2011 International Conference on Document Analysis and Recognition

Table Detection in Noisy Off-line Handwritten Documents

Jin Chen
Dept. of Computer Science & Engineering
Lehigh University
Bethlehem, PA 18015, USA
jic207@cse.lehigh.edu

Daniel Lopresti
Dept. of Computer Science & Engineering
Lehigh University
Bethlehem, PA 18015, USA
lopresti@cse.lehigh.edu

***Abstract*—Table detection can be a valuable step in the analysis of unstructured documents. Although much work has been conducted in the domain of machine-print including books, scientific papers, etc., little has been done to address the case of handwritten inputs. In this paper, we study table detection in scanned handwritten documents subject to challenging artifacts and noise. First, we separate text components (machine-print, handwriting) from the rest of the page using an SVM classifier. We then employ a correlation-based approach to measure the coherence between adjacent text lines which may be part of the same table, solving the resulting page decomposition problem using dynamic programming. A report of preliminary results from ongoing experiments concludes the paper.**

***Keywords*-Off-line handwriting; table detection; noisy documents;**

I. Introduction

In handwritten document analysis, detecting the presence of tables is useful for several reasons: (1) tables and table text can interfere with downstream attempts to perform handwriting recognition; (2) tables are worthy of attention in themselves for the information they contain which can, for example, form the basis for "question-answering" systems. Real handwritten documents, however, often contain artifacts and noise which make the problem challenging. Figure 1 shows a noisy handwritten page that contains two tables written in Arabic. The page image is skewed as a result of scanning and contains large "clutter" around the border. Ruling lines are present which overlap the handwriting. In addition, the basic nature of handwriting means that table rows and columns may be difficult to segment visually. Handwritten tables may or may not make use of hand-drawn ruling lines to delimit table cells, and distinguishing these from pre-printed ruling lines is also likely to be hard.

Hu, *et al.*, summarized the problem of table understanding as consisting of two sub-problems: detection and recognition [1]. Tables can be expressed in a variety media [2], [3], *e.g.*, ASCII, HTML, PDF, however, here we only consider those which are written on paper. Most past work on table detection has been for machine-print, as discussed below.

Table recognition assumes identified table regions and the goal is to find the *physical structure* and the *logical structure* of the table model [4]. There has been much work dealing with table recognition [5], [6].

Table detection, on the other hand, focuses on finding table regions. Laurentini and Viada use horizontal and vertical rulings as initial evidence for tables in machine-printed documents. They then perform several tests to exclude non-tabular areas, such as drawings with horizontal or vertical lines [5]. Hu, *et al.*, introduce a table detection method that does not rely on ruling lines and also is medium independent [1]. They first detect *inside space* white-streams within text lines and then compute the correlation between lines. Next, they solve the problem of optimally decomposing a page using dynamic programming (DP), finding the best way to partition the page into a single large table or multiple smaller ones. A recent paper by Shafait and Smith extends detection to multi-column document pages [7]. Namboodiri addresses the problem of table detection and recognition for on-line handwriting [8]. Making use of spatial information, he examines the handwriting at the stroke level and then employs multiple methods to detect tables.

In this paper, we address table detection in noisy off-line handwritten documents. We first locate and remove clutter from around the border of the page. Next, following a bottom-up approach, we divide the page into small tiles so that we can use a two-class Support Vector Machine (SVM) to classify text vs. non-text tiles. For classification, "Gradient-Structural-Concavity" (GSC) features are used. We then adapt the optimization approach proposed by Hu, *et al.* to find the best decomposition of the input page into some number of tables. Since handwriting is "messier" than machine-print, we need to modify the computation to increase the weight of inside-space correlations.

The remainder of this paper is organized as follows: in Section II, we describe our pre-processing methods for identifying clutter and pre-printed ruling lines. Next, we introduce text/non-text classification using an SVM in Section III. We explain the correlation-based similarity measure and the dynamic programming algorithm in Section IV, followed by our experimental setup in Section V. We present preliminary results in Section VI and conclude in Section VII.

1520-5363/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDAR.2011.88

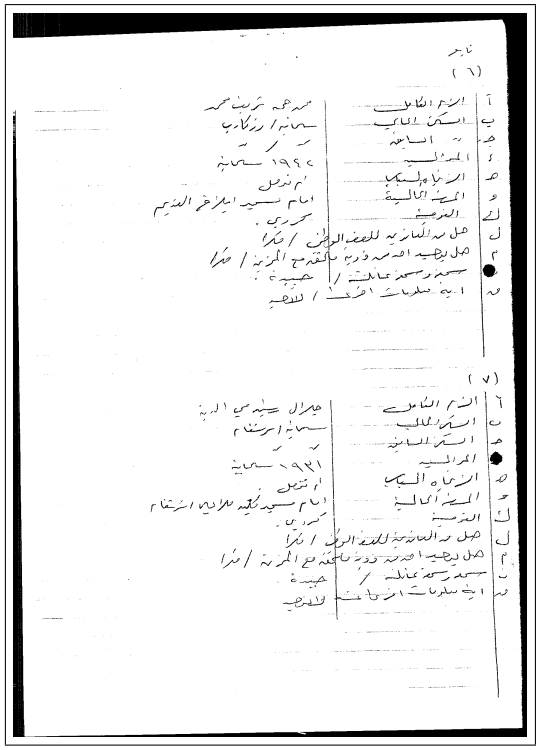


Figure 1: A noisy handwritten page containing tables.

II. Pre-processing

A. Clutter Removal

Background clutter is detected and removed using the approach of Agrawal and Doermann [9]. We first apply the Distance Tranform to the input page image. Since in the transformed image, clutter regions usually have large distance values, we binarize the page into *half-residual* cores using a threshold. Then from these cores we measure the increase in the number of pixels at each iteration while decreasing the threshold. A sudden rise in this number provides an estimate of the threshold that will best exclude clutter. Output from clutter removal is shown in Figure 2a.

B. Ruling Line Detection

Pre-printed ruling lines often overlap with handwriting, so it is important to be able to detect them. In our current work, we estimate the page skew by averaging the skews of horizontal lines, if they are present. The classical Hough Transform projects each point onto a set of sinusoidal curve points in the (ρ, θ) plane (the Hough Space). We employ an effective variant in our work [10]. In each iteration, we select a point randomly from the remaining point set, and then compute its sinusoidal curve in the Hough space and update the accumulation matrix. If the guard of the current maximum votes is larger than the threshold, then we search in each direction from the current position for the end points of the line segment. Since ruling lines might be degraded, short gaps (up to 15 pixels) are tolerated. Once the search stops, we record the coordinates of the line segment end points, exclude the corresponding points from the accumulation matrix, and proceed with the next iteration.

III. Text Classification

In noisy handwritten documents, touching characters are common and various artifiacts (*e.g.*, ruling lines) may interfere with the handwriting. Thus, we adopt a bottom-up approach to detecting text regions. First, we divide the page image into small equal-sized tiles. Then we extract features from the tiles for classification by the SVM.

A. Feature Extraction

Structural features such as Gradient-Structural-Concavity [11] are used to capture shape characteristics such as loops, branch-points, endpoints, and dots. These are multi-resolution features that combine three different shape attributes of text: gradients that representing the local orientations of strokes; structural information that extends the gradient to longer distances and provides information about stroke trajectories; and concavities that captures stroke relationships at longer distances.

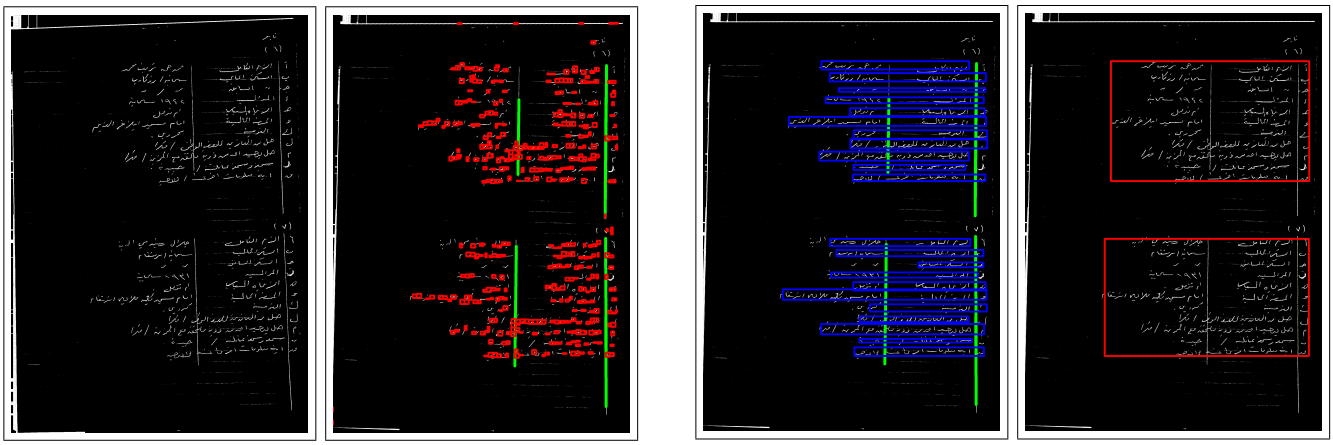
B. Text Classification

In our current work, we do not distinguish between handwriting and machine printed text, so the classification is a two-class problem between text and non-text tiles. We employ the libSVM toolkit [12] using the RBF kernel because it offers better discriminability than the linear kernel, while using less parameters than the polynomial kernel. For the penalty of mis-classification during training, we set the cost $c = 10000$. To facilitate SVM training and testing, we normalize feature vectors to the unit hyper-cube. Figure 2b shows an example of detected text tiles within a page.

IV. Table Detection

Having identified text tiles via the SVM classification, we use horizontal projection profiles (HPP's) to decide the most probable text lines for table rows. We first estimate the height, $\mathcal{H}$, of text lines by examining the sequence of peaks in the HPP's. We then use $\mathcal{H}$ to decide the boundaries for each text line. Finally, we exclude trivial candidate lines that contain fewer than five text tiles. The results of candidate table row detection are shown in Figure 2c.

Our table detection algorithm using candidate table rows is adapted from Hu, *et al.* [1]. For quick reference, we review that earlier work and identify the necessary changes for our application. At the highest level, the algorithm defines an optimal way of decomposing an entire page into



(a) After clutter removal. (b) Detected text tiles. (c) Segmented candidate table rows. (d) Detected tables.

Figure 2: Snapshots for the intermediate results of our table detection approach.

some number of tables given a similarity measure between individual rows. This decomposition is formulated as:

$$score[i,j] = max \begin{cases} tab[i,j] \\ max_{i \le k < j}\{score[i,k] + score[k+1,j]\} \end{cases} \quad (1)$$

where $1 \le i < j \le n$, and the boundary condition is:

$$score[i,i] = tab[i,i] \quad 1 \le i \le n \quad (2)$$

The value $tab[i,j]$ represents the score when interpreting rows i through j as a single table:

$$tab[i,j] = max \begin{cases} merit_{pre}(i,[i+1,j]) + tab[i+1,j] \\ tab[i,j-1] + merit_{app}([i,j-1],j) \end{cases} \quad (3)$$

where $1 \le i < j \le n$, and the boundary condition is:

$$tab[i,i] = 0 \quad 1 \le i \le n \quad (4)$$

In other words, $tab[i,j]$ is computed by either pre-pending the first row $Row[i]$ to the beginning of $tab[i-1,j]$, or appending the last row $Row[j]$ to the end of $tab[i,j-1]$.

Next, $merit_{app}(\cdot)$ and $merit_{pre}(\cdot)$ are defined as the summations of decaying correlation scores:

$$merit_{pre}(i,[i+1,j]) = \sum_{k=i+1}^{j} \frac{1}{e^{\gamma(k-i-1)}} \times lncorr(i,k) \quad (5)$$

and

$$merit_{app}([i,j-1],j) = \sum_{k=i}^{j-1} \frac{1}{e^{\gamma(j-1-k)}} \times lncorr(k,j) \quad (6)$$

where γ is a constant parameter that controls the exponential decay and is been set to 0.1 empirically.

At the lower level of the algorithm, the similarity between two candidate table rows is computed as the *inside space* correlation $lncorr(\cdot,\cdot)$. Here "inside space" means white-space that resides between two foreground components (text tiles in our discussion). Since we have detected table rows and the tiles they contain, we quantize each row using the tile size for the correlation computation, similar to the use of character width for machine-print documents.

The computation of the line correlation $lncorr(\cdot,\cdot)$ is defined differently in our work since we have found that the original correlation measure is not strong enough to cope with the large spatial variation in handwriting. Instead, we measure lengths and accumulate the score by multiplying by two. Hu, *et al.*'s method treats two tables as one when they are vertically separate and there are no text lines between them, as shown in Eq. 1. Therefore, when adjacent rows ($j = i+1$) are vertically separated by fours times the row height ($\mathcal{H}$) or more, they receive penalty correlation scores (-100 in our experiments).

In the end, we back-track through the $score[\cdot]$ matrix to recover the optimal decomposition. At the same time, we record associated k-values in Eq. 1 as the dividing rows between distinct tables.

V. Experimental Setup

A. Data Preparation

We have tested our method using a dataset provided by the Linguistic Data Consortium (LDC) [13]. This consists of 61 Arabic handwritten documents scanned at a resolution of 300 dpi and then binarized. The dataset is made up of real-world documents, so the handwriting is "messy" and unconstrained, in contrast to pages prepared specifically for research. The ground-truth for text regions (handwriting, machine-print) is available as bounding polygons. For table detection, we randomly selected 20 pages as the test set and the remaining 42 pages for training the SVM classifier.

B. Performance Measures

Various performance measures have been proposed for evaluating table detection algorithms. Simple measures include *precision* and *recall* [14]. More sophisticated ones include computing the similarity of two documents in terms of their table structures [1]. We use area-ratio-based measures proposed by Shafait and Smith [7], as explained below.

Shafait and Smith use bounding boxes to describe detected tables and the ground-truth. Denote G_i as the bounding box for the i-th ground-truthed table, and D_j as the one for the j-th detected table on a page. Then the overlap ratio between these two tables is defined as:

$$A(G_i, D_j) = \frac{2|G_i \cap D_j|}{|G_i| + |D_j|}, \qquad A \in [0, 1] \tag{7}$$

where $|G_i \cap D_j|$ is the joint area of two tables, and $|G_i|, |D_j|$ are the individual areas of two tables. They further categorize detection results as:

Correct Detection: $|A| \geq 0.9$ with a one-to-one correspondence between detected and ground-truth tables.

Partial Detection: $0.1 < A < 0.9$ with a one-to-one correspondence between detected and ground-truth tables.

Over-segmentation: multiple detected tables correspond to one ground-truth table.

Under-segmentation: multiple ground-truth tables correspond to one detected table.

Missed Table: ground-truth tables have marginal overlap with detected ones, $i.e., A \leq 0.1$.

False Positive Detection: detected tables have marginal overlap with ground-truth ones, $i.e., A \leq 0.1$.

Area Precision:

$$\frac{\text{Area of ground-truth regions in detected regions}}{\text{Area of all detected table regions}} \tag{8}$$

Area Recall:

$$\frac{\text{Area of ground-truth regions in detected regions}}{\text{Area of all ground-truth table regions}} \tag{9}$$

VI. Experimental Results

Good recall for text tiles in the SVM classification is critical for the table detection algorithm. We tried tile sizes of 20×20, 25×25, 30×30, and 35×35, and found that 25×25 gave the best performance: 94.63%. We correctly classified $35,291$ out of $37,293$ tiles for the 20 test documents.

We plot our table detection results in Figure 3. Among the 23 tables present in the 20 test pages, our algorithm detected nine correctly and six partially correctly. Although obtaining a high percentage for *correct detection* seems hard, we did achieve reasonable performance in terms of *area precision* (77.6%) and *area recall* (84.0%).

In terms of errors, we observed a relatively high percentage of *over-segmentation*: 26.1%. One reason for this might be failures in text tile classification. Another might be that in the test pages, several tables had cells that were left blank. Both of these issues will result in low correlation scores. An example of over-segmentation is shown in Figure 4a.

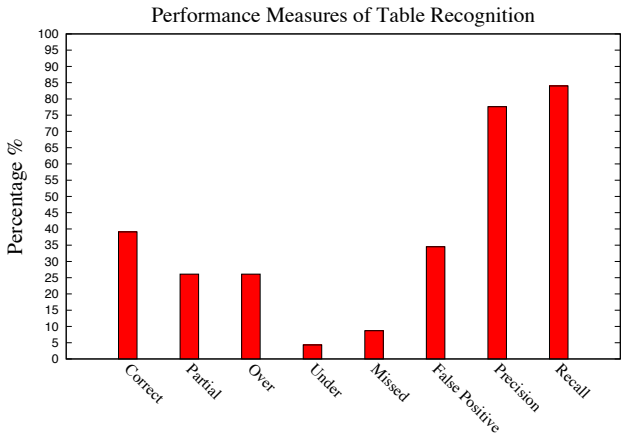


Figure 3: Experimental results based on a set of performance measures.

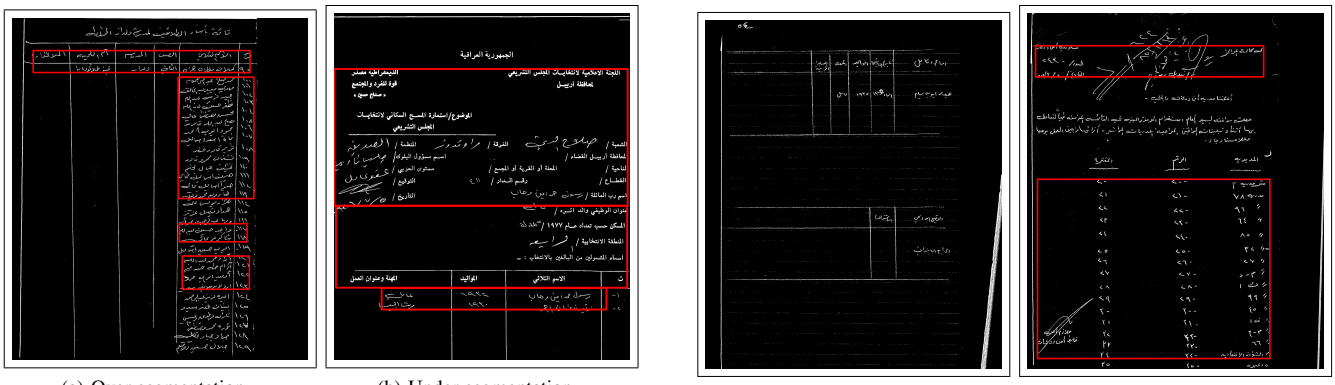
Figure 4b shows a case of under-segmentation. This page has a complicated layout: letter-like text, a table (form) that mixes machine-print and handwritten text, and a table with rulings. Since the detected table (the center red rectangle) overlaps with the form and the table at the bottom, we consider this to be an under-segmentation. Note that the letter-like region at the top of the page was mistakenly detected as a table. This is because the inside space between the first two lines is taken to resemble a table structure, which is a mistake.

In some cases, the decision as to when to separate two tables went wrong, as shown in Figure 4c. Since rows in these tables are far apart, they were mistakenly assigned negative correlation scores and thus were missed by the detection algorithm. One possible solution is to adapt the threshold to the spacing between rows.

Other document layouts, such as signature blocks, pose problems as well. Figure 4d shows such a situation. In this page, the signature at the top of the page and the text adjacent to it show strong correlation, so they are detected as part of a table. The same is true at the bottom of the page: the signature should not be included in the table region.

Although our test data is written in Arabic, the method we have described should be script-independent. GSC features are widely used for document analysis in a number of different languages. Testing on other datasets is ongoing.

We believe it should be possible to improve table detection accuracy in a number of ways. Instead of using equal-sized tiles, we might consider using variable-sized ones so that the stroke curvature of text is preserved. A more precise way of segmenting text tiles into candidate table rows would benefit the algorithm. Finally, artifacts and noise can corrupt the correlation computation. By first excluding signatures, logos,



(a) Over-segmentation. (b) Under-segmentation. (c) Missed tables. (d) False positive case at the top.

Figure 4: Snapshots for the different types of errors seen in our experiments: over-segmentation, under-segmentation, missed tables, and false positive tables. Note that Figure 4b contains multiple errors – see the text for explanation.

etc., we should be able to further improve performance.

VII. Conclusions and Future Work

Although much related work has been done in the domain of machine-print, table detection is likely to be more challenging for unconstrained off-line handwritten documents. In this paper, we have investigated this problem, basing our approach on the text/non-text classification of small image tiles, and then applying a bottom-up approach to group tiles into candidate table rows. We showed how to measure the correlation between potential table rows and presented a dynamic programming algorithm to solve the resulting optimization problem. Preliminary experimental results seem promising, but also suggest areas in which improvements must be made, several of which are now under investigation.

Acknowledgments

We thank Dr. Xujun Peng for providing an implementation of the GSC feature set. This work is supported by a DARPA IPTO grant administered by Raytheon BBN Technologies.

References

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

[2] R. Zanibbi, D. Blostein, and J. Cordy, "A survey of table recognition: models, observations, transformations, and inferences," *International Journal on Document Analysis and Recognition*, vol. 7, no. 1, pp. 1–16, 2003.

[3] D. Embley, M. Hurst, D. Lopresti, and G. Nagy, "Table-processing paradigms: a research survey," *International Journal on Document Analysis and Recognition*, vol. 8, no. 2, pp. 66–86, 2006.

[4] X. Wang, "Tabular abstraction, editing, and formatting," Ph.D. dissertation, University of Waterloo, 1996.

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

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

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

[8] A. Namboodiri, "On-line handwritten document understanding," Ph.D. dissertation, Michigan State University, 2004.

[9] M. Agrawal and D. Doermann, "Clutter Noise Removal in Binary Document Images," in *International Conference on Document Analysis and Recognition*, 2009, pp. 556–560.

[10] B. Lamiroy, D. Lopresti, H. Korth, and J. Heflin, "How carefully designed open resource sharing can help and expand document analysis research," in *Proc. Document Recognition and Retrieval XVIII (IS&T/SPIE International Symposium on Electronic Imaging)*, 2011.

[11] J. Favata and G. Srikantan, "A multiple feature/resolution approach to handprinted digit and character recognition," *International Journal of Image Systems and Technology*, vol. 7, no. 4, pp. 304–311, 1998.

[12] C.-C. Chang and C.-J. Lin, in *LIBSVM: a library for support vector machines*, 2001, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[13] "The linguistic data consortium," http://www.ldc.upenn.edu/.

[14] T. Kieninger and A. Dengel, "An approach towards benchmarking of table structure recognition results," in *International Conference on Document Analysis and Recognition*, 2005, pp. 1232–1236.