

Multi-oriented English Text Line Identification

U. Pal, S. Sinha, and B. B. Chaudhuri

Computer Vision and Pattern Recognition Unit
Indian Statistical Institute; 203 B. T. Road, Kolkata-108 India
umapada@isical.ac.in

Abstract. There are many artistic documents where text lines of a single page may have different inclinations (orientations). To enhance the ability of document analysis system, we have to extract text line in multiple orientations. In this paper, we propose a robust technique to detect English text lines of arbitrary orientation in a single document page. We propose here a bottom-up approach where the connected components are at first labelled. They are then clustered into word groups. Text lines of arbitrary orientation are identified from the estimation of these word groups. From an experiment of 3700 text lines, we obtained an accuracy of 98.3% by the proposed method.

1 Introduction

In Optical Character Recognition (OCR), the text lines in a document must be located before recognition. When the text lines in a document image are parallel to one another (single oriented documents) simple techniques like projection profile, component nearest neighbor clustering method [3,5] etc. are good enough to identify them. But there are many artistic documents where text lines may be printed in several orientations. Extraction of individual lines from these documents is more difficult since simple approaches like projection profile or component nearest neighbor clustering will not work in these documents.

The pieces of published work on line extraction from documents containing multi-oriented lines are few. Goto and Asu [2] proposed a local linearity based technique to identify text line orientations in English and Chinese documents. The main limitation of their method is that it cannot handle text of variable sized characters. Fletcher and Kasturi [1] proposed a method to extract text lines in arbitrary orientations from mixed text and graphics region of English documents. The character bounding box information and the Hough transform are used in this method. This method too cannot handle text lines of arbitrary size. In another method proposed by Hones and Litcher [4] line anchors are first found in the document image and then text lines are generated by expanding the line anchors. Recently, we proposed a robust multi-skew detection technique for two Indian scripts Bangla and Devnagari [6]. Most Bangla and Devnagari characters have horizontal line at the top. When two or more characters sit side by side to form a word, these horizontal lines touch and generate a long line called 'Matra' or 'Shirokekha'. Based on this Matra/Shirokekha feature the orientation of a line is determined. This method will not work on English because of absence of Matra/Shirokekha.

In this paper, we propose a robust technique to extract English text lines of arbitrary orientations with variable size and style, and to detect the skew of individual text line. Here individual text lines are assumed straight in arbitrary orientation. The procedure is as follows. At first, we check whether text in a document is written in white on black background. If yes, we convert white pixels to black, and black pixels to white. Next, the connected component labeling is done. Characters of individual words are then grouped based on a size independent association rule. Next, grouping of words with respect to individual lines is done based on the normal distance among the base-lines (reference line) of grouped words.

2 Component grouping technique

Before component grouping, we detect whether characters of a text line are written in black on white background, or white on black background. For example, see Fig. 1(a) where characters in some text lines are written in black on white background and characters in other text lines are written in white on black background. To detect the nature (black or white) of a background of a text line we first use component labeling and find the average area (Ψ) of the components. If the area of a component is greater than $\alpha \times \Psi$, we compute the number of black pixels in the component. If the number of black pixels covers 60% of the area of the component then we assume that the component is a background component. The white pixels inside the component represent the text. We convert the black pixels into white and white pixels into black in order to obtain black text on white background. From the experiment the value of α is chosen as 5.

During component labeling, the lowermost point, the left candidate point, the right candidate point and the component height for each labeled object are computed for faster processing in future steps. If multiple lowermost pixels are there then the leftmost one is chosen as lowermost point of the component. Left (right) candidate point of a component is the intersection point of the leftmost (rightmost) column and lowermost row of the component. In other words, left and right candidate points of a component are two lower corner points of the bounding box of the component. The candidate points (marked by small circles) of two characters 'T' and 'o' are shown in Fig.2. Based on these candidate points and the height of the components, labeled components are merged into groups of individual words. For a group, two *anchor* components (left and right anchor components) are maintained and average height (H) of the components in the group is computed. The left (right) anchor component is the leftmost (rightmost) component of the group.

The grouping is done as follows. First, an arbitrary component is chosen (here we choose the topmost component of the document) and a group is formed using this component. Initially, the left anchor component and right anchor component are set as the same because the number of component in this group is only one now. Next, for grouping a component, say C , we check (a) whether at least one of the candidate points of the component is situated in the candidate region (candidate

region is described later) of an existing group and (b) whether the height of the bounding box of C is greater than half of H of the group. If for the component C the above conditions are satisfied by an existing group G_i , then C is merged with G_i . The anchor components and the average height of the components of the group G_i are modified accordingly. Otherwise, a new group is created by the component C.

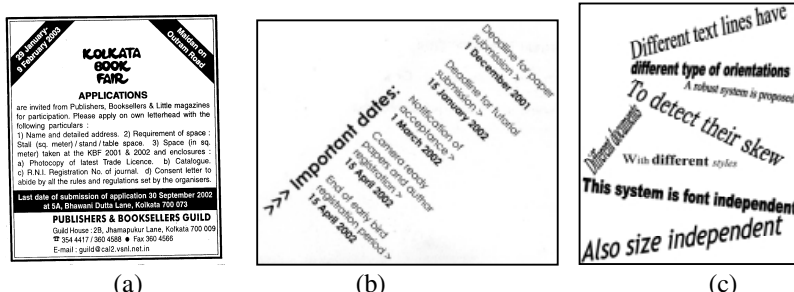


Fig.1: Examples of multi-oriented documents (a) from a newspaper (b) from ICPR-2002 call for papers (c) a synthetic document.

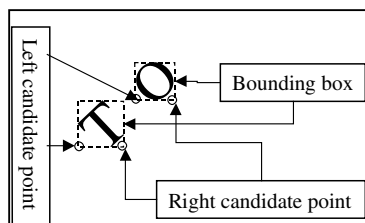


Fig.2: Candidate points of two characters are shown.

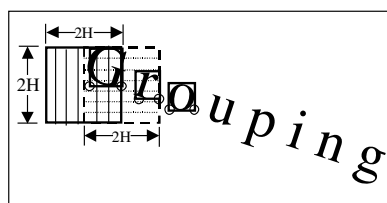


Fig.3: The approach of candidate region detection.

Modification of anchor components is done as follows. If the left candidate point of C is in the right side of the left candidate point of the right anchor component of the group G_i then C is assigned as the right anchor component of G_i . Similarly, modification of left anchor component is done if the left candidate point of the newly included component is in the left side of the left candidate point of the left anchor component of the group. In a similar way, the grouping of all the components of the document is done. Generally, we get N groups if there are N words in a document.

Note that components like dots of 'i' and 'j', comma(,) and fullstop (.) etc. are not included in the word grouping because of their smaller height. These components will be kept as isolated components. Because of the size independent technique, which is dynamic in nature, we are able to isolate these small components. If we use average height of all components of a document as a threshold for component selection then we may be able to isolate these dots and punctuation marks but some components with smaller height may also be isolated if a document contains text line of variable size. For example, see Fig.1(c). Here, all components of the word 'style' will be isolated along with the dots and fullstop if

we consider all components of the document for average component height computing.

Candidate Region detection:

From left candidate point of the left anchor component of a group, a H neighborhood (a square of side $2H$, H is the average height of the components in the group) is selected considering the left candidate point as centre. H neighborhood of the right candidate point of the left anchor component is also selected. These H neighborhoods of the two candidate points of a group are the candidate region of the left anchor component of the group. Similarly, candidate region selection of the two candidate points of the right anchor component is done. The candidate region of the left anchor component 'G' of a word 'Grouping' is shown in Fig.3. Here, candidate region for the left (right) candidate point is shown by solid (dotted) lines. Note that the area of candidate region is not always the same. Depending on the present average height of the components of a group the candidate region is selected. Because of this dynamic nature our method can handle text lines of arbitrary size and orientations.

3 Line extraction techniques

Line identification approach is based on the detection of base-lines of groups. The base-line of a text line is shown in Fig.4. If a text line is horizontal then its base-line extraction is very simple. The imaginary line passes through the maximum number of lowermost points of the components is the base-line. If a text line is not horizontal then its base-line detection is difficult. Here, we propose a robust technique for base-line detection of an arbitrary oriented line. If the number of components in a group is three or more then only we compute the base-line of the group. Based on the number of components in a group, the approach is divided into two categories: (a) base-line detection for a group containing three components and (b) base-line detection for a group with four or more components. Now let us discuss these detection procedures.

Base-line detection for a group containing three components: Suppose C_1 , C_2 and C_3 ($C_3 \geq C_2 \geq C_1$) denote the heights of the bounding box of three components of a group. We can get three (3C_2) pairs from these three components. Take a pair and draw a line (L) through the lowermost points of its two components. The perpendicular distance of the lowermost point of the third component is computed from L . Same procedure is done for other pairs. If any of these perpendicular distances is less than 0.1 than the line giving minimum perpendicular distance is the candidate-line. Else, we test the same procedure to obtain a candidate line from the uppermost points, instead of lowermost points. If a candidate line is obtained during the consideration of lowermost points then the candidate line itself is the base-line. Else, if a candidate line is obtained during the consideration of uppermost points then the line passing through the lowermost of the smallest component and parallel to the candidate line is the base-line. If we do not get a candidate line either from lowermost or from uppermost points then we draw a line (l_1) through the lowermost points of the smallest component (C_1) and one of the any two components (C_3 and C_2) of the group. Then through the topmost point of C_1 a

line l_2 parallel to l_1 is drawn. If l_2 touches at least one of the topmost point of C_3 and C_2 then l_1 is the Base line of the group. If no l_1 is obtained then we treat this group as an isolated group.

Base-line detection for a group containing four or more components:

Take two lowermost points A and B of any two components of a group and find the perpendicular distances of the lowermost points of the other components of the group from the line AB. If AB is the base-line then generally we can group these distances into two classes. Length of distances of one class will be very small (nearly zero). This will happen for those characters whose lowermost points lie on or near the line AB. Length of distances of other class will be higher because of the characters having descender. Individual variance of these two classes of distances will be nearly zero, and hence the sum of these variances will be nearly zero. For an illustration see Fig.5. The word image shown in this figure has six characters and hence we can get ${}^6C_2=15$ combination of two characters. We compute the variance of the distances obtained by considering two components of each of these combinations as A and B. The combination for which sum of individual variances of the two classes will be minimum is noted and the line passing through the lowermost points of the components of this combination is considered as the base-line. For example, consider the lowermost points of two components 'u' and 'e' of Fig.5(a) as A and B. The line drawn through A and B is shown in Fig.5(a). Length of perpendicular distances of the lowermost points of the other components are shown by small line segments. It can be noted that these length of perpendicular distances of different characters can be grouped into two classes. Length of one class of such distances is very small. This small distance is available because of the characters 'u', 'r', 'e' and 'l'. The other type of distances are obtained for the characters 'p' and 'y' of the line. If we compute variance of this individual class it will be nearly zero. Thus, the line through the line A and B is the base-line. If we consider the lowermost points of 'p' and 'y' then also we get two classes of distance with zero individual class variance (see Fig.5(b)). Thus, the line passing through lowermost points of 'p' and 'y' of Fig.5(b) may also be considered as base-line. It may be noted that in Fig.5(a) the line passes through two characters without descender, and in Fig.5(b) the line passes through the two characters with descender. Thus, the line passing through two characters without descender or the line passing through two characters with descender will be treated as base-line. The line passing through two characters of which one of them has descender will not be considered as base-line because in this case we will get different distances for different characters in the group (as shown in Fig.5(c)) and we cannot group these distances into two classes of nearly zero variance. Although we may get two base-lines (as discussed above) we consider the line which is above the other as the actual base-line. Thus, the line passing through the lowermost points of two components 'u' and 'e' of Fig.5(a) will be considered as the base-line.

This base-line is extended both-ways in such a way that its two ends lie on the bounding box of the group. We call this extended base-line as *reference line*. The leftmost and rightmost extreme points (we call these points as reference points) of this line are stored for future use. Thus, for a group we have a reference line and

two reference points. The reference line and reference point of a group is marked in Fig.5. Let S be the set of all such reference lines (RL) obtained from individual groups. Groups of two or less components will be considered as isolated groups.

To extract individual text lines, we have clustered the members of S corresponding to individual text lines. The clustering of S into groups of individual text lines is done as follows. Let C_L be the reference line of S obtained from a group having maximum number of components. From C_L or its continuation find the perpendicular distances to the left and right reference points of other RLs in S . The RLs corresponding to the text line containing C_L will have nearly equal distances from both the reference points. The RLs obtained by this criterion can be clustered in terms of these distances.

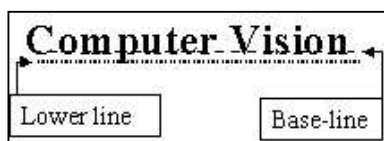


Fig.4. Base-line is shown.

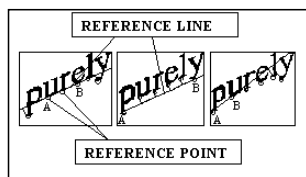


Fig.5: The approach of base-line detection.

The co-ordinates of the left reference point (x_{lc}, y_{lc}) of the leftmost group and the right reference point (x_{rc}, y_{rc}) of the rightmost group of each cluster are also stored for future convenience. To get the next cluster corresponding to another line, same procedure is used on S omitting those RLs which have already been clustered from S . At the end, K clusters are usually produced if there are K text lines in the document.

The clustering approach may be viewed as filling parametric space bins as in the Hough transform. For a reference line of S , perpendicular distances of its two reference points are computed from the C_L . Those RLs for which the distance of both the reference points is less than D are placed in a bin along with C_L . At the same time, the co-ordinates of the left reference point (x_{lc}, y_{lc}) of the leftmost group and the right reference point (x_{rc}, y_{rc}) of the rightmost group of individual clusters are updated accordingly. For example, suppose a new group q enters an existing cluster, say E , with left reference point (x_{lc}, y_{lc}) and right reference point (x_{rc}, y_{rc}) of a text line. If the left reference point (x_q, y_q) of q is to the left of (x_{lc}, y_{lc}) of E i.e. if $x_q < x_{lc}$ then we make $x_{lc} \leftarrow x_q$, and $y_{lc} \leftarrow y_q$. Else, no modification is done. The right reference point (x_{rc}, y_{rc}) of the cluster E is treated in a similar manner. The angle of the line joining (x_{lc}, y_{lc}) and (x_{rc}, y_{rc}) with the horizontal direction is computed to get the angle of orientation of the individual text line.

The value of D is estimated as follows. For a documents, character size smaller than 6 points is very unlikely. Then, the minimum spacing between two lines is 6 points (single spacing). If the document is digitized at p dpi then the minimum distance between two text lines will be $p \times 6/72$ pixels = $p/12$ pixels (since 72 points = 1 inch). For a document digitized at 300 dpi, we get minimum distance between two text lines as $300/12 = 25$ pixels. Based on this estimation we choose D as 25.

Now, we merge all the isolated component (elements of isolated groups are also considered as isolated components) which are not considered earlier for grouping.

To merge isolated components, we follow three steps. In the first step, we check whether the middle point (by middle point of a component we mean the middle point of the bounding box of the component) of an isolated component falls in the Core Area (CA) of a cluster or not. By CA of a cluster L we mean the rectangular region bounded by four lines L_b , L_t , L_l , L_r , where L_b is the reference line of L , L_t is parallel to L_b and passing through the top of the component having highest bounding box height among all components in L . L_l (L_r) is the line perpendicular to L_b and passing through the left (right) candidate point of the group. These four lines of a group are shown in Fig.6. If the middle point of an isolated element falls in CA of a cluster we merge the isolated element with that cluster. For example, in Fig.6 the isolated component 'a' and isolated group 'is' fall in the CA of the first line and hence we merge them with the line. Note that except 'a' and 'is' other words of this line are merged in a single cluster by the clustering technique stated above. In the second line isolated component 'A' does not fall in its CA. In the second step, we compute the normal distance of the lowermost point of the components (which are not grouped by the first step) to the reference lines of the existing clusters. If the distance of an isolated component is less than the height of the CA of a cluster then the component is marked as a possible candidate for merging with that cluster. If a component is marked by two or more clusters for possible merging candidate we proceed for the third step. Else, the component is merged with the respective cluster. In the third step we merge a component by nearest neighbor principle from CA. We compute Euclidean distance of the lowermost point of the component from the CA of each cluster. The component is merged with that cluster from which this distance is minimum. To compute the distance of a component from a cluster we compute Euclidean distance of the lowermost point of the component from the four corner points (shown by small circle in Fig.6) of the rectangular CA of the cluster. Minimum among these four distances is the distance of the component from the cluster. For example, the component 'A' of the second line of Fig.6 is identified by the two lines as the distance of the lowermost point of this component from the reference lines of the two lines is very small. By nearest neighbor technique this component is merged with the second line.

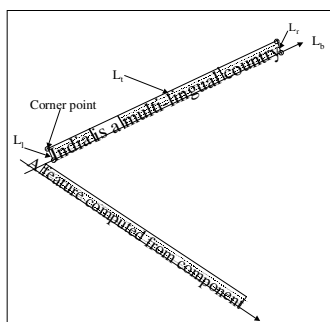


Fig.6. Isolated elements grouping. Words of a line cluster are marked by dotted lines.

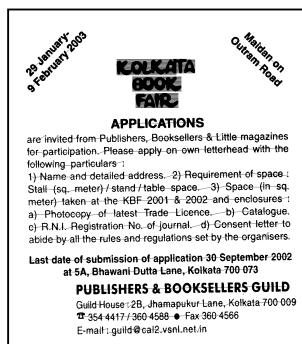


Fig.7: The line extraction result of the image shown in Fig.1(a)

4 Results and Discussion

For experiment, 3700 text lines were considered from different documents like magazines, newspapers, advertisements, computer printouts etc. Both the single and multi-oriented documents were considered for experiment. Among these 3700 text lines 900 lines were taken from multi-oriented documents (like Fig.1) and 2800 lines are from single oriented documents (where all text lines of a document have similar orientation). The images were digitized by flatbed scanner at a resolution of 300 dpi. For the experiment we consider single column document pages.

To check whether text lines are extracted correctly or not we connect all components of individual cluster by line segments (as shown in Fig.7). These lines are drawn through the mid point of the components of the clusters. By viewing the results on the computer's display we check the line extraction result manually. If all components of a text line are extracted by the algorithm we say line extraction is correct. From the experiment it is observed that overall accuracy of the proposed method is about 98.3%. From the experiment we also note that most of the errors occur due to one or two characters having ascenders or descenders.

One of the significant advantages of the proposed method is its flexibility. Our scheme is independent of font, size and style of the text line. For the flexibility testing, text lines with different popular fonts, sizes and styles (bold and italics) were considered.

The proposed method can segment text lines written in black on white background or white on black background. For example see Fig.7, where segmentation result of the image of Fig.1(a) is shown. In Fig. 1(a), characters in some text lines are written in black on white background and characters in some text lines are written in white on black background.

Acknowledgement: One of the authors (S. Sihna) acknowledges Ministry of Information Technology, Govt. Of India for his appointment under TDIL project.

References:

- [1] L. A. Fletcher and R. Kasturi, "A robust algorithm for text string separation from mined text/ graphics images", IEEE PAMI vol.10, pp.910-918, 1988.
- [2] H. Goto and H. Aso, "Extracting curved lines using local linearity of the text line", International journal of Document Analysis and Recognition, vol.2 pp. 111-118, 1999.
- [3] G. Nagy. S.seth and M. Viswanathan, "A prototype document image analysis syatem for technical journals.", Computer, vol.25, pp. 10-22, 1992.
- [4] F. Hones and J. Litcher, "Layout extraction of mixed mode documents", Machine vision and applications, vol.7, pp.237-246, 1994.
- [5] L. O'Gorman, "The document spectrum for page layout analysis", IEEE PAMI., vol. 15, pp. 1162-1173, 1993.
- [6] U. Pal, M. Mitra and B. B. Chaudhuri, "Multi-Skew Detection of Indian Script documents", In Proc. 6th ICDAR pp. 292-296, 20001.