Efficient Word Segmentation and Baseline Localization in Handwritten Documents Using Isothetic Covers

Mousumi Dutt, Bengal Engineering and Science University, India
Aisharjya Sarkar, Bengal Engineering and Science University, India
Arindam Biswas, Bengal Engineering and Science University, India
Partha Bhowmick, Indian Institute of Technology, Kharagpur, India
Bhargab B. Bhattacharya, Indian Statistical Institute, India

ABSTRACT

Analysis of handwritten documents is a challenging task in the modern era of document digitization. It requires efficient preprocessing which includes word segmentation and baseline detection. This paper proposes a novel approach toward word segmentation and baseline detection in a handwritten document. It is based on certain structural properties of isothetic covers tightly enclosing the words in a handwritten document. For an appropriate grid size, the isothetic covers successfully segregate the words so that each cover corresponds to a particular word. The grid size is selected by an adaptive technique that classifies the inter-cover distances into two classes in an unsupervised manner. Finally, by using a geometric heuristic with the horizontal chords of these covers, the corresponding baselines are extracted. Owing to its traversal strategy along the word boundaries in a combinatorial manner and usage of limited operations strictly in the integer domain, the method is found to be quite fast, efficient, and robust, as demonstrated by experimental results with datasets of both Bengali and English handwritings.

Keywords: Baseline Detection, Handwritten Document Processing, Isothetic Cover, Isothetic Polygon, Word Segmentation

INTRODUCTION

A handwriting portrays the characteristics of an individual, and hence has been studied in numerous disciplines including experimental psychology, neuroscience, engineering, anthropology, forensic science, etc. (Plamondon, 1993; Plamondon & Leedham, 1990; Simner et al., 1994, 1996; Galen & Morasso, 1998; Galen & Stelmach, 1993; Wan et al., 1991).

DOI: 10.4018/jdls.2011070101

Copyright © 2011, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
The analysis of handwritings has been quite important in recent times with the advancements of document digitization (Hole & Ragh, 2011; Saba, 2011; Terrades, 2010; Zhu et al., 2009), biometric authentication (Henniger & Franke, 2004; Hoque et al., 2008; Low et al., 2009; Makrushin, 2011; Schimke et al., 2005; Vielhauer, 2006; Vielhauer & Scheidat, 2005), forensic science (Franke & Köppen, 2001; Mâadeed et al., 2008; Mahmoudi et al., 2009; Pervouchine et al., 2008), etc. The result of analysis strives to interpret, verify, and recognize a particular handwritten document. The most difficult problem in the area of handwriting recognition is segmentation of cursive handwriting. The infinitude of different types of human handwritings amidst the similarities in the shapes of different characters renders the problem even more difficult. Hence, over the last few years, various works have been presented for specific domains, e.g., Bengali character recognition (Majumdar & Chaudhuri, 2007; Parui et al., 2008), text line identification (Chaudhuri & Bera, 2009), numeral recognition (Bhattacharya & Chaudhuri, 2009), check sorting (Gorski et al., 1999), address reading (Srihari & Keubert, 1997), tax reading (Srihari et al., 1996), office automation (Gopisetty, 1996), automated postal system (Vajda et al., 2009), etc.

Handwriting recognition techniques are based on either holistic or analytic strategies. In the holistic method, a top-down approach is employed where the whole word is recognized by comparing its global features against a limited size lexicon (Guillevic & Suen, 1998). On the other hand, analytic strategies adopt the bottom-up approach starting with characters, strokes, etc., eventually producing the meaningful text (Wang & Jean, 1994; Mohamed & Gader, 1996; Mao et al., 1998; Kim & Govindaraju, 1997). Clearly, in connection with handwriting recognition, it is important to extract/segment the words in a cursive writing such that the task of segregating the individual characters and strokes may be taken up. The segmented regions may be found out from the peaks of the projection profile of a gray-level image (Lee et al., 1996). Proper baseline extraction is important to segment out words correctly. To avoid the problems arising out of ascending and descending portions of the characters, information about the upper and the lower baselines are necessary. In general, the baselines are extracted from the projection profile (Guillevic & Suen, 1998).

As per the existence practice, the text line segmentation is usually done by considering a subset of the connected components in a document image (Louloudis et al., 2009). Word segmentation is achieved using the distinction of inter- and intra-word gaps using a combination of two different distance metrics.

To overcome the disadvantage of different distance measures in word segmentation, a gap metric based on the average distance is used for word segmentation (Huang & Srihari, 2008). A number of works have been reported in the literature for line and word segmentation in other languages, e.g., Chinese, Arabic, etc. Chinese word segmentation has various applications on Chinese text processing (Haizhou & Baosheng, 1998). The algorithm for detection of straight or curved baselines for Arabic handwritten text can be applied on online handwriting or off-line handwritten writing (Boubaker et al., 2009). A method for precise identification of ascending or descending parts of the words has been proposed using lexicon based search in Aida-zade & Hasanov (2009).

In our work, we have devised a novel method to segment out the words in a cursive handwriting by using the outer isothetic covers of the words constituting the handwritten document (Figure 1). The method has also the advantage of extracting the upper and the lower baselines by analyzing these covers corresponding to the handwritten words. Owing to the combinatorial nature of its traversal strategy while deriving the minimum-area isothetic covers corresponding to the words, and hence describing the covers as sequences of theirs vertices (or/and edges), it is endowed with an easy solution to locate the baselines of different words — whatsoever be their patterns and letter-wise constitution — as shown in this paper. A fast decision policy based on the covers, coupled with usage of operations strictly
in the integer domain, make it fast, robust, and efficient, thereby depicting its compliance for real-time applications.

Proposed Work

The outer isothetic cover (OIC) of a 2D object of any shape can be obtained using the algorithm TIPS, given in Biswas et al. (2010). It may be noted that, since the algorithm TIPS cannot find out multiple polygons (for multiple objects), it has been modified in the proposed work to obtain the complete set of isothetic polygons for the objects present in the handwritten document. In Figure 1, two handwritten Bengali words chosen from a handwritten text, and their corresponding OICs and baselines detected by our algorithm are shown. The outer isothetic covers (shown in blue) in this example are obtained for grid size $g = 3$; the upper and the lower baselines (shown in red) are extracted by analyzing these isothetic covers.

Construction of the OIC

In order to segment out the words in a handwritten text, we construct the outer isothetic covers (OIC) of the corresponding handwritten document image after its binarization (Otsu, 1979). To do this, we impose an isothetic set of grid lines having the grid size $g$ on the binarized image. Let $Q_1, Q_2, Q_3,$ and $Q_4$ be the four quadrants incident at a grid point $p(i, j)$, as shown in Figure 2. The grid point $p$ is decided to be a vertex depending on how many of the quadrants have object containment. Interestingly, there arise $2^4 = 16$ different arrangements considering object containments of these four quadrants, which can be reduced to five cases. Let $C_q$ ($q = 0, 1, \ldots, 4$) denote the case of all the arrangements for which $q$ out of 4 squares are occupied by the object. If $p$ belongs to Case $C_q$, then it is a $90^\circ$ vertex of the isothetic polygon; and if it is a $270^\circ$ vertex, then it belongs to $C_p$. For Case $C_q$, if the diagonal quadrants are occupied, then $p$ is considered as a $270^\circ$ vertex; otherwise, $p$ is a non-vertex grid point lying on some edge of the polygon. For case $C_q$, $p$ is just an ordinary grid point lying outside the polygon, whereas, for case $C_p$, $p$ is a grid point lying inside the polygon.

Word Segmentation

As mentioned earlier, in order to draw multiple polygons corresponding to multiple words in a handwritten document image, we have modified the algorithm TIPS [24]. With a proper grid size, each polygon corresponds to a single word barring intermingled words, punctuation marks, or dot marks over certain characters of the alphabet.

The document is first binarized with a suitable threshold. Then the grid points are traversed in the row-major order until a $90^\circ$ vertex (‘start vertex’) is found. Subsequent grid points are
classified, marked as ‘visited’, and the direction is determined from each such grid point until the start vertex is reached. This completes the outer isothetic cover corresponding to a word. The procedure is iterated over the remaining set of unvisited grid points until the next 90° vertex is found, which will subsequently derive the polygon corresponding to another word. Finally, all the grid points are visited and the algorithm reports the vertex sequences of all the isothetic polygons corresponding to the words in the input document.

The set of extracted polygons may contain some small (in perimeter) polygons, which may be due to the dot over a character of the alphabet or some punctuation mark. These smaller polygons are easily identified by their perimeters. If two or more words are unusually close or unmanageably intermingled with respect to the grid size, then we get a single polygon for these words. However, the major strength of the algorithm is that words with usual inter-word spacing can be easily segmented from skewed handwriting without resorting to skew correction. Figure 3 shows a handwritten paragraph and the segmentation result of its constituent words. The polygons (shown in blue) represent the different words. It has also rightly extracted the small perimeter objects like dots and punctuation marks in Bengali, which can be useful for further analysis of the document, if required.

**Setting the grid size g:** In order that each isothetic polygon corresponds to a word and hence results in a sequence of vertices, specifying an appropriate grid size is necessary to get a correct output of word segmentation and subsequent baseline extraction. Based on the fact that the inter-character gap in a word is smaller than the inter-word gap in a handwritten document, we have designed our algorithm to adaptively change the value of g depending on the font size and word spacing, as follows.

Let \( P^{(g)}_k \) and \( P^{(g)}_{k+1} \) be two consecutive polygons in a line corresponding to the grid size \( g \). For each grid point \( p_k \in P^{(g)}_k \) (i.e., lying on the boundary of \( P^{(g)}_k \)), we find the nearestpoint \( p'_{k+1} \in P^{(g)}_{k+1} \) and store the corresponding distance, which is given by

\[
d(p_k, P^{(g)}_{k+1}) = \min \{d(p_k, p'_{k+1})\}
\]

where:

\[
d(p_k, p'_{k+1}) = \max(\mid x_k - x_{k+1} \mid, \mid y_k - y_{k+1} \mid)
\]

considering that \((x_k, y_k)\) and \((x_{k+1}, y_{k+1})\) are the respective coordinates of \( p_k \) and \( p'_{k+1} \). Then the distance between \( P^{(g)}_k \) and \( P^{(g)}_{k+1} \) is estimated as

\[
d(P^{(g)}_k, P^{(g)}_{k+1}) = \min_{p_k', p_{k+1}'} d(p_k', p_{k+1}')
\]

We consider all inter-polygonal distances and find their minimum and maximum, name-
Interestingly, as the value of \( g \) is gradually increased, \( d_{min}(g) \) increases and \( d_{max}(g) \) decreases. Each distance is classified to the Class \( C_{min}(g) \) or Class \( C_{max}(g) \), depending on its closeness to \( d_{min}(g) \) and \( d_{max}(g) \). For example, in Figure 4, for \( g = 2 \), we have \( d_{min}(g) = 2 \) and \( d_{max}(g) = 46 \); hence, the distance \( d = 4 \) is classified to \( C_{min}(g) \). Until the number of distances in \( C_{min}(g) \) falls below that in \( C_{max}(g) \), \( g \) is made to increase, and the lowest value of \( g \) for which \( |C_{min}(g)| < |C_{max}(g)| \) is considered in our algorithm for word segmentation. For example, in Figure 4, when \( g = 6 \), we have \( |C_{min}(g)| = 0 < |C_{max}(g)| = 1 \), and so \( g = 6 \) is chosen.

The algorithm \textsc{WordSegment} takes the input handwritten document as a digital object, \( A \) (Figure 5). Initializations are done in Step 1. The while loop (Step 2) continues as long as the condition \(|C_{min}(g)| > |C_{max}(g)|\) is satisfied. The set of isothetic polygons, \( P(A) \), is obtained.
using the procedure \textit{OIC} (Step 3). The distances between each two consecutive polygons in a line, \(P_{k}^{(g)}\) and \(P_{k+1}^{(g)}\), are obtained (Steps 4-5). The procedure \textit{DIST} computes the smallest distance between any two vertices in the two polygons. Once the inter-polygon distances are computed, the minimum and the maximum amongst these distances are determined by \textit{FINDMAXMIN} (Step 6). Finally, all the distances (computed in Steps 4-5) are classified using the procedure \textit{CLASSIFY}. The procedure \textit{CLASSIFY} (Step 8) classifies each distance \(d_{k}\) if the value of \(d\) is closer to \(d_{\text{max}}\) (checked by calling a procedure \textit{NEAREST}, not shown here for its obviousness), then it is in class \(C_{\text{max}}\) (Steps 2-3), otherwise \(d_{k}\) is closer to \(d\) and hence classified to \(C_{\text{max}}\) (Steps 4-5).

**Baseline Extraction**

The baselines, upper and lower, are determined after determining the word polygons. It is easy to see that (i) the number of vertices of any isothetic polygon (here, a word polygon) is always even (to be precise, it is a multiple of 4), and (ii) the number of its horizontal edges equals that of its vertical edges. Hence, if there are \(n = 2m\) vertices in the word polygon, \(P_{w}\), corresponding to the \(k\)th word, \(w_{k}\), then \(P_{w}\) contains \(m\) horizontal and \(m\) vertical edges. As the algorithm starts from the top-left \(90^{\circ}\) vertex of \(P_{w}\), namely \(v_{0}\), and traverses along the edges of \(P_{w}\) in a way such that the corresponding word \(w_{k}\) always lies left (Figure 2), the length of the horizontal edge \(e_{h}^{(2i+1)}\) \((i = 0, 1, \ldots, m)\) outgoing from the vertex \(v_{2i+1}\) of \(P_{w}\) and ending at the next vertex \(v_{2i+2}\) of \(P_{w}\) is computed. If the respective coordinates of \(v_{2i+1}\) and \(v_{2i+2}\) be \((x, y)\) and \((x_{2i+1}, y_{2i+2})\), then the value stored at the array \(E_{2i+1}\) is increased by the length of \(e_{h}^{(2i+1)}\), i.e., \(|x_{2i+2} - x_{2i+1}|\). Note that \(E\) is a 1-D array and \(E[j]\) is initialized as 0 for \(j = 0, 1, \ldots, m\) before envisaging each new word, \(w_{k}\). For each word \(w_{k}\), we reconsider a local coordinate system with its origin at \(v_{0}^{(0)}\) = (0, 0). Thus, after traversing \(m\) horizontal edges of \(P_{w}\) outgoing from the vertices \(v_{0}, v_{1}, \ldots, v_{2m}\) in succession, the entry \(E[j]\) contains the sums of lengths of all the horizontal edges whose ordinates (in the local coordinate system of \(w_{j}\)) are \(j\).

A detailed demonstration on a typical handwritten Bengali word is given in Figure 6. As evident from this figure, after considering the lengths of all horizontal edges in \(E\) (blue histogram in Figure 5), we find two local maxima in \(E\). As we traverse along \(E[0], E[1], \ldots\), the first maximum gives the upper baseline and the next maximum gives the lower. For example, in Figure 5, shows that \(E = \langle 2, 2, 15, 11, 2, 3, 3, 10, 8, 3, 3 \rangle\), whereby the upper baseline is at \(j = 10\). Figure 6 shows the upper and the lower baselines of the words of a part of the document shown in Figure 3. It may be seen, almost all the baselines (shown in red) are correctly detected by the algorithm.

The algorithm \textit{BASELINE} (Table 2) to detect the upper and the lower baselines of each word works on the output generated by the algorithm \textit{WORDSEGMENT}. For each word polygon \(w\), the length of each horizontal line, \(e_{h}\), is calculated (Step 3) and stored in an array \(E[y]\) (Step 4). The two baselines (upper and lower) correspond to the two maximum values (i.e., frequencies) in the array \(E[j]\), and are detected by calling the procedure \textit{FINDBASEMAX} (Step 5).

**Experimental Results**

The algorithm is implemented in C on an Intel(R) Core(TM)2 Duo CPU E4500 2.20 GHz machine, the OS being Mandriva Linux Release 2008. The algorithm is tested on a different variety of handwritten documents. Results of word segmentation and baseline detection on a Bengali handwriting have been already shown in Figure 7 with necessary explanations. To show the algorithm works on different handwriting traits, the algorithm is tested on various text samples written by five different persons. Figure 9 shows some results produced by our algorithm on the samples collected from these
Figure 4. Setting the grid size $g$ in an adaptive way

<table>
<thead>
<tr>
<th>$g$</th>
<th>$d_{\text{min}}$</th>
<th>$d_{\text{max}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>50</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>48</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>45</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>44</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>45</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>42</td>
</tr>
</tbody>
</table>

Results for $g = 1, 2, \ldots, 6.$

$$g = 3: d_{\text{min}} = 3, d_{\text{max}} = 45$$

Table 1. Algorithm WORDSEGMENT

<table>
<thead>
<tr>
<th>Steps:</th>
<th></th>
<th>Steps:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. $g \leftarrow 1,</td>
<td>C_{\text{min}}^{(g)}</td>
<td>\leftarrow 1,</td>
</tr>
<tr>
<td>2. while $</td>
<td>C_{\text{min}}^{(g)}</td>
<td>&gt;</td>
</tr>
<tr>
<td>3. $P(A) \leftarrow \text{OIC}(A, g)$</td>
<td>3. $C_{\text{min}} \leftarrow C_{\text{min}} \cup d_i$</td>
<td></td>
</tr>
<tr>
<td>4. for each $(P_k^{(g)}, P_{k+1}^{(g)})$</td>
<td>4. else</td>
<td></td>
</tr>
<tr>
<td>5. $d(P_k^{(g)}, P_{k+1}^{(g)}) \leftarrow \text{DIST}(P_k^{(g)}, P_{k+1}^{(g)})$</td>
<td>5. $C_{\text{max}} \leftarrow C_{\text{max}} \cup d_i$</td>
<td></td>
</tr>
<tr>
<td>6. $(d_{\text{min}}, d_{\text{max}}) \leftarrow \text{FINDMAXMIN}(d)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7. for each $d_i$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8. $\text{CLASSIFY} \leftarrow (d_i, d_{\text{min}}, d_{\text{max}})$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9. $g \leftarrow g + 1$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

five persons. As evident from these results, the proposed algorithm has significant strength and efficiency in detecting the words and their baselines irrespective of the scripting style.

The algorithm is also tested on English handwritings where it works with quite a high level of accuracy. Results of word segmentation and baseline extraction of two English handwriting samples are shown in Figure 10. It may be noted that in case of very small gaps between adjacent lines of handwritten documents, two words of two adjacent lines may lie “too close”, thereby giving rise to a single polygon. Such erroneous results are quite unlikely to occur, and exhibited in Figure 11.
Figure 5. Analysis of edge-length array $E[j]$ for a word polygon to detect the baselines

![Figure 5](image)

Figure 6. Baselines of the extracted words for $g = 3$

![Figure 6](image)

Table 2. Algorithm of baseline detection

<table>
<thead>
<tr>
<th>Steps:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. for $i = 0 \ldots m$</td>
</tr>
<tr>
<td>2. for edge $v_{2i+1}$, $v_{2i+2} \in W$</td>
</tr>
<tr>
<td>3. $e_{2i} \leftarrow x_{i+2} - x_i - 1$</td>
</tr>
<tr>
<td>4. $E[y_{2i+1}] \leftarrow E[y_{2i+1}] + e_{2i+1}$</td>
</tr>
<tr>
<td>5. FINDBASEMAX($E$)</td>
</tr>
</tbody>
</table>

Copyright © 2011, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
CONCLUSION

A novel technique to segment out words in a handwriting document using the concept of outer isothetic covers (OIC) for an appropriate grid size is presented in this paper. The procedure for selection of grid size, explained in this paper, uses an adaptive technique. The baselines of words can also be successfully extracted by analyzing the OICs of words. The word segmentation technique may be used to design an improved algorithm that will work efficiently even on a skewed document. Apart from the word segmentation, the OICs can be used to develop several other document processing tools, such as writer identification or handwriting analysis, skew detection, etc. As evident from this paper, the combinatorial image analysis, if applied in an appropriate way, is quite useful to solve various real-world problems of image processing and pattern recognition in general, and document image analysis in particular.

ACKNOWLEDGMENT

A preliminary version of this paper appeared in the Proceedings of 12th International Conference on Frontiers in Handwriting Recognition, Kolkata, India, 2010 (Sarkar et al., 2010).
Figure 8. Result of Word Segmentation (left, in blue) and baseline extraction (right, in red) of two English handwritten samples

Figure 9. Segmented words (in blue) detected by our algorithm on two handwritten English script
REFERENCES


---

**ENDNOTE**

1 The first edge $e_1 = \langle v_0, v_1 \rangle$ is always vertical as the algorithm traces a polygon with the object always lying to its left (Figure 2).

Mousumi Dutt graduated from St. Thomas’ College of Engineering and Technology, Kolkata, and received ME degree in Information Technology from Bengal Engineering and Science University, Shibpur. She is now pursuing PhD from the Department of Information Technology from Bengal Engineering and Science University, Shibpur. Her primary research interest is digital geometry and its application in image processing.

Aishariya Sarkar graduated from Narula Institute of Technology, Kolkata, and and received ME degree in Information Technology from Bengal Engineering and Science University, Shibpur. Her specific areas of research include digital geometry and image analysis.

Arindam Biswas graduated from Jadavpur University, Kolkata, India, and received his master’s and doctorate degree both from the Indian Statistical Institute, Kolkata, India, where he has submitted his doctoral thesis also. He is currently an Assistant Professor in the Department of Information Technology, Bengal Engineering and Science University, Shibpur, India. His research interests include digital geometry, image processing, approximate shape matching and analysis, medical image analysis, and biometrics. He has published over 40 research papers in international journals, edited volumes, and refereed conference proceedings, and holds one US patent.
Partha Bhowmick graduated from the Indian Institute of Technology, Kharagpur, India, and received his Masters and PhD from the Indian Statistical Institute, Kolkata, India. He is currently an Assistant Professor in Computer Science and Engineering Department, Indian Institute of Technology, Kharagpur, India. His primary research area is digital geometry, with applications to low-level image processing, approximate pattern matching, document image processing and analysis, shape analysis, and biometrics. He has published more than 60 research papers in international journals, edited volumes, and refereed conference proceedings, and holds 4 US patents.

Bhargab B. Bhattacharya received the BSc degree in physics from the Presidency College, Kolkata, the BTech and MTech degrees in radiophysics and electronics, and the PhD degree in computer science all from the University of Calcutta, India. Since 1982, he has been on the faculty of the Indian Statistical Institute, Kolkata, where he is a full professor. He held visiting positions at the Department of Computer Science and Engineering, University of Nebraska-Lincoln, USA, at the University of Potsdam, Germany, and at the Indian Institute of Technology Kharagpur, as VSNL chair professor. His research interests include logic synthesis and testing of VLSI and nano-scale circuits, physical design, digital geometry, and image-processing architecture. He has published more than 250 papers in archival journals and in refereed conference proceedings and holds ten US patents. He is on the editorial board of the Journal of Electronic Testing – Theory and Applications. He is a fellow of the Indian National Academy of Engineering, a fellow of the National Academy of Sciences, India, and a fellow of IEEE.