on a 2GHz AMD Athlon machine running Linux. A hint from the implementation point of view is that when using double precision floating point numbers, the likelihood in Equation 3 goes to zero when the value inside any of the exponents gets larger than 746. Hence large portions of search space that do not fit the layout model are quickly discarded by the search algorithm.

2.3 Learning Model Parameters

An important aspect of the presented statistical layout analysis approach is that it can be trained without the need

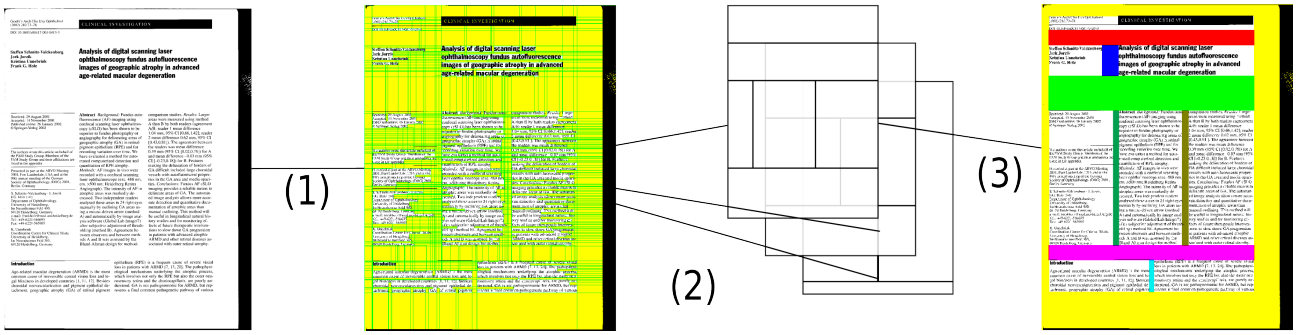


Figure 3. Overview of the statistical layout matching algorithm. First, a whitespace cover of page background is extracted, as depicted by the yellow rectangles. Then, the whitespace rectangles are matched to model rectangles for different layout model components. Finally, the best fitting model and the corresponding whitespaces are extracted.

for page segmentation ground-truth. Learning layout models from a set of training images can be done in two steps.

In the first step, the goal is to find out the structure of layout model components. This step is done by grouping documents of the same layout together and defining a structural layout model for each layout. In the present work, this task is done manually. First, the user selects documents with the same layout. Then, a structure layout model is built from one document of that layout with the help of an interactive GUI that is specifically designed for that purpose.

In the second step, the goal is to learn geometric variability of the structural layout models built in the first step. For this purpose, an EM-like training algorithm is used. Consider training images $\{1, 2, \cdots, T\}$. The total quality of matching a layout model on this training set can be computed as:

$$q = -\sum_{i=1}^{T} \log p_i(\hat{W}|M) \tag{5}$$

The training algorithm tries to minimize this quantity iteratively. An outline of the training procedure is as follows:

1. Initialize model parameters to some fixed values. Set mean values to the attributes of corresponding whitespaces selected by the user, and variance to some small arbitrary values.

2. Compute $q^{(0)}$ for training set using initial model by matching the initial model to all documents in the training set.

3. The matching result gives a set of whitespace rectangles for each training image that best match the model rectangles. Compute model parameters using maximum likelihood estimation from the obtained whitespaces.

4. Compute $q^{(1)}$ for training set using updated model parameters

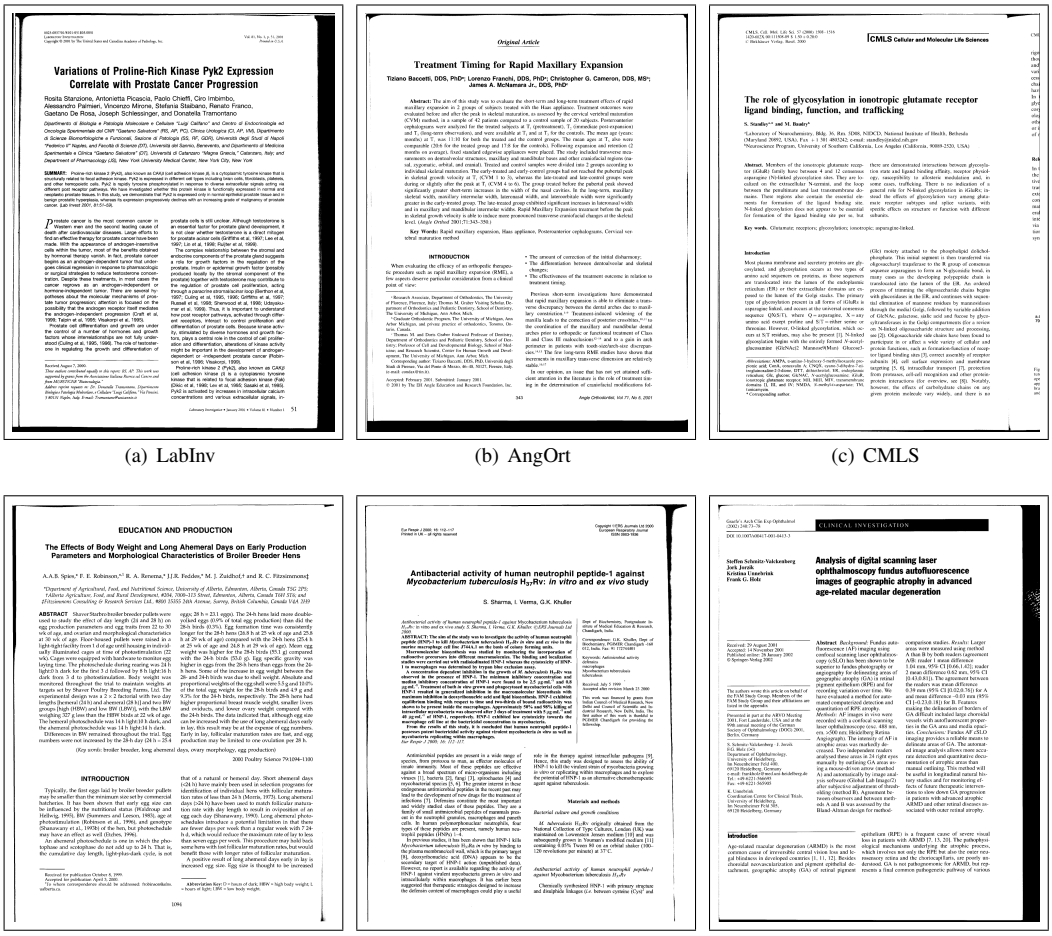
5. If $q^{(t)} \geq q^{(t-1)}$, then terminate; otherwise continue at Step 3

3 Experiments and Results

The statistical layout analysis algorithm presented in this paper exhibits several key properties that are essential for layout analysis tasks in large scale application. To evaluate the performance of the algorithm, a subset of the publicly available MARG dataset [7] was chosen. The MARG dataset is naturally suitable for this purpose since it was developed as a part of the efforts made in digitizing the US National Library of Medicine. Therefore, it contains a large variety of journal layouts with several examples of title pages from each journal. The journal layouts are categorized into nine classes based on the geometric arrangement of logical page blocks (title, author, affiliation, abstract). Since this classification is made based on logical layout elements, layouts from two different classes might look identical for geometric layout analysis. Secondly, for two journals belonging to the same class, geometric page layout might differ a lot.

In this work, six journals were chosen from the MARG dataset that had different geometric layouts of the page. These journals are:

- Laboratory Investigations (LabInv)
- Angle Orthodontist (AngOrt)
- Cellular and Molecular Life Sciences (CMLS)
- Poultry Science (PouSci)



(a) LabInv (b) AngOrt (c) CMLS

(d) PouSci (e) ERS (f) SCC

Figure 4. Example images from each of the six journals used in the experiments.

- European Respiratory Journal (ERS)
- Supportive Care in Cancer (SCC)

An example image from each of these journals is shown in Figure 4. There are 142 images of these journals collectively in the MARG dataset. The number of samples per class are too small for a reasonable training of the layout model of that class. Therefore, 1158 more samples from these journals were obtained through the central library of Technical University of Kaiserslautern. The PDF files obtained from the library were rendered as images at 300-dpi. After obtaining a reasonably sized dataset, preprocessing steps (binarization, skew correction, border noise removal) were performed where necessary using open-source tools part of the OCRopus OCR system [5]. The dataset was then partitioned into five parts to separate training and test sets. Five fold cross-validation was then used in all experiments to obtain reliable results.

Three experiments were then designed to evaluate the performance of the algorithm in three different use-cases. The first experiment, described in Section 3.1, aims at testing if the obtained page segmentation is correct, given a document image and its layout model. The experiment presented in Section 3.2 tests the ability of the proposed approach to find the correct model for an unknown document image type. Finally, Section 3.3 shows the results of the new approach on the subset of the MARG dataset.

3.1 Experiment 1

In this experiment, the performance of the statistical model matching approach is tested on synthetic document images where the model is known. Thus, the method tries to match the correct model to the document image. The most likely segmentation of the page according to the model is obtained. A segmentation is considered correct if the resulting segmentation is the same as the canonical text to block mapping, grouping logical text blocks together. If a seg-

mentation maps text of different blocks together (e.g. text lines from the abstract together with text lines from the title), the segmentation is considered wrong.

Total accuracy for this test is 99.6%. In total 1153 samples of 1158 where segmented correctly. Four out of five errors were made on documents from the ERS journal, whereas one error was made for the SCC journal.

3.2 Experiment 2

This experiment focusses on the ability of the method to find the correct model for an unknown document layout type belonging to one of the trained models. The method finds the most likely model for a given document image. A correct segmentation is again defined as being the canonically correct mapping of the text to the blocks. In this case, matching the wrong model also leads to an incorrect segmentation.

This test yielded 57.5% of correctly matched models. The confusion matrix showed that the simple model of LabInv journal matched many documents with more complex layouts. A closer investigation of this problem showed that if a layout model is a subset of another layout model, the simpler model will always fit in the documents of the more complex model. Additionally, the likelihood of match defined by: $q = \log p(\hat{W}|M)$ will usually have lower values for complex models due to additional Gaussians involved for each model cut (see Equation 2). To avoid this problem, first the quality per cut is computed by simply dividing the log-likelihood of match by the number of model cuts. Then, the per-cut quality is normalized by the complexity of the model to give complex models a better score as compared to their sub-models when both have a good matching score. The quality function thus obtained is:

$$q = -\frac{\log p(\hat{W}|M)}{N^2} \tag{6}$$

The use of this quality function increased the total accuracy to 99.6%, which means that 1153 documents out of the 1158 were segmented correctly. The confusion matrix can be found in Table 1.

3.3 Experiment 3

The test setup in this experiment is the same as for the previous one, except for the test data, which in this case consists of document images from MARG dataset. Matching layout models proceeded in two steps. First a model for the page frame was matched on the document to find its page frame. Then, the layout model was matched inside the detected page frame. Again the canonical correctness of the text to block mapping is used as correctness measure. The total accuracy for this test is 95.1% using the normalized quality of fit (Equation 6). In absolute numbers this means that 135 out of the 142 documents have been successfully segmented. The confusion matrix can be found in Table 2.

A closer look at the results showed that the errors are mainly due to complex models being fit to simple layouts. In the case of LabInv, which is a two column layout having a one column title part and a two column footer, the AngOrt model consisting of a one column title, two column main text part and a one column footer fitted well, and was chosen due to the normalization of the quality.

4 Conclusion and Outlook

This paper presented a novel statistical approach to layout analysis. The presented approach is based on top-down modeling of page layouts using a mixture of structural layout models. The geometric variability of individual layout components was modeled as a multi-variate Gaussian distribution. An algorithm for finding the globally optimal match of a layout model to a target document was presented. Initial results on documents collected by the author and on the MARG dataset showed high accuracy for geometric layout analysis. More comprehensive performance evaluations and comparison to other methods is part of on-going work and will be reported in an upcoming paper.

Acknowledgments

This work was partially funded by the BMBF (German Federal Ministry of Education and Research), project IPeT (01 IW D03).

References

[1] H. S. Baird. Background structure in document images. In H. Bunke, P. Wang, and H. S. Baird, editors, *Document Image Analysis*, pages 17–34. World Scientific, Singapore, 1994.

[2] D. S. Bloomberg, G. E. Kopec, and L. Dasari. Measuring document image skew and orientation. In *Proc. SPIE Document Recognition II*, pages 302–316, San Jose, CA, USA, Feb. 1995.

[3] T. M. Breuel. Two geometric algorithms for layout analysis. In *Proc. Document Analysis Systems*, pages 188–199, Princeton, NY, USA, Aug. 2002.

[4] T. M. Breuel. The hOCR microformat for OCR workflow and results. In *Proc. Int. Conf. on Document Analysis and Recognition*, pages 1063–1067, Curitiba, Brazil, Sep. 2007.

[5] T. M. Breuel. The OCRopus open source OCR system. In *Proc. SPIE Document Recognition and Retrieval XV*, pages 0F1–0F15, San Jose, CA, USA, Jan. 2008.

[6] A. Dengel, A. Luhn, and B. Ueberreiter. Model based segmentation and hypothesis generation for the recognition of printed documents. In *Proceedings of the SPIE-87*, pages 89–100, Cannes, France, Nov. 1987.

Journal	LabInv	AngOrt	CMLS	PouSci	ERS	SCC
LabInv	124		3			
AngOrt		169				
CMLS			213			
PouSci				224		
ERS			1		204	
SCC			1			219

Table 1. Confusion matrix for the matching using the normalized quality of fit (Equation 6). Only 5 documents are segmented using the wrong model type.

Journal	LabInv	AngOrt	CMLS	PouSci	ERS	SCC
LabInv	0	5				
AngOrt		13	1			
CMLS			38			
PouSci				25		
ERS			1		48	
SCC						11

Table 2. Confusion matrix for documents from the MARG dataset. The correct layout was found for 135 out of 142 documents.

[7] G. Ford and G. R. Thoma. Ground truth data for document image analysis. In *Symposium on Document Image Understanding and Technology*, pages 199–205, Greenbelt, MD, USA, April 2003.

[8] D. Gao, Y. Wang, H. Hindi, and M. Do. Decompose document image using integer linear programming. In *Proc. Int. Conf. on Document Analysis and Recognition*, pages 397–401, Curitiba, Brazil, Sep. 2007.

[9] A. C. Kam and G. E. Kopec. Document image decoding by heuristic search. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(9):945–950, 1996.

[10] T. Kanungo and S. Mao. Stochastic language models for style-directed layout analysis of document images. *IEEE Trans. on Image Processing*, 12(5):583–596, 2003.

[11] K. Kise, A. Sato, and M. Iwata. Segmentation of page images using the area Voronoi diagram. *Computer Vision and Image Understanding*, 70(3):370–382, 1998.

[12] G. E. Kopec and P. A. Chou. Document image decoding using markov source models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(6):602–617, 1994.

[13] J. Liang, R. M. Haralick, and I. T. Phillips. A statistically based, highly accurate text-line segmentation method. In *Proc. Int. Conf. on Document Analysis and Recognition*, pages 551–555, Bangelore, India, Sep. 1999.

[14] G. Nagy, S. Seth, and M. Viswanathan. A prototype document image analysis system for technical journals. *Computer*, 7(25):10–22, 1992.

[15] L. O'Gorman. The document spectrum for page layout analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(11):1162–1173, 1993.

[16] F. Shafait, D. Keysers, and T. M. Breuel. Performance evaluation and benchmarking of six page segmentation algorithms. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(6):941–954, 2008.

[17] F. Shafait, J. van Beusekom, D. Keysers, and T. M. Breuel. Page frame detection for marginal noise removal from scanned documents. In *SCIA 2007, Image Analysis, Proceedings*, volume 4522 of *Lecture Notes in Computer Science*, pages 651–660, Aalborg, Denmark, June 2007.

[18] M. Shilman, P. Liang, and P. Viola. Learning non-generative grammatical models for document analysis. In *Proc. Int. Conf. on Computer Vision*, pages 962–969, Beijing, China, Oct. 2005.

[19] A. L. Spitz. Style-directed document segmentation. In *Proc. Symp. Document Image Understanding Technology*, pages 195–199, Baltimore, MD, USA, Apr. 2001.

[20] T. Tokuyasu and P. A. Chou. Turbo recognition: a statistical approach to layout analysis. In *Proc. SPIE Document Recognition and Retrieval VIII*, pages 123–129, San Jose, CA, USA, Jan. 2001.

[21] K. Y. Wong, R. G. Casey, and F. M. Wahl. Document analysis system. *IBM Jour. of Research and Development*, 26(6):647–656, 1982.