

A COMPLETE SYSTEM FOR DETECTION AND RECOGNITION OF TEXT IN GRAPHICAL DOCUMENTS USING BACKGROUND INFORMATION

Partha Pratim Roy, Josep Lladós

Computer Vision Center, Universitat Autònoma de Barcelona, 08193 Bellaterra (Barcelona), Spain

Umapada Pal

Computer Vision and Pattern Recognition Unit, Indian Statistical Institute, Kolkata - 108, India

Keywords: Graphics Recognition, Optical Character Recognition, Convex Hull, Skeleton Analysis.

Abstract: Automatic Text/symbols retrieval in graphical documents (map, engineering drawing) involves many challenges because they are not usually parallel to each other. They are multi-oriented and curve in nature to annotate the graphical curve lines and hence follow a curvi-linear way too. Sometimes, text and symbols frequently touch/overlap with graphical components (river, street, border line) which enhances the problem. For OCR of such documents we need to extract individual text lines and their corresponding words/characters. In this paper, we propose a methodology to extract individual text lines and an approach for recognition of the extracted text characters from such complex graphical documents. The methodology is based on the foreground and background information of the text components. To take care of background information, water reservoir concept and convex hull have been used. For recognition of multi-font, multi-scale and multi-oriented characters, Support Vector Machine (SVM) based classifier is applied. Circular ring and convex hull have been used along with angular information of the contour pixels of the characters to make the feature rotation and scale invariant.

1 INTRODUCTION

The interpretation of graphical documents does not only require the recognition of graphical parts but the detection and recognition of multi-oriented text. The problem for detection and recognition of such text characters is many-folded. Text/symbols many times touch/overlap with long graphical lines. Sometimes, the text lines are curvi-linear to annotate graphical objects. And the recognition of such characters is more difficult due to the usage of multi-oriented and multi-scale environment. We show a map in Fig.1 to illustrate the problems.

Automatic extraction of text/symbols in graphical documents like map is one of the fundamental aims in graphics recognition (Fletcher and Kasturi, 1988), (Cao and Tan, 2001). The spatial distribution of the character components and their sizes, can be measured in a number of ways, and fairly reliable classification can be obtained. Using Connected Component (CC) analysis (Fletcher and Kasturi, 1988), (Tombre et al., 2002) and some heuristics based on

text features isolated text characters can be separated from long graphical components. Difficulties arise however, when either there is text and symbol embedded in the graphics components, or text and symbol touched with graphics. Luo et. al (Luo et al., 1995) uses the directional mathematical morphology approach for separation of long linear segments from character strings. Cao and Tan (Cao and Tan, 2001) proposed a method for extracting text characters that are touched to graphics. It is based on the observation that the constituent strokes of characters are usually short segments in comparison with those of graphics.

There are a few pieces of published work on extraction of multi-oriented and curved text lines in graphical/artistic document. Due to curve nature of text lines, their segmentation is a challenging task. Goto and Aso (Goto and Aso, 1999) proposed a local linearity based method to detect text lines in English and Chinese documents. In another method, proposed by Hones and Litcher (Hones and Litcher, 1994), line anchors are first found in the document image and then text lines are generated by expanding the line an-

chors. These methods cannot handle variable sized text, which is the main drawback of the methods. Loo and Tan (Loo and C.L.Tan, 2002) proposed a method using irregular pyramid for text line segmentation. Pal and Roy (Pal and Roy, 2004) proposed a head-line based technique for multi-oriented and curved text lines extraction from Indian documents.

For the recognition purpose of multi-oriented characters from engineering drawings, Adam et al. (Adam et al., 2000) used Fourier Mellin Transform. Some of the multi-oriented character recognition systems consider character realignment. The main drawback of these methods is the distortion due to realignment of curved text. Parametric eigen-space method is used by Hase et al. (Hase et al., 2003). Xie and Kobayashi (Xie and Kobayashi, 1991) proposed a system for multi-oriented English numeral recognition based on angular patterns. Pal et al. (Pal et al., 2006) proposed a modified quadratic classifier based recognition method for handling multi-oriented characters.



Figure 1: Example of a map shows orientation of text line and their characters.

In our proposed method, combination of connected component and skeleton analysis has been used to locate text character layer. The portions where graphical long lines touch text are marked. Using Hough transform and skeleton analysis, these portions are analyzed for text part separation. To handle the wide variations of texts in terms of size, font, orientation, etc., our approach includes the background information of the characters in a text line. It guides our algorithm to extract text lines from the documents containing multi-oriented and curved text lines. To get this background portion we apply the water reservoir concept which is one of the unique features of our proposed methodology. For recognition purpose, to make the system rotation invariant, the features are mainly based on the angular information of the external and internal contour pixels of the characters, where we compute the angle histogram of successive contour pixels. Circular ring and convex hull have been used to divide a character into several zones

and zone wise angular histogram is computed to get higher dimensional feature for better performance. SVM classifier has been applied for recognition of multi-oriented and multi-scale characters.

The organization of the rest of the paper is as follows. Text layer extraction methodology are discussed in Section 2. The curve text line segmentation methods are explained in Section 3. Feature extraction for multi-oriented character recognition and Support Vector Machine are detailed in Section 4. Results and discussion are given in Section 5. And finally, conclusion is included in Section 6.

2 TEXT/SYMBOL EXTRACTION

For the experiment of present work, we considered real data from map, newspaper, magazine etc. We used a flatbed scanner for digitization. Digitized images are in gray tone with 300 dpi. We have used a histogram based global binarization algorithm to convert them into two-tone (0 and 1) images (Here '1' represents object point and '0' represents background point). The digitized image may contain spurious noise points, small break points and irregularities on the boundary of the characters, lead to undesired effects on the system. For removing these we have used a method discussed in (Roy et al., 2004).

2.1 Component Classification using CC Analysis and Skeleton Information

In map, text and graphics appear simultaneously. They frequently touch each other and sometimes overlap. Here, the aim is to separate them into two layers mainly *Text* and *Graphics* layers. We used the connected component analysis (Tombre et al., 2002) for initial segmentation of isolated text components. The geometrical and statistical features (Fletcher and Kasturi, 1988) of the connected component are good enough to group a component into one between text or graphics layer. For each connected component, we use a minimum enclosing bounding box which describes the height and width of the character shape. The components are filtered to be as a member of a text component based on its attributes (rectangular size, pixel density, ratio of dimensions, area). A histogram on the size of components is analyzed for this purpose. By a correct threshold selection obtained dynamically from the histogram, the large graphical components are discarded, leaving the smaller graphics and text components. In our experiment, the threshold T is considered as,

$$T = n \times \max(A_{mp}, A_{avg}) \quad (1)$$

where, A_{mp} and A_{avg} are frequency of most populated area and average area respectively. The value of n was set to 3 from the experiment (Roy, 2007). The problems arise when some characters (“joined characters”) cannot be split due to touching together. We integrate skeleton information (Ahmed and Ward, 2002) to detect the long segments and to analyze them accordingly. Long segments are assumed to be as part of graphics. From the skeleton, we measure each segment based on their minimum enclosing bounding box (BB). The length (L_s) of a segment is computed as,

$$L_s = \text{Max}(\text{Height}_{BB}, \text{Width}_{BB}) \quad (2)$$

If there exist a segment of $L_s \geq T$ in a component, we consider that component having graphical component. Using these skeleton and connected component analysis, we separate all the components into 5 groups namely, Isolated characters, Joined characters, Dash components, Long components and Mixed components (Roy et al., 2007). If there exists no long segment in a component then it is included into one of the isolated character/symbol, joined character or dash component group. Otherwise it is considered as mixed or long component.

2.2 Removal of Long Graphical Lines

The mixed components are analyzed further for the separation of long segments. We perform Hough Transform to detect the straight lines present in the mixed component. In Hough space, all the collinear pixels of a straight line will be found intersecting at the same point (ρ, θ) , where ρ and θ identify the line equation. Depending on accumulation of pixels the straight lines are sorted out. Some characters may touch with these straight lines. To separate the characters from the lines, we compute the stroke width L_w of the straight line by scanning following the equation of line. Then the portions of the line where the width is more than L_w are separated from straight line.

We assumed that the length of segments of the characters are smaller compared to that of graphics. Hence, the skeleton is analyzed to check the presence of long segments in mixed components. All the skeleton segments are decomposed at the intersection point. The segments having $L_s \geq T$ are chosen for elimination. The remaining portion after removal of long straight and curve line are considered as either isolated characters or joined characters according to their statistical features. We show the segmentation of long straight lines of Fig.2(a) in Fig.2(b) and removal of curve lines of Fig.2(c) in Fig.2(d) & Fig.2(e).

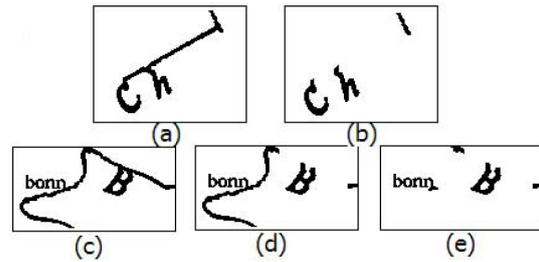


Figure 2: Separation of Text and Graphical lines.

3 TEXT LINE EXTRACTION

To detect text line in a document the water reservoir principle (Pal et al., 2003) has been used in the extracted text character layer. If water is poured from a side of a component, the cavity regions of the background portion of the component where water will be stored are considered as reservoirs of the component. Some of the water reservoir principle are shown in Fig.3. By top (bottom) reservoirs of a component we mean the reservoirs obtained when water is poured from top (bottom) of the component. The background region based feature obtained using water reservoir concepts help our line extraction scheme. The line detection process is discussed below.

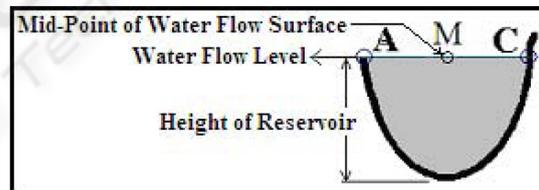


Figure 3: A top water reservoir and different features are shown. Water reservoir is marked by grey shade.

3.1 Initial 3-Character Clustering

For character clustering, the stroke width (S_w) of individual component (character) is computed (Roy et al., 2008b). For each component (say, D_1) we find two nearest components using boundary-growing algorithm (Roy et al., 2008b). Let, the two nearest components of D_1 are D_2 and D_3 . Also let, c_i be the center of minimum enclosing circle (MEC) of the component D_i . The components D_1 , D_2 and D_3 will form a valid 3-character cluster if they are a) similar in size b) linear in fashion and c) inter-character spacing is less than $3 \times S_w$. The size similarity is tested based on height information of the minimum enclosing bounding box of the component. Let, H_i be the height of a component D_i . The component D_1 will be similar in

size of its neighbour component D_2 if

$$0.5 \times H_2 \leq H_1 \leq 1.5 \times H_2 \quad (3)$$

Linearity is tested based on the angular information of c_1 , c_2 and c_3 . If the angle formed by $c_2c_1c_3 \geq 150^\circ$ then we assume the components are linear in fashion.

3.2 Grouping of Initial Clusters

From the initial clustering, we will get several 3-character clusters which are to be grouped together to have larger clusters. We make a graph (G) of these components of different clusters, where components of all the clusters are considered as nodes. An edge between two component nodes exists if they are from a same cluster and they are neighbors to each other. If the 3-character clusters come from the same text line then a node should have at most two edges. But, after making the graph a node may have 3 or more edges. This situation occurs when two or more text-lines cross each other or they are very close. The nodes having 3 or more edges are considered for removal. If the angle of a node (of degree 2) with respect to two connected nodes is less than 150° , then it is also marked for removal. These nodes are removed and all the corresponding edges are deleted from G . Thus the graph G will be split into sub-graphs or a set of components. In each of these sets, we will have components (nodes) having maximum 2 neighbor and they are linear in fashion. Elements of each sub-graph are considered as a large cluster.

3.3 Computation of Cluster Orientation

Using inter-character background information, orientations of the extreme characters of a cluster group are decided and two candidate regions are formed based on these orientations. For each cluster group we find one pair of characters from both of the extreme sides of the cluster. To find background information water reservoir concept is used. To do so, first convex hull of each character is formed to fill up the cavity regions of the character, if any. Next the resultant components are joined by a straight line through their centre of MEC. This joining is done to make the character pair into a single component to find the water reservoir in the background. Now water reservoir area is computed of this joined character in 8 directions at 45° interval. The area of water reservoirs in opposite directions are added to get the total background area in that direction. This is done for other 3 orientations. The orientation, in which maximum area is found, is detected and water flow-lines of corresponding reservoirs are stored. The mid-points of the water flow-lines of the two reservoirs are the candidate points.

This gives the orientation of the extreme characters of a cluster and it helps us to extend the cluster group for text line extraction.

3.4 Extension of Cluster Group

For each extreme pair characters of a cluster group, we know its candidate points and orientation. Using these information, we find a direction (D) perpendicular to this orientation and a key-point (K) which is along D , passes through the middle point of the candidate points and touches the bounding box of the cluster. For cluster extension to extract line, at first we generate a candidate region of rectangular mask at the key-point. Candidate regions for two key points of a cluster 'INTRODUCING' are shown in Fig.4(c). Let α be the set of candidate clusters and isolated characters obtained from the text layer. We use a bottom-up approach for line extraction and the approach is as follows. First, an arbitrary cluster (say, topmost left cluster) is chosen from α and a line-group L is formed using this cluster. Two candidate regions are detected from this cluster. For each line-group we maintain two anchor candidate regions (ACR): left and right ACR. Since at present L has only one cluster, the left and right ACRs of L are the two candidate regions of the cluster. Next, we check whether there exists any extreme character of another cluster or individual component whose a portion falls in these ACRs of the line-group. If the orientation of extreme character of another cluster is similar and their size is similar, then we include it in L . The ACRs are modified accordingly. The extension of this line-group continues in both sides, till it does not find any component or cluster in any ACR or it reaches the border of the image.

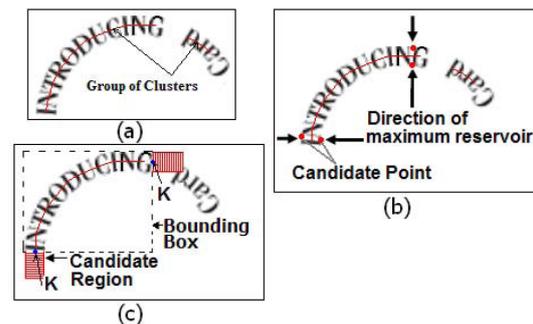


Figure 4: Example of candidate region detection from a cluster. (a) Two clusters (b) Candidate points of the two extreme pairs of characters for the cluster "INTRODUCING" (c) Key points are marked by 'K'. Candidate region is marked by hatched-line box.

Components clustered into a single line-group are the members of a single text line. To get other text

lines we follow the same steps and finally we get N number of line-groups if there are N text lines in a document. Since, we initially excluded the punctuation marks. So, punctuation marks will not be included by the above approach. To include such components in the line-group of their respective text lines we use a region growing technique (Pal and Roy, 2004).

4 CHARACTER RECOGNITION

Text characters of a single line are found to be aligned in a curvi-linear way to describe long graphical lines. So, we need a rotation invariant feature for character recognition. The feature used in our experiment and recognition method are explained below.

4.1 Feature Extraction

For a given text character, internal and external contour pixels are computed and they are used to determine the angular information feature of the character. Given a sequence of consecutive contour pixels $V_1 \dots V_i \dots V_n$, of length n ($n \geq 7$), the angular information of the pixel V_i is calculated from the orientation of vector pairs V_{i-k}, V_i and V_i, V_{i+k} . For better accuracy, we take the average of 3 orientations for each pixel, considering $k=1, 2$ and 3 . The angles obtained from all the contour pixels of a character are grouped into 8 bins corresponding to eight angular intervals of 45° (337.5° to 22.5° as bin no. 1, 22.5° to 67.5° as bin no. 2 and so on). For a character, frequency of the angles of 8 bins will be similar even if the character is rotated at any angle in any direction. For illustration, see Fig.5. We divide a character into several zones and zone-wise angular information is computed to get higher dimensional features. Circular ring and convex hull have been used for this purpose (Roy et al., 2008a).

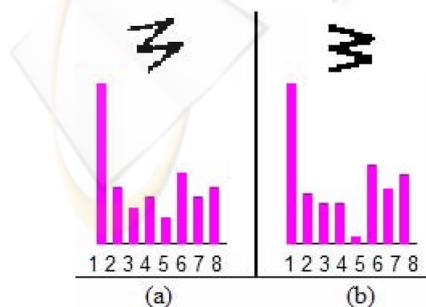


Figure 5: Input images of the character 'W' in 2 different rotations and their angle histogram of contour pixels are shown. The numbers 1-8 represent 8 angular bins.

4.1.1 Circular Ring based Division

A set of four circular rings is considered here and they are defined as the concentric circles considering centre as the centre of minimum enclosing circle (MEC) of the character and the minimum enclosing circle as the outer ring. The radii of the rings are in arithmetic progression. Let R_1 be the radius of MEC of the character, then the radii (outer to inner) of these four rings are R_1, R_2, R_3 and R_4 , respectively. Where $R_1 - R_2 = R_2 - R_3 = R_3 - R_4 = R$, where R is $R_1/4$. These rings divide the MEC of a character into four zones.

4.1.2 Convex Hull based Division

Convex hull rings are computed from the convex hull boundary. We compute 4 convex hull rings and we consider the outermost convex hull ring (say C_1) as the convex hull itself. Other 3 convex hull rings are similar in shape and computed from C_1 by reducing its size. The 2nd ring can be visualized by zooming out the C_1 with R pixels inside. Other 2 rings are computed similarly.

4.1.3 Reference Point and Reference Line Detection

If we compute information of angular histogram on the character portions in each of the ring, then we will get 32 (4 rings \times 8 angular information) dimensional feature. To get more local feature for higher accuracy, we divide each ring into few segments. To do such segments, we need a reference line (which should be invariant to character rotation) from a character. The reference line is detected based on the background part of the character using convex hull property. The mid-point of residue surface width (RSW) of the largest (in area) residue of a character is found and the line obtained by joining this mid-point and the centre of MEC of the character is the reference line of the character. If there are two or more largest (in area) residue then we check the height of these largest residue and the residue having largest height is selected. If the heights of the largest residue are same, then we select the residue having maximum RSW. The mid-point of the RSW of the selected residue and centre of MEC of the character is the reference line. The mid-point of RSW is the reference point. If no residue is selected by above, we consider the farthest contour point (P_f) of the character from the centre of MEC and the line obtained by joining P_f and the centre of MEC is the reference line. Here, P_f is the reference point. A reference line can segment each ring into two parts and if we compute the feature on each

of the segment, then we will get 64 dimensional features ($4 \text{ rings} \times 2 \text{ segments} \times 8 \text{ angular information}$). If we take another reference line perpendicular to this reference line, then each ring will be divided into 4 segments and as a result, we will get 128 dimensional features ($4 \text{ rings} \times 4 \text{ segments} \times 8 \text{ angular information}$). See Fig.6, where two reference lines PP' and QQ' are shown. To get 256 dimensional features each ring will be divided into 8 block segments.

To get different segments sequentially we consider the segment that starts from the reference point as segment number 1 (say S_1). Starting from S_1 if we move anti-clockwise then the segments obtained from outer ring block (R_1-R_2) are designated as 1st, 2nd...8th. Similarly, from the (R_2-R_3) ring block, we will get 9th, 10th...16th segment. Other segments are obtained in similar way. To get size independent features we normalize them. For normalization we divide the number of pixels in each segment by the total number of contour pixels.

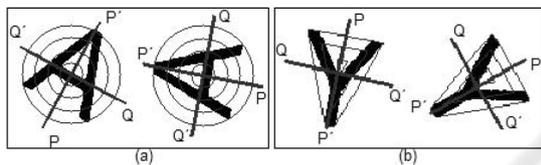


Figure 6: Reference lines PP' and QQ' are shown with (a) circular ring division in character 'A' (b) convex hull ring division in character 'Y'.

4.2 Recognition by SVM Classifier

We use Support Vector Machine (SVM) classifier for recognition. The SVM is defined for two-class problem and it looks for the optimal hyper-plane which maximizes the distance, the margin, between the nearest examples of both classes, named support vectors (SV_s). Given a training database of M data: $x_m \parallel m = \{1..M\}$, the linear SVM classifier is then defined as:

$$f(x) = \sum_j \alpha_j x_j \cdot x + b \quad (4)$$

Where, $\{x_j\}$ is the set of support vectors and the parameters α_j and b have been determined by solving a quadratic problem (Vapnik, 1995). The linear SVM can be extended to a non-linear classifier by replacing the inner product between the input vector x and the $SV_s x_j$, to a kernel function k defined as:

$$k(x, y) = \phi(x) \cdot \phi(y) \quad (5)$$

This kernel function should satisfy the Mercer's Condition (Vapnik, 1995). Some examples of kernel functions are polynomial kernels $(x \cdot y)^p$ and Gaussian kernels $\exp(-\|x - y\|^2/c)$, where c is a real number. We

use Gaussian kernel for our experiment. Details of SVM can be found in (Vapnik, 1995).

5 RESULT AND DISCUSSION

From our data we noticed that, text/symbols and graphical lines are many times together in touching/overlapped way. In text string, the arrangement of characters are of both linear and curvilinear. They appear in multi-orientation way to describe the corresponding symbols. We built a dataset of these extracted graphical characters for recognition purpose. The size of this dataset is 8250. Some examples of such data are discussed as follows. The text characters are of different font and size. To get an idea of data quality, we have shown some samples of a character 'R' in Fig.7. Both uppercase and lowercase letters of different fonts are used in the experiment. So we should have 62 classes (26 for uppercase, 26 for lowercase and 10 for digit). But because of shape similarity of some characters/digits, here we have 40 classes. We are considering arbitrarily rotation (any angle up to 360 degrees) so, some of the characters like 'p' and 'd' are considered same since, we will get the character 'p' if we rotate the character 'd' 180 degrees.



Figure 7: Some images of character 'R' from the dataset are shown.

To check whether a text line is extracted correctly or not we connect all components by line segments that are clustered in an individual line. These line segments are drawn through the center of the minimum enclosing circles of the components. By viewing the results on the computer's display we check the line extraction results manually. To give an idea about different ranges of accuracy of the system we divide the accuracy into two categories: (a) 97-100% (b) $\leq 97\%$. Accuracy of line extraction module is measured according to the following rule. If out of N components of a line, M components are extracted in favor of that line by our scheme then the accuracy for that line is $(M \times 100)/N\%$. The results are given in Table1.

Table 1: Text line segmentation result.

Number of lines	% of components
325	97-100%
30	$\leq 97\%$

One of the significant advantages of the proposed line extraction method is its flexibility. Our scheme is independent of font, size, style and orientation of the text lines. As we mentioned earlier, our assumption is that, distance between two lines of a document is greater than inter-character distance of the words. But sometimes distance between two words of two different text lines is very small and hence our method generates errors in some of these cases. Another drawback of our method is that it will not work if the characters are broken and that broken part cannot be joined through preprocessing. Here, neighborhood component selection will not be proper. So, direction from water reservoir concept cannot give the candidate region properly and errors occur. Also, our proposed method may not work properly if there are many joining characters in a string.

For recognition, the dataset has been tested using cross validation technique. For this purpose, we divided the dataset into 5 parts. We trained our system on 4 parts of the divided dataset and tested on remaining part of the data. From the dataset, we have obtained 96.54% (95.78%) recognition accuracy using circular (convex hull) based feature of dimension 256. Recognition accuracy obtained from circular and convex hull features with their different dimension are given in Table2. From the experiment we noted that better accuracy can be achieved combining circular and convex hull features. Combining circular and convex hull features of 256 dimension each we got 512 dimension feature. Using this 512 dimensional combined feature we achieved 96.73% accuracy from our SVM classifier in this dataset. From the experiment we also noticed that better results were obtained in case of bigger font-size characters.

Table 2: Character recognition result.

Feature Type	Feature Dimension		
	32	128	256
circular ring	90.54	96.01	96.54
convex hull	82.77	93.76	95.78

In Fig.8(b) we have shown the detected text lines and the recognition result of corresponding text characters of Fig.8(a). Here, all the text lines have been extracted correctly though there are some words in curvi-linear text lines, for e.g. "ATLANTIC OCEAN". The recognition result is very encouraging. Sometimes, due to "joining characters" and overlapping lines, the recognition of few characters are not correct. For e.g. in the word "Tagus", the joining character "gu" is mis-recognized as 'a'. From Fig.8(b), it may be noted there are some small graphical borders which were not eliminated due to our CC

analysis and hence we got erroneous result. We also noticed that most of the errors occurred due to similar shape structures. We noted that highest error occurred from the character pair 'K' and 'k', 'f' and 't' and 't' and 'l' pair. This is because of their shape similarity. Other errors occurred mainly from noisy data where residue from convex hull has not been extracted properly. This wrong residue detection sometimes influences error. In comparison, we have checked that, Adam et al. (Adam et al., 2000) received 95.74% accuracy on real English characters, whereas our system performs better with 96.73%.

6 CONCLUSIONS

In this paper we proposed a complete system for graphical documents. Here, we separated text from graphical components and extracted the corresponding text lines. The multi-oriented text characters are recognized using convex hull information. From the experiment, we have obtained encouraging result.

ACKNOWLEDGEMENTS

This work has been partially supported by the Spanish projects TIN2006-15694-C02-02 and CONSOLID-ERINGENIO 2010 (CSD2007-00018).

REFERENCES

- Adam, S., Ogier, J. M., Carlon, C., Mullot, R., Labiche, J., and Gardes, J. (2000). Symbol and character recognition: application to engineering drawing. In *International Journal on Document Analysis and Recognition (IJDAR)*.
- Ahmed, M. and Ward, R. (2002). A rotation invariant rule-based thinning algorithm for character recognition. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Cao, R. and Tan, C. (2001). Text/graphics separation in maps. In *Proc. of International Workshop on Graphics Recognition (GREC)*.
- Fletcher, L. A. and Kasturi, R. (1988). A robust algorithm for text string separation from mixed text/graphics images. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Goto, H. and Aso, H. (1999). Extracting curved lines using local linearity of the text line. In *International Journal on Document Analysis and Recognition (IJDAR)*.
- Hase, H., Shinokawa, T., Yoneda, M., and Suen, C. Y. (2003). Recognition of rotated characters by eigen-space. In *International Conference on Document Analysis and Recognition (ICDAR)*.

Hones, F. and Litcher, J. (1994). Layout extraction of mixed mode documents. In *Machine vision and applications*.

Loo, P. K. and C.L.Tan (2002). Word and sentence extraction using irregular pyramid. In *Document Analysis Systems*.

Luo, H., Agam, G., and Dinstein, I. (1995). Directional mathematical morphology approach for line thinning and extraction of character strings from maps and line drawings. In *International Conference on Document Analysis and Recognition (ICDAR)*.

Pal, U., Belad, A., and Choisy, C. (2003). Touching numeral segmentation using water reservoir concept. In *Pattern Recognition Letters*.

Pal, U., Kimura, F., Roy, K., and Pal, T. (2006). Recognition of english multi-oriented characters. In *International Conference on Pattern Recognition*.

Pal, U. and Roy, P. P. (2004). Multi-oriented and curved text lines extraction from indian documents. In *IEEE Transactions on SMC - Part B*.

Roy, K., Pal, U., and Chaudhuri, B. B. (2004). A system for joining and recognition of broken bangla numerals for indian postal automation. In *In Proc. of 4th ICVGIP*.

Roy, P. P. (2007). An approach to text/graphics separation from color maps. In *M.S. thesis, CVC, UAB, Barcelona*.

Roy, P. P., Pal, U., Llados, J., and Kimura, F. (2008a). Convex hull based approach for multi-oriented character recognition from graphical documents. In *International Conference on Pattern Recognition*.

Roy, P. P., Pal, U., Llados, J., and Kimura, F. (2008b). Multi-oriented english text line extraction using background and foreground information. In *Proc. of Document Analysis Systems*.

Roy, P. P., Vazquez, E., Llados, J., Baldrich, R., and Pal, U. (2007). A system to segment text and symbols from color maps. In *Proc. of International Workshop on Graphics Recognition (GREC)*.

Tombre, K., Tabbone, S., Peissier, L., Lamiroy, B., and Dosch, P. (2002). Text /graphics separation revisited. In *Proc. of Document Analysis Systems*.

Vapnik, V. (1995). The nature of statistical learning theory. Springer Verlag.

Xie, Q. and Kobayashi, A. (1991). A construction of pattern recognition system invariant of translation, scale-change and rotation transformation of pattern. In *Trans. of the Society of Instrument and Control Engineers*.

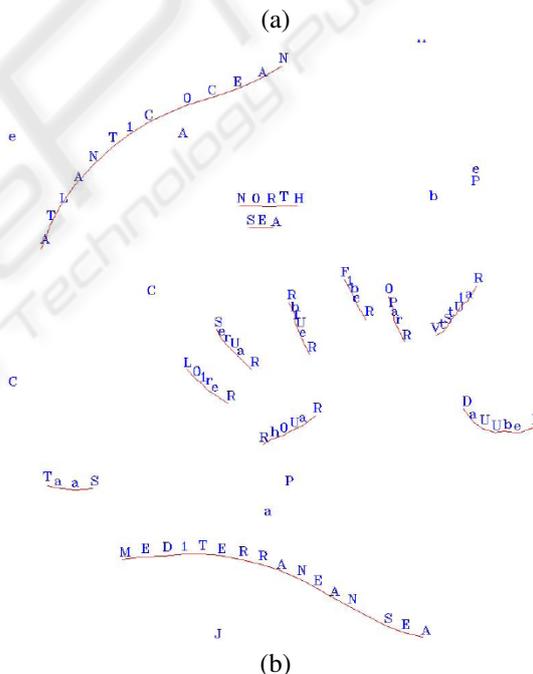
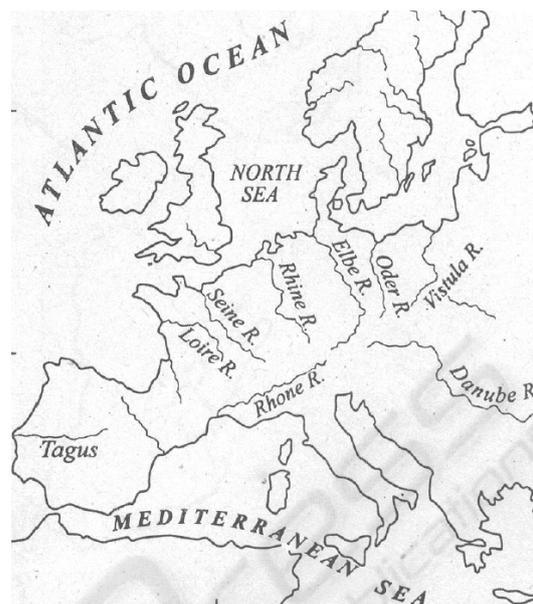


Figure 8: (a)Original Image (b) Line extraction and individual character recognition result of (a).