Model-based Tabular Structure Detection and Recognition in Noisy Handwritten Documents

Jin Chen and Daniel Lopresti
Computer Science and Engineering
Lehigh University
Bethlehem, PA 18015, United States
{*jic207, lopresti*}@cse.lehigh.edu

Abstract

Tabular structure detection and recognition can be a valuable step in the analysis of unstructured documents. The noisy handwritten documents we try to analyze may contain pre-printed rulings as the substrate, hand-drawn rulings, machine-printed text, handwritten text, and signatures, in addition to the tabular structures which we wish to decompose into basic cells, rows, and columns. Although work has been done to machine-printed documents, noisy handwritten documents may require modified and/or new techniques. In this work, we try to detect and decompose tabular structures into 2-D grids of table cells simultaneously. First, we detect "key points" that help determine the physical and logical structure of tables. Then, we make use of the 2-D grid assumption to build grids of key points. Finally, we extract structural features for the Min-Cut/Max-Flow algorithm to recognize tabular structures. Experiments on 22 tables which contain 584 table cells show a cell precision of 100% and a cell recall of 93.3%.

1. Introduction

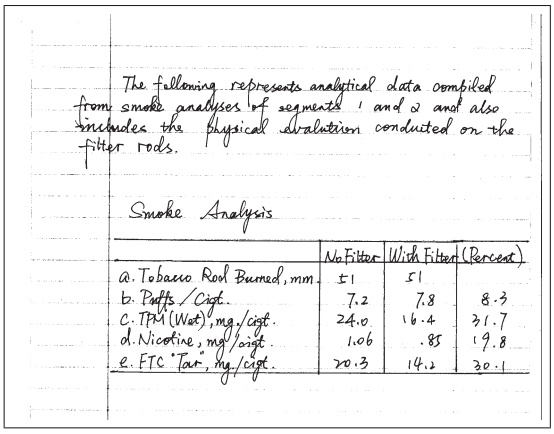
Tables are one common way in people's communication, including web pages, data spreadsheets, machine-printed documents, and handwritten ones. As an indexing scheme, tables have *physical* and *logical* structure [9, 21, 8]. Physical structure describes the locations of table components, e.g., headers, rows, columns, cells, rulings. Logical structure defines how table components connect to each other to form a set of relational n-tuples [22]. For example, a cell can be defined in the logical structure as *(Row[i],Column[j])*, and it can also be defined in the physical structure as the rectangular region in the table.

The target documents we try to analyze retain characteristics of documents from people's daily lives, where varies of components may present: machine-printed text, handwriting, pre-printed rulings, hand-drawn rulings, signatures, etc. In addition, flexible tabular structures such as empty cells call for modifying existing techniques and/or proposing new ones to handle the complexities. See Figure 1 for an example. First, it is obvious that existing ruling-based table analysis methods should take into account of the pre-printed rulings. Second, those who assume complete table rulings may fail since in this table, several horizontal table rulings are not drawn. For white-space based methods, on the other hand, some of them are expected to miscapture the space at the end of the second table row.

Hu *et al.*, summarize the problem of table analysis as two sub-problems: detection and recognition [11]. Table detection focuses on finding table regions. Laurentini and Viada use horizontal and vertical rulings as initial evidence for tables in machine-printed documents, and then employ several tests to exclude non-tabular areas [12]. Some other work does not rely on the presence of ruling lines. Hu *et al.*, introduce an *inside-space* based table detection method that does not rely on rulings and is also medium independent [11]. Our previous work tried to justify its efficacy on handwritten inputs and found out that their approach can not handle all the complexities [5]. Shafait and Smith extend the work to multi-column documents [16].

Table recognition usually assumes identified table regions and the goal is to find the physical structure and the logical structure of the table model [9, 21, 8]. Lots of work deals with machine-printed table recognition [10, 3, 7]. Gatos *et al.* [7], make use of the complete table rulings to recognize the table structure while Hirayama [10] relies on dynamic programming (DP) to align table columns. Richarz *et al.*, recently propose a

978-0-7695-4774-9/12 $26.00 © 2012 IEEE
DOI 10.1109/ICFHR.2012.233



The following represents analytical data compiled from smoke analyses of segments 1 and 2 and also includes the physical evaluation conducted on the filter rods.

Smoke Analysis

	No Filter	With Filter	(Percent)
a. Tobacco Rod Burned, mm	51	51	
b. Puffs / Cigt.	7.2	7.8	8.3
c. TPM (Wet), mg./cigt.	24.0	16.4	31.7
d. Nicotine, mg./cigt.	1.06	.85	19.8
e. FTC "Tar", mg./cigt.	20.3	14.2	30.1

Figure 1: A sample document containing a table.

method of tabular structure recognition for their semi-supervised transcription system in handwritten historical weather reports [15]. Making use of the pre-printed table substrate, they use the Hough transform to detect the horizontal and vertical rulings that constitute the tabular structures.

In this work, we try to detect and recognize tabular structures simultaneously and decompose them into 2-D grids of table cells. To focus on modeling of the tabular structures, we start the processing with manually labeled documents. Next, we detect "key points" at the intersections of white streams within text lines and between text lines. Then, we make use of hand-drawn rulings to validate key points, and then generate imaginary key points around missing cells to form regular grids. Finally, we extract structural features from the grids and employ the Min-Cut/Max-Flow algorithm to decide the most probable key points in tables.

For the rest of this paper, we first describe the annotation rules and the inputs for our experiments in Section 2. Next, we introduce the idea of key points, the rationale behind it, and an algorithm for detection in Section 3. Then, we model the table structures as 2-D grids of key points and introduce the structural features that are used in the Min-Cut/Max-Flow algorithm, in Section 4. Finally, we describe the experimental setup and results in Section 5 and conclude in Section 6.

2. Document Annotation

We start processing with manually labeled words using oriented rectangles because there is no need to reinvent the wheel: (1) there is a large amount of existing techniques to segment text lines, words (e.g., [18]); (2) pre-printed rulings lines may be detected by existing methods as well (e.g., [4]);

Using GEDI [6], we labeled rulings and words with oriented rectangles and associated "run-length encoding (RLE)" was automatically recorded, which can further be decoded into a sequence of pixel points. In addition, we grouped text within the same line to make use of correctly segmented text lines for key point detection. We labeled pre-printed rulings and hand-drawn rulings differently because the latter represents the author's intent of organizing and isolating table components from the other parts of a document.

3. Key Point Detection

The idea of key points resembles the idea of inside-space in Hu *et al.*'s work on table detection [11] where inside-space means the white space between two text blobs. However, it differs in that key points reside in between two text lines and thus offer more structural information about neighboring text lines. See Figure 2a for an example of key points. As we can see, a key point is indeed a local region in which any horizontal or vertical cuts will not affect the table cells. Thus, in the following discussion and demonstration, we use a cross sign to represent a key point – the height and the width define the local rectangular region.

One important question is what the minimum width of a key point should be? Small width will introduce many false-alarming key points since the inter-word spacing in plain text lines will be captured. On the other hand, large width will suppress the key point detection in tables. We estimate the threshold spacing $\mathcal{W}$ by assuming the inter-word spacing $Z = \{z_1, \ldots, z_n\}$ follows a bimodal distribution, where one Gaussian represents the inter-word spacing in plain text lines, and the other the spacing between table columns:

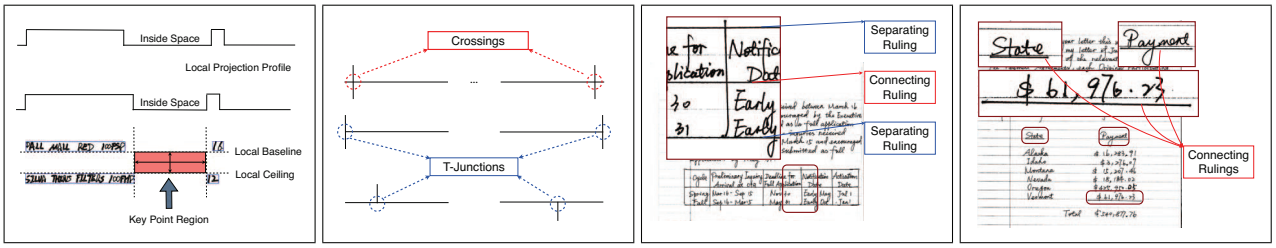
$$p(z) = \pi_1 \mathcal{N}(\mu_1, \Sigma_1) + \pi_2 \mathcal{N}(\mu_2, \Sigma_2) \qquad (1)$$

where π_i is the weight between two Gaussian distributions and $\pi_i = 1$ is set in our experiments. Given the inter-word spacings collected from each page, we employ the Expectation Maximization (EM) algorithm to find the maximum-likelihood estimates of all the parameters in μ_i and Σ_i [1]. EM is an iterative procedure with two steps. At the Expectation-step, we compute the probability λ_{ik} of sample k belonging to Gaussian i using the current model parameters:

$$\lambda_{ik} = \frac{\mathcal{N}_i(z; \mu_i, \Sigma_i)}{\sum_{j=1}^{2} \mathcal{N}_j(z; \mu_j, \Sigma_j)} \qquad (2)$$

At the Maximization-step, the model is updated using the computed probabilities from the Expectation-step:

$$\mu_i = \frac{\sum_{j=1}^{n} \lambda_{ik} z_k}{\lambda_{ik}} \qquad (3)$$



(a) Key point detection.

(b) Examples of crossings and T-Junctions.

(c) Examples of separating and connecting rulings.

(d) Connecting rulings.

Figure 2: Key point detection and validation using hand-drawn rulings.

$$\Sigma_i = \frac{\sum_{j=1}^{n} \lambda_{ik}(z_k - \mu_i)(z_k - \mu_i)^T}{\sum_{j=1}^{n} \lambda_{ik}} \quad (4)$$

To facilitate parameter estimation with limited number of samples (usually less than 100 in a page), we set the covariance matrix Σ_i to be $\gamma_i \mathbf{I}$, *i.e.*, an identity matrix with a scaling factor. In addition, the K-Means algorithm is employed to initialize the model. After the EM estimation, we set $\mathcal{W} = \mu_1 + 2\sqrt{\gamma_1}$, assuming $\mathcal{N}_1$ represents the inter-word spacings in plain text lines. Figure 3a shows some detected key points.

3.1 Hand-drawn Ruling Impact

After detecting all key points, we make use of hand-drawn rulings to confirm or exclude them. For vertical ones, they usually indicate text separation. For horizontal ones, however, they may indicate concatenation of adjacent text lines, or separation of table rows from plain text lines. Thus, we classify horizontal rulings into separating ones and connecting ones, based on the *crossings* and *T-Junctions*, as shown in Figure 2b:

i) Separating rulings always have T-Junctions only while connecting rulings should have at least one crossing, as shown in Figure 2c.

ii) If there are multiple hand-drawn ruling segments approximately on the same horizon, then such rulings are classified as connecting rulings, as shown in Figure 2d.

In addition, we also observe that a separating hand-drawn ruling followed by a connecting one usually indicates the region for table column headers. For table headers, it is possible for people to write the headers in multiple text lines due to space constraints (Figure 2c). In such cases, it is better not to segment table headers by their physical appearance, but to preserve their logical meaning. Therefore, we group text lines in such regions during table row detection.

Moreover, if any vertical hand-drawn rulings intersect handwritten words, we segment them by building a local horizontal projection profile (HPP) and finding a max-margin cut between two neighboring words. If two words are horizontally isolated such that the HPP can not generate a reasonable cut, the average x-value of two end points in the vertical ruling is used.

Assuming a hand-drawn ruling traverses through a key point region, it has several impact:

i) Vertical hand-drawn rulings can validate key points with widths smaller than $\mathcal{W}$.

ii) Connecting rulings can validate key points with heights larger than the average spacing between text lines $\mathcal{L}$.

iii) Separating rulings can in-validate key points with reasonable heights, e.g., approximately $\mathcal{L}$.

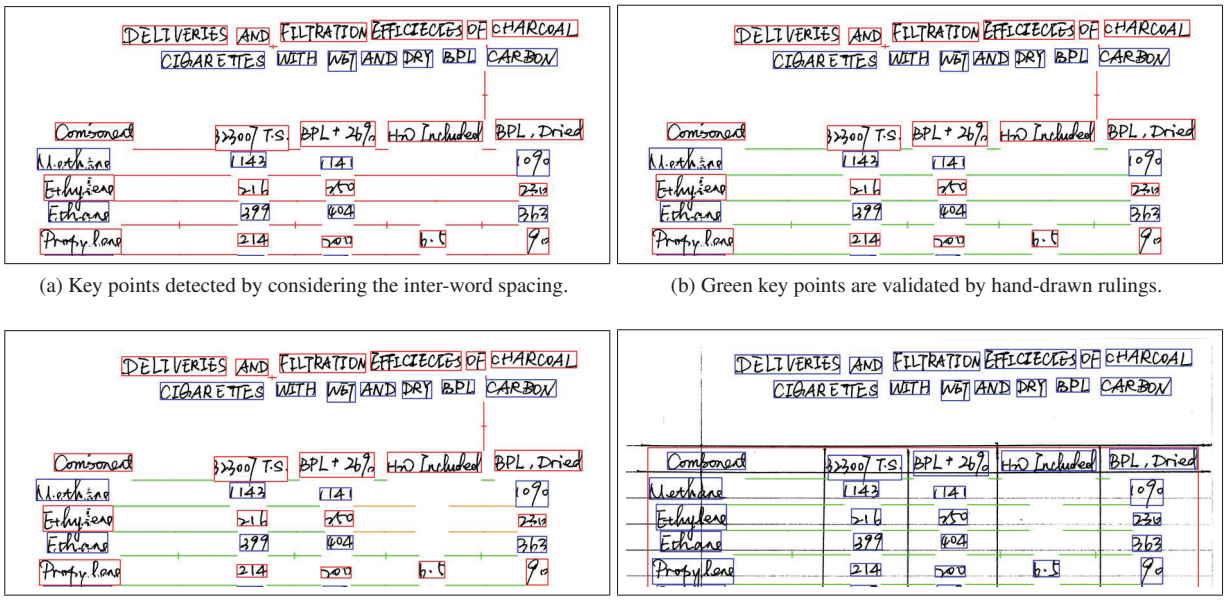
Green cross signs in Figure 3b are validated key points.

3.2 Key Point Grid Generation

This step is necessary when several tables cells are missing such that associated key points are missing or wrong. First, we cluster key points into groups using the Basic Sequential Algorithmic Scheme (BSAS) clustering [19]. Next within each cluster, we build a HPP of key points to find salient columns of key points. Then we traverse each row of key points to see if any key points are missing at expected positions. If so, we add imaginary key points correspondingly. In cases where a key point spans over two columns of key points (Figure 3b), we split it into two. See Figure 3c for the 2-D grid of key points where orange cross signs are imaginary key points (KIs), and green ones are validated key points (KRs). After generating grids of key points, we validate them by the graph model, as discussed in the following section.

4. Graph Model for Tables

Conditional Random Fileds (CRFs) are discriminative graph models for labeling tasks such as text



(a) Key points detected by considering the inter-word spacing.

(b) Green key points are validated by hand-drawn rulings.

(c) Merged key points are split to match the grid pattern.

(d) Key points detected by the graph cut algorithm.

Figure 3: Snapshots of key points in different stages.

stream identification and document image segmentation [13, 17, 14]. Denote $X = \{x_i\}$ to be the observed features from each node in the graph, and $Y = \{y_i\}$ to be random variables over corresponding labels, *i.e.*, 0 means false-alarming key points and 1 means valid ones. Then the joint distribution over the labels y_i given an observation x_i has the following form:

$$p(y_i|x_i) \propto \exp(pA(y_i, X) + q \sum_{\{(i),(j)\}\in E} I(y_i, y_j, X)) \tag{5}$$

where $A(\cdot)$ is called the *association* potential measuring data penalty scores, and $I(\cdot)$ is the *interaction* potential measuring influences of neighboring nodes. Figure 4a shows the CRF topology in our experiments. It turns out that the maximization problem of $p(y_i|x_i)$ can be solved by the Min-Cut/Max-Flow algorithm [2], where valid key points are associated to the *source* node and the rest to the *sink* node, as shown in Figure 4b.

In our work, we define $A(y_i, X)$ to be:

$$A(y_i, X) = \sum_{j\in CN} \frac{ColFeat(j)}{\exp(-R(CN))} + \sum_{j\in RN} \frac{RowFeat(j)}{\exp(-1)} \tag{6}$$

where CN means all the nodes in the same column and $R(\cdot)$ counts the number of KRs in the column. RN means the immediate left and right neighboring nodes.

In our experiments, both $ColFeat(\cdot)$ and $RowFeat(\cdot)$ generate single-value features. Denote w as the width between two vertically adjacent key points on the same column, and $\overline{w}$ the average width in the column. Then $ColFeat(\cdot)$ is computed as:

$$ColFeat(KR) = \begin{cases} 0 & \text{if } w < \overline{w} \\ 1 & \text{otherwise} \end{cases}$$

$$ColFeat(KI) = \begin{cases} -1 & \text{if } w < \overline{w} \\ 0 & \text{otherwise} \end{cases}$$

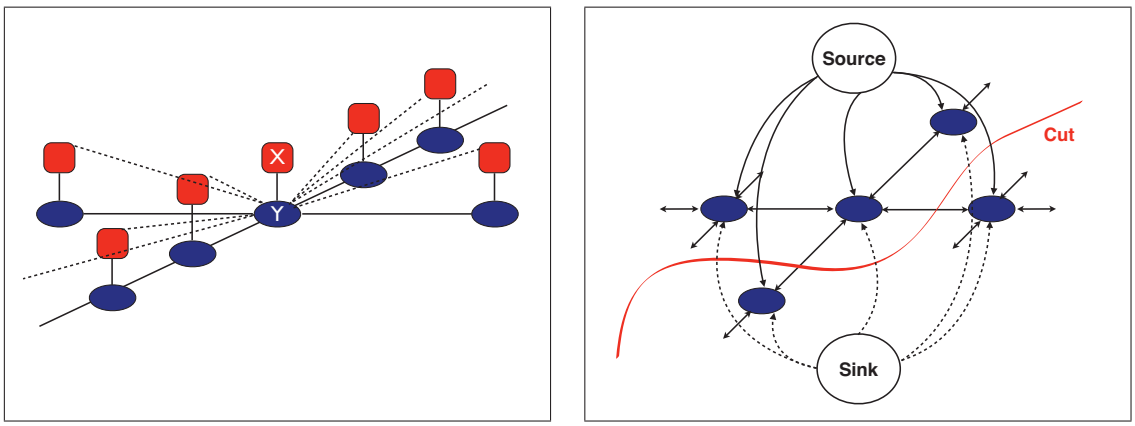
$RowFeat(\cdot)$ is computed by finding handwritten words between two horizontally adjacent key points on the same row:

$$RowFeat(\cdot) = \begin{cases} 1 & \text{if any HW words exist} \\ 0 & \text{otherwise} \end{cases}$$

After computing association potentials for each node in the graph, we define the *source capacity* of a node to be $A(y_i, X)$ and the *sink capacity* to be $1 - A(y_i, X)$. If the source capacity is positive, the initial label for the node is 1, otherwise it is 0. Based on the initial labels, we can compute the interaction potential as follows:

$$I(y_i, y_j, X) = exp(y_i \times y_j) \tag{7}$$

The parameters p and q adjust the weights of association potential and interaction potential. Ideally, these parameters should be learned from a training dataset. Due to the small-scale dataset we have for now (which is actively growing), however, we are unable to train the model extensively. Thus, in current experiments, we manually set the weights to be equal.



(a) The topology of CFRs used in experiments. Note all the nodes within the same column have impacts on the central node, while only the immediate left and the right neighboring nodes have.

(b) The graph model in Min-Cut/Max-Flow algorithms. Each node represents a hidden node in the CFRs. The arrows represent the flow directions of each node.

Figure 4: Table models used in our experiments.

An example result is shown in Figure 3d. With all the key points computed by the CRF model, the table region is defined by the bounding box that contains all the associated table rows. Having grouped key points into columns, we group text to table columns so that we get a 2-D grid of table cells.

5. Experiments

5.1 Data Preparation

We collected machine-printed documents that contain tables from the Tobacco800 dataset [20] and asked students to copy from these documents to paper. Students chose where to break lines, the spacing between words and table columns, paper sheets, and writing instruments. To mimic the characteristics of handwritten tables, we added/removed rulings in tables and left certain cells empty. As an ongoing event, so far we have collected 20 handwritten pages from two students, and we aim for a dataset with 200 pages or more.

Each collected handwritten document was scanned at 600 DPI into PDF files, using an HP copier with the bitonal setting under the plain text mode. The dimension of extracted TIFF images is $5100w \times 6600h$.

5.2 Experimental Results

Figure 3 shows some results during the tabular structure detection and recognition. Currently, the performance evaluation is based on manual investigation. We have labeled 584 table cells (369 key points) from 20 pages with 22 handwritten tables. If a cell is properly defined by its neighboring key points, we consider it a valid table cell. Then, we compute the *cell precision* and *cell recall*, as well as the *key point precision* and *key point recall*. For both cell precision and key point precision, we obtained a 100% accuracy. While for cell recall we obtained 93.3% and key point recall 93.0%. In terms of running time, our algorithm completed an image in less than 10 seconds on a Quad-core Intel 3.0 GHz machine, and we have not make an effort to optimize the execution yet.

More errors occurred when no hand-drawn rulings presented and some table rows are vertically isolated from the others. In addition, when the spacings between table columns varied too much, small spacings were mistakenly classified as inter-word spacings in plain text lines such that several key points were missing.

Although the precision/recall measures are intuitive and commonly used in literature, we consider them still difficult to show "how well" the system performs. Eventually, we want to measure how well the algorithmic results are consistent with a human user's perception. This is left for future work.

6. Conclusions

In this paper, we introduced a graph model based approach of detecting and decomposing tabular structures in noisy handwritten documents. First, we detected key points between two text lines and made use of hand-drawn rulings to validate/exclude them. Then, we built a 2-D grid of key points to model tabular structures and to compute structural features in the graph. Finally, the Min-Cut/Max-Flow algorithm validated key points in the grid globally. Experiments on 22 tables which contain 584 table cells showed a cell precision of 100% and

a cell recall of 93.3%.

As for the future work, more documents will be collected, annotated, and tested. The goal is to obtain a dataset with 200 samples or more. In addition, we plan to design more discriminative structural features for training and testing in the graph model.

Acknowledgment

This work is supported by a DARPA IPTO grant administered by Raytheon BBN Technologies.

References

[1] J. Bilmes. A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian Mixture and Hidden Markov Models. Technical report, Internation Computer Science Institue and Computer Science Division, University of California at Berkeley, 1998.

[2] Y. Boykov and V. Kolmogorov. An experimental comparison of Min-Cut/Max-Flow algorithms for engergy minimization in vision. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 26(9):1124–1136, 2004.

[3] F. Cesarini, S. Marinari, L. Sarti, and G. Soda. Trainable table location in document images. In *International Conference on Pattern Recognition*, pages 236–240, 2002.

[4] J. Chen and D. Lopresti. A model-based ruling line detection algorithm for noisy handwritten documents. In *Proceedings of the 11th International Conference on Document Analysis and Recognition*, pages 404–408, September 2011.

[5] J. Chen and D. Lopresti. Table detection in noisy offline handwritten documents. In *Proceedings of the 2011 11th International Conference on Document Analysis and Recognition*, pages 399–403, September 2011. 399-403.

[6] D. Doermann, E. Zotkina, and H. Li. GEDI-a groundtruthing environment for document images. In *The ninth International Workshop on Document Analysis Systems*, pages 519–522, 2010.

[7] B. Gatos, D. Danatsas, I. Pratikakis, and S. Perantonis. Automatic table detection in document images. In *Proceedings of the Third International Conference on Advances in Pattern Recognition*, pages 609–618, 2005.

[8] J. Handley. *Electronic imaging technology, Chapter 8 (Document Recognition)*. IS&T/SPIE Optical Engineering Press, 1999.

[9] R. Haralick. Document image understanding: geometric and logical layout. In *Proceedings of the 1994 Computer Vision and Pattern Recognition*, pages 385–390, 1994.

[10] Y. Hirayama. A method for table structure analysis using DP matching. In *International Conference on Document Analysis and Recognition*, pages 583–586, 1995.

[11] J. Hu, R. Kashi, D. Lopresti, and G. Wilfong. Medium-independent table detection. In *Document Recognition and Retrieval VIII (IS&T/SPIE Electronic Imaging*, pages 44–55, 2001.

[12] A. Laurentini and P. Viada. Identifying and understanding tabular material in compound documents. In *International Conference on Pattern Recognition*, pages 405–409, 1992.

[13] S. Nicolas, J. Dardenne, T. Paquet, and L. Heutte. Document image segmentation using a 2D Conditional Random Field model. In *Proceedings of the 2007 International Conference on Document Analysis and Recognition*, pages 407–411, 2007.

[14] X. Peng, H. Cao, R. Prasad, and P. Natarajan. Text extraction from video using Conditional Random Fields. In *Proceedings of the 2011 11th International Conference on Document Analysis and Recognition*, pages 1029–1033, 2011.

[15] J. Richarz, S. Vajda, and G. Fink. Towards semi-supervised transcription of handwritten historical weather reports. In *Proceedings of the 10th International Workshop on Document Analysis Systems*. To appear, 2012.

[16] F. Shafait and R. Smith. Table detection in heterogeneous documents. In *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, pages 65–72, 2010.

[17] S. Shetty, H. Srinivasan, S. Srihari, and M. Beal. Segmentation and labeling of documents using Conditional Random Fields. In *Proceedings of the 2007 Document Recognition and Retrieval IV, Proceedings of SPIE*, pages 6500U–1–11, 2007.

[18] Z. Shi, S. Setlur, and V. Govindaraju. A steerable directional local profile technique for extraction of handwritten Arabic text lines. In *Proceedings of the 2009 International Conference on Document Analysis and Recognition*, pages 176–180, 2009.

[19] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic Press, 2009.

[20] University of Maryland, Laboratory for Language and Media Processing (LAMP). Tobacco-800 signatures and logos dataset. http://lamp.cfar.umd.edu.

[21] X. Wang. *Tabular abstraction, editing, and formatting*. PhD thesis, University of Waterloo, 1996.

[22] R. Zanibbi, D. Blostein, and J. Cordy. A survey of table recognition: models, observations, transformations, and inferences. *International Journal on Document Analysis and Recognition*, 7(1):1–16, 2003.