Voronoi++: A Dynamic Page Segmentation approach based on Voronoi and Docstrum features

Mudit Agrawal and David Doermann
Institute of Advanced Computer Studies
University of Maryland
College Park, MD, USA
{mudit, doermann}@umd.edu

Abstract

This paper presents a dynamic approach to document page segmentation. Current page segmentation algorithms lack the ability to dynamically adapt local variations in the size, orientation and distance of components within a page. Our approach builds upon one of the best algorithms, Kise et. al. work based on Area Voronoi Diagrams [10], which adapts globally to page content to determine algorithm parameters. In our approach, local thresholds are determined dynamically based on parabolic relations between components, and Docstrum based angular and neighborhood features are integrated to improve accuracy. Zone-based evaluation was performed on four sets of printed and handwritten documents in English and Arabic scripts and an increase of 33% in accuracy is reported. [1]

1. Introduction

With the advancements in information and communication technology, various forms of paper documents are being scanned in order to be interpreted and indexed. The bigger vision however, is to treat paper as a legitimate form of media (like magnetic tapes and optical discs) which can be both machine and human readable [15]. One challenge is that, the variety of paper documents being scanned today is much more diverse than what it was several years ago. Many new scripts, more complex, non-Manhattan, computerized page layouts and various font styles are making this vision challenging. Furthermore, a much larger percentage of handwritten material is being acquired which does not adhere to traditional layout constraints. Character recognition as well as various established pre-processing modules such as noise removal, layout analysis and zone classification are effected by this increase in complexity.

The process of identifying the layout structures of a document image can be physical (process of dividing the document into physical homogeneous zones) or logical (process of assigning logical roles and relations to detected zones). Page segmentation algorithms fall into the category of physical layout analysis. They perform segmentation of a document page into homogeneous zones, each consisting of only one physical layout structure [17]. Layout analysis can be pixel based or texture based segmentation [8], but we assume the final result is a region segmentation. In texture-based segmentation, isolated points or small areas could be classified as zonal objects disregarding the *connectivity* aspect of an object. In contrast, we are concerned with non overlapping geometric zones where document components are separated by white space. Such connected component based approaches use macro level content information, and can be further classified into Manhattan [13, 18, 3, 4] and non-Manhattan layouts [15, 10, 9].

1.1 Manhattan layout based

There are four representative algorithms for Manhattan layout based page segmentation. The run-length smearing algorithm (RLSA) designed by Wong et. al [18] is one of the oldest technique to segment the page into homogeneous regions. It smears components in each zone into each other using the perceived text direction and some thresholds, forming a distinct component per zone. X-Y Cut Page Segmentation [13] is a tree-based top-down algorithm, which starts with entire page as root. Based on horizontal and vertical projection profiles of foreground pixels, each node is split recursively till minimum formed regions are contained in its leaves. In 2002, Baird devised Whitespace Analysis method [3], another top-down algorithm, based on the analysis of the background structure in document images. It generates a sorted list of maximal elongated white-

[1] The partial support of this research by DARPA through BBN/DARPA Award HR001108C0004 and the US Government through NSF Award 1150713501 is gratefully acknowledged

978-0-7695-3725-2/09 $25.00 © 2009 IEEE
DOI 10.1109/ICDAR.2009.270

space rectangles. They are then unified one by one (till a stop rule is satisfied) to generate a sequence of enclosed segmentations. Constrained Text-line detection by Breuel [4] builds on white-space and X-Y cut, with an added consideration of column-separations, etc. (gutters) while grouping text-lines. This has a direct advantage over smearing technique where presence of a known gutter may skew the thresholds.

1.2 Non-Manhattan layout based

The focus of this paper is on more generic non-Manhattan layouts that are prominent in handwritten or annotated documents. Connected component analysis, skew [5, 15, 7] and analysis of background [16, 2, 14] have been used by researchers to perform page segmentation on non-Manhattan layouts. Kise et. al. [10] and Gorman [15] are the most widely cited algorithms to perform geometric page segmentation on non-Manhattan layouts of non overlapping zones. Evaluation experiments [17] have shown that the Voronoi based approach excels on a mixed dataset of both handwritten and machine printed documents for diverse scripts such as English and Arabic. Nevertheless, there are still challenges which the technique faces with such datasets. The goal of this paper is to highlight those shortcomings and describe our solutions. We propose a more deterministic approach, called Voronoi++. Instead of linear relations between distance and area ratio of connected components, we show that dynamically determining these relations parabolically and combining angular and neighborhood features from Docstrum based approach improves accuracy.

This paper is organized as follows. In Section 2 we give an overview of Voronoi and Docstrum algorithms. Section 3 lists the challenges the Voronoi based approach faces in dealing with a diverse set of data and how Voronoi++ overcomes them using new techniques and features. This is followed by evaluation in Section 4 and conclusion in Section 5.

2. Overview

2.1. Area Voronoi Based Segmentation

The central idea of the algorithm is creation of Voronoi edges between pairs of connected components using area based Voronoi tessellation [10]. Each edge bisects two points on contours of different components. A physical zone is a fusion of these Voronoi cells, formed by the elimination of Voronoi edges based on two features:

1. Minimum distance

$$d(E) = \min_{1 \le i \le m} d(p_i, q_i) \tag{1}$$

where p_i and q_i are pair of points on connected components (CCs) P and Q, constituting i^{th} edge between them

2. Area Ratio

$$a_r(E) = \frac{max\ of\ areas\ of\ 2\ CCs}{min\ of\ areas\ of\ 2\ CCs} \tag{2}$$

An edge is deleted if it satisfies the following two criteria:

$$\frac{d(E)}{T_{d1}} < 1 \quad or \tag{3}$$

$$\frac{d(E)}{T_{d2}} + \frac{a_r(E)}{T_a} < 1 \tag{4}$$

where $T_{d1} < T_{d2}$, T_{d1} relates to inter-character spacing, T_{d2} relates to inter-word/line spacing

2.2. Docstrum Segmentation

Docstrum performs transitive closure on within-line components to obtains lines and then on lines to form regions. The thresholds for transitivity are based on the properties of distance and angle of each connected component with its K nearest neighbors [15]. The advantage of Docstrum over the Voronoi based approach is its 'semi-local' behavior. Each component looks at K nearest neighbors to make a decision of its association, unlike Voronoi where decision is solely nearest neighbor based. In spite of this, Docstrum has been designed mainly for text-only documents.

3. Voronoi++

3.1. Dynamic Distance Threshold

In Voronoi based approach, global thresholds T_{d2} and T_a are determined statistically from the document. T_{d2} is based on second most frequent distance associated with the edges, which corresponds to the inter-word separation of prominent text-size in the document. T_{d2} and T_a have an inverse linear relation with each other, hence neighbors with higher a_r need lower $d(E)$ to form an edge between them (as shown in Figure 1a). Typically T_a is chosen such that it forces an edge between components of drastically different sizes ($a_r = T_a$) separated by an epsilon ($d(E) \rightarrow 0$) distance. Fixing such global thresholds gives rise to problems which can be categorized as follows:

Over segmenting larger fonts: Larger text has greater word-separation than the normal size text, T_{d2} is based on. a_r still being close to 1, larger font text lines often get segmented into zones containing individual words or characters. Our proposed solution is to increase the word-separation threshold locally by a factor of size of component to the most frequent component size (not average) on

the page if both components are greater than the frequent text size.

$$T_{d2} = T_{d2} * fa_r$$
$$fa_r = \frac{average\ area\ of\ components}{most\ frequent\ area} \quad (5)$$

Grouping dissimilar text sizes: Sectional headings, with $1 < a_r < 4$, tend to merge with the text if they are separated by distances similar to the inter-line distance of prominent text. Though a_r accounts for the difference in the areas of the two components, the relation between increase in a_r and decrease in $d(E)$ is still linear. Hence, an increase in a_r even by a factor of 4 (for example) does not decrease $d(E)$ requirement for edge-formation by appropriate amount. Decreasing T_{d2} by a factor of a_r, improves the chances of edge-formation between dissimilar size components as follows:

$$T_{d2} = T_{d2} - T_{d2} * a_r/T_a \quad (6)$$

Substituting this in equation 4,

$$\frac{d(E)}{T_{d2} - T_{d2} * a_r/T_a} + \frac{a_r(E)}{T_a} < 1$$

Solving, we get:

$$T_{d2}.a_r^2 - 2.T_{d2}.T_a.a_r - T_a^2.d(E) + T_{d2}.T_a^2 = 0 \quad (7)$$

This is an equation of a conic section of the form $Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0$. Since $B^2 - 4AC = 0$ and $A \neq 0$, the curve is indeed a parabola and demonstrates the parabolic relation. The new relation is depicted in the Figure 1b.

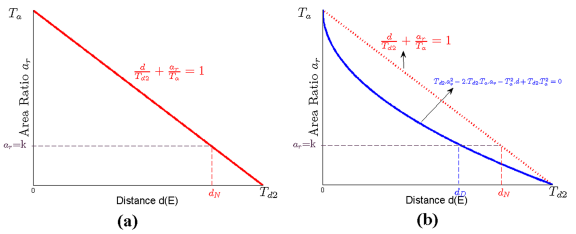


Figure 1: (a) Linear (b) Parabolic relation between inter-word distance and area ratio

3.2. Combining Docstrum and Voronoi features

Angles: Voronoi weighs two components solely on the basis of their distance. In degraded documents, missing components can increase inter-character or inter-word gaps, forcing edges at those places. This may create unpleasant results as lines are split at two or more places, leading to vertical splits in a zone. In Docstrum, each nearest-neighbor pair i,j is described by a 2-tuple $D_{ij}(d, \Phi)$ of distance d and angle ϕ between centroids of the two components. Due to concentration of mass at various locations, the centroids of components may differ in ordinate positions even for similar size characters. Hence, a character's bounding box centroid is used. Then the most frequent angle α between a neighborhood pair is calculated. The deviation of a neighborhood angle from α is calculated as a Gaussian function as follows:

$$dev = e^{-\frac{(x-\alpha)^2}{2.\sigma^2}}.K$$

where σ is the deviation allowed (chosen value 30 °), K is the factor limit with which T_{d2} can be increased. We chose K to be 0.5, allowing T_{d2} to be increased by a factor of 1.5

$$T_{d2} = T_{d2} + T_{d2}.dev \quad (8)$$

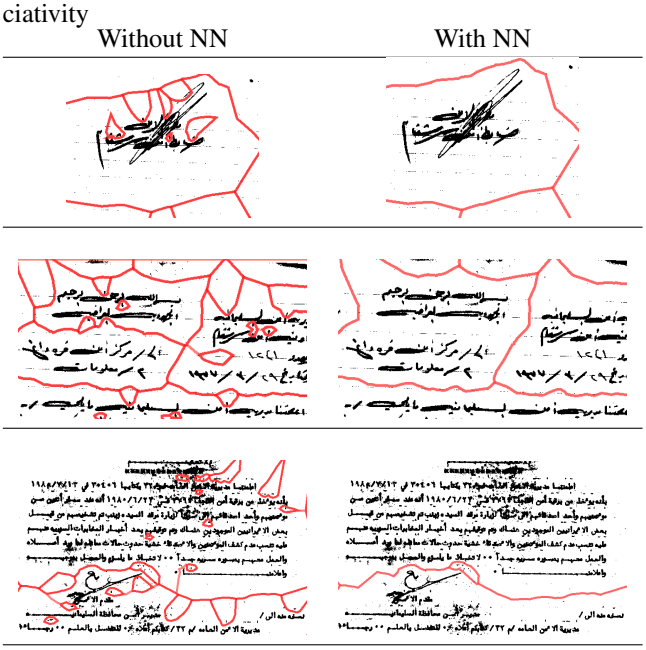
Nearest Neighbors: Arabic handwritten documents tend to have a lot of diacritics. The area ratio of a diacritic with its corresponding consonant/vowel is often large and the Voronoi based approach forces edges around such diacritic. Increasing the noise threshold, causes diacritics to be thresholded as noise and hence do not participate in edge formation. This strategy does not work for the stated problem for two reasons. First, associating diacritics, in many languages like French and Arabic, with an arbitrary region introduces significant problems. Second, choosing a higher noise threshold to eliminate such components as noise, leads to the removal of small characters (falling below the threshold) in edge formation, hence creating a virtual 'gap' within single (or multiple) words. This can create edges there, forcing over-segmentation.

In our solution, we note the association of any component, smaller than the most frequent size of component, to its nearest neighbor removes the possibility of such an edge. This is done by associating a nearest neighbor with each component. Results are shown in Table 1.

3.3. Word Separation Threshold using Valley Finding

The appropriate values of T_{d1} and T_{d2} are determined based on global layout and resolution. The frequency distribution of distances associated with edges is utilized to determine the two peaks corresponding to T_{d1} and T_{d2} respectively, as shown in Figure 2. For a raw frequency distribution $f_r(d)$, the smoothed distribution $f(d)$ is calculated using a window size ω, to get rid of local peaks [10]. However, it is not possible to choose ω correctly without knowing the real distance between the global peaks, which in turn depends on the correct value of ω. Hence, local extremas still remain in this process and subsequently affect global peak determination for T_{d1} and T_{d2}. The Voronoi

Table 1: Comparison of results with and without NN associativity



based approach chooses the highest frequency distance as T_{d1} (character separation) and the second highest peak (after smoothing) as T_{d2}. No 'valley' determination is performed before determining the next most frequent distance. This leads to local extremas being picked, as shown in Figure 2. The result is over segmentation of the document.

The histogram is bimodal, having modes or peaks at character and word/line separations. We treat this histogram as a valley, having foothills between two mountains. Hence, selecting a threshold depends on how much effort it takes to get to one boundary or the other. At each point in the histogram, we associate the cost of leaving the valley. This is the minimum cost to reach either of the boundaries. Costs are incremented to the far bins only when the histogram is increasing on the right and decreasing on the left. The bin with maximum cost is the point in the valley [1]. This can be expressed as:

$$
\begin{aligned}
Cost_L(d) &= \sum_{k \le d, h(k-1) > h(k)} [h(k-1) - h(k)] \\
Cost_R(d) &= \sum_{k \ge d, h(k) < h(k+1)} [h(k+1) - h(k)] \\
Cost(d) &= \min(Cost_L(d), Cost_R(d)) \\
V_d &= \max_d(Cost(d)) \qquad (9)
\end{aligned}
$$

Largest peak on the right of V_d corresponds to T_{d2}. This avoids local peaks confusion and gives the best estimate of inter-word separation.

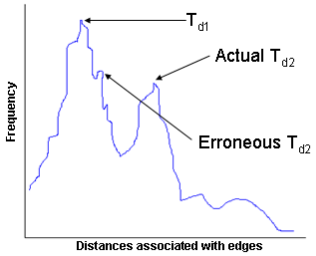


Figure 2: Frequency histogram of distances associated with edges

3.4. Polygonal Zones

Voronoi based page segmentation algorithm returns voronoi edges (line segments) separating zones. These edges can be traversed to form polygons, which can later be approximated to form minimum edge polygons. This avoids zone overlaps in case of rectangular representation of polygonal regions. Voronoi++ builds polygonal regions out of edges produced by voronoi segmentation in the following way. Each edge is represented by a pair of points $(x_1, y_1), (x_2, y_2)$. A region is formed by iteratively traversing adjacent edges. By associating each point with the edge id and then sorting them on x followed by y, reduces the problem to $O(nlogn)$ from $O(n^2)$ in the search space for the consecutive edge. Every edge can be a part of exactly two regions whereas a vertex is shared by more than two edges (or regions). During edge traversal, the rightmost or leftmost successive edge at a vertex is chosen based on the vertex's previous traversals. Once decided, a region its fully traversed through all its edges using the same turn. Extra edges are added on the boundaries, as voronoi segmentation does not produce those.

4. Evaluation

[17, 12] measure overall error as the percentage of ground-truth text-lines correctly contained within zones. A ground-truth text-line is said to lie completely within one detected zone if the area overlap between the two is significant. The drawback of that approach, however, is that if the segmentation algorithm outputs the whole page as one segment, the text-line split and missed errors go away and accuracy is higher. In order to avoid this, we compare ground-truth zones with result zones. A result zone is said to be detected, if its foreground pixels overlap with those of ground-truth above a user specified percentage. This is a much stricter evaluation scheme in terms of zone detection. We evaluated the Voronoi++ approach on datasets of printed and handwritten documents in English and Arabic scripts. The polygonal regions are outputted in xml format

designed for LAMP's GEDI tool [11]. This tool is used to label the data and visualize the segmentation and evaluation results. We compared our approach on 200 document images, half of which were randomly picked from University of Washington III(UW-III) database [6]. Figure 3 compares the number of zones returned by Voronoi and Voronoi++ based approaches. The increase in accuracy by 33% (Table 2) also proves that the decrease in number of zones does not increase merge errors.

Table 2: Number of zones and accuracy comparison of Voronoi and Voronoi++

	Voronoi	Voronoi++
Number of Segments	526	362
Precision	35.36%	68.78%
Recall	58.49%	78.30%

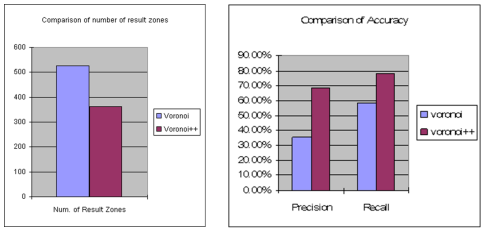


Figure 3: (a) Number of segments (zones) returned (b) Recall and Precision

5. Conclusion and Future Work

We have presented a dynamic approach to document page segmentation based on Area Voronoi Diagrams. Over segmentation of larger fonts and distinction based on dissimilar fonts is achieved by dynamically varying thresholds based on local parameters. The paper also proposes a combination scheme of Docstrum and Voronoi based approaches. Voronoi associates only distance metric to each Voronoi edge. Adding angular and neighborhood features from Docstrum approach improved segmentation, especially for degraded and Arabic documents. A more deterministic valley-finding algorithm finds word-separation threshold more accurately, without being stuck at local extremas. We compared the Voronoi and Voronoi++ approaches using polygon-based zonal comparison evaluation scheme and showed that Voronoi++ improves the accuracy by 33%. In our future work, we would like to enhance the approach by training the algorithm on its features using optimization approaches. An edge with associated distance below the threshold, when surrounded by edges with distances above threshold, is called a *single weak-edge*. Sometimes regions are 'opened' by few of such intermittent weak-edges. Associating neighbor edge distances with each edge, gives semi-local behavior to the algorithm. Such semi-local features will be our focus for the future.

References

[1] R. L. Allen and D. Mills. *Signal Analysis: Time, Frequency, Scale, and Structure*. Wiley-IEEE Press, 2004.

[2] A. Antonacopoulos and R. Ritchings. Flexible page segmentation using the background. *Proc. 12th Int'l Conf. on Patt. Reco.*, 2:339–344 vol.2, Oct 1994.

[3] H. S. Baird. Background structure in document images. In *Advances in Structural and Syntactic Pattern Recognition*, pages 17–34. World Scientific, 1994.

[4] T. M. Breuel. Two geometric algorithms for layout analysis. In *Workshop on Document Analysis Systems*, pages 188–199. Springer-Verlag, 2002.

[5] L. A. Fletcher and R. Kasturi. A robust algorithm for text string separation from mixed text/graphics images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10(6):910–918, 1988.

[6] I. Guyon, R. M. Haralick, J. J. Hull, and I. T. Phillips. Data sets for ocr and document image understanding research. In *Proc. of SPIE - Document Recognition IV*, pages 779–799. World Scientific, 1997.

[7] F. Hones and J. Lichter. Layout extraction of mixed-mode documents. *MVA*, 7(4):237–246, 1994.

[8] A. Jain and Y. Zhong. Page segmentation using texture analysis. volume 29, pages 743–770, May 1996.

[9] A. K. Jain and S. Bhattacharjee. Text segmentation using gabor filters for automatic document processing. *Mach. Vision Appl.*, 5(3):169–184, 1992.

[10] K. Kise, A. Sato, and M. Iwata. Segmentation of page images using the area voronoi diagram. *Comput. Vis. Image Underst.*, 70(3):370–382, 1998.

[11] S. J. M. Roth and D. Doermann. Gedi: Ground truth. editor and document interface. In *Summit on Arabic and Chinese Handwriting Recognition*, 2006.

[12] S. Mao and T. Kanungo. Automatic training of page segmentation algorithms: An optimization approach. In *Proc. of Int'l Conf. on Patt. Reco.*, pages 531–534, 2000.

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

[14] N. Normand and C. Viard-Gaudin. A background based adaptive page segmentation algorithm. In *Proc. Third Int'l Conf. on Doc. Analysis and Recog.*, page 138, Washington, DC, USA, 1995. IEEE Computer Society.

[15] L. O'Gorman. The document spectrum for page layout analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(11):1162–1173, 1993.

[16] T. Pavlidis and J. Zhou. Page segmentation and classification. *CVGIP: Graph. Models Image Process.*, 54(6):484–496, 1992.

[17] F. Shafait, D. Keysers, and T. M. Breuel. Performance comparison of six algorithms for page segmentation. In *7th IAPR Workshop on Document Analysis Systems*, pages 368–379. Springer, 2006.

[18] K. Y. Wong, R. G. Casey, and F. M. Wahl. Document Analysis System. *j-IBM-JRD*, 26(6):647–656, Nov. 1982.