Text Segmentation from Land Map Images

Samit Biswas and Amit Kumar Das

CST Department, BESU, Shibpur, Howrah, India
samitbiet@yahoo.com, amit@cs.becs.ac.in

Abstract. Text segmentation from scanned map images is a challenging task as the texts in maps have wide variation in font, skew, spatial spread, touching components and a myriad background. We present a segmentation process by separating map components to three layers based on their areas. Further analysis, on spatial spread and node density from component graph leads to segmentation with encouraging performance.

1 Introduction and Related Work

Maps are ideal sources for getting geographical, historical, social and political information of a particular period. These pieces of information are mostly embedded in the maps as texts. However, the task of text extraction in maps is far more complex than other types of documents. The presence of multiple fonts, text touching with other line/line like structures representing other components, complex background and orientation of the texts have made it particularly challenging. Due to the complex nature of the problem, literature on text extraction from map images is not as rich as text segmentation from documents of other types. Related major works are summarised next which would be a cue to the backdrop of the approaches taken so far.

Tofani and Kasturi [1] extracted texts from color map images. At first the Authors' separate individual color overlays or alternatively separated each feature overlays (e.g., annotation, contour line, physical boundaries land use area shading etc.) based on their color. Then black/pink pixel layer of the map image and text extracted based on area thresholding was done. Li et al. [2] separated texts and line arts from USGS topographic maps by analysing the connected components. The maps were consists of street lines and labels. The constrained considered for component separation were defined such as space width between two characters(1.4 times their average width), characters of a text resides on a line, street casings (supposed that all street casings were connected in a map, not broken) were of 0.1 mm wide and separated 0.5 mm. Cao et al. [3] detected characters from complex documents (i.e., characters overlaps lines) using the differences of the length of line segments in character and line. Authors manually cut out the neighboring area of overlapping text with lines. The obtained piece of images were used to separate text and lines. Roy et al. [4,5] separated the maps into different layers according to color features and then connected component features, different size thresholds were used for this initial layering and

P. Maji et al. (Eds.): PReMI 2013, LNCS 8251, pp. 521–529, 2013.

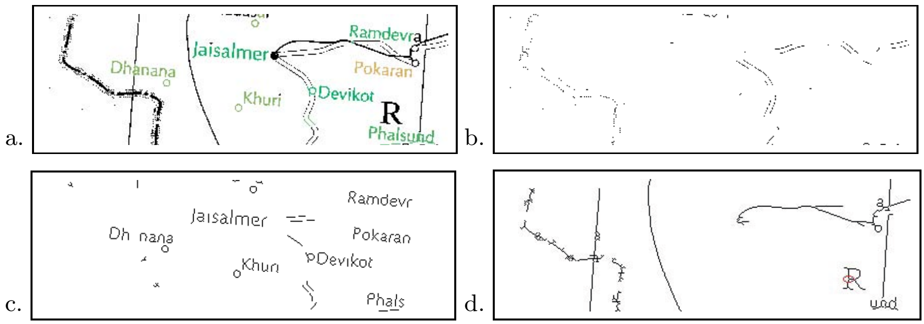


Fig. 1. a) Map image, colored texts are extracted using our approach; **b)** First layer; **c)** Middle layer; and **d)** Last layer; A junction point is shown encircled with red color

then skeleton information were used to identify text characters. Hough Transform was used to detect long straight lines. Curved lines separated by analyzing the length of major axis of decomposed components. Pezeshk and Tutwiler [6,7] used Multi Angled Parallelism (MAP) as a tool to separate text and graphics layer in maps. Authors are introduced a line representation method and a set of directional morphological operations to extract the text layer from intersecting linear features in scanned map images. In [8,9,10] authors extracted text from map images based on Radon transform based projection profile, CC analysis and graph based modeling respectively. These worked well for map images which contain *Bangla* text.

2 Proposed Approach

Map components are placed in three layers (first, middle and last) based on the size of the components. This layering is similar to the work reported in [1,4,5]. However, the novelty here is in the unique treatment of the components in middle and last layer based on spatial distribution of components (in the middle layer) and forming a graph by selecting nodes (pixels) on the components in the last layer. Scanned gray image, I_{gr} is first binarized (I_{bw}) and then thinned (I_{th}). I_{th} is analysed using the areas of the connected components (CC) to form multiple layers. Following subsections explain layering, analysis of each layer and combining the missing components.

2.1 Layer Formation Based on the Area of Each Component

The CCs are divided into three layers based on its area only. *First* layer contains most of the small components (such as dots, dashes, etc.). Map components which belongs to texts (words) reside within a moderately bigger range and are grouped in the *middle* layer. The *last* layer contains most of the big components; like boundary lines, rivers, portion of texts touched with other elements etc.

a
b

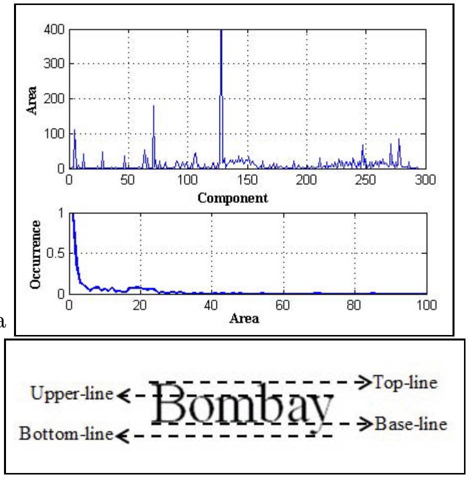


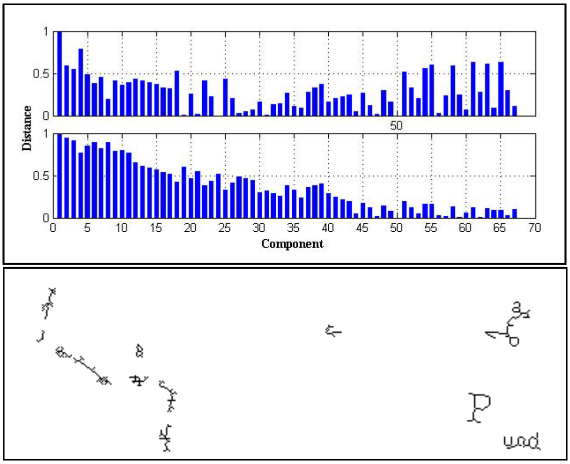
Fig. 2. a) Upper half shows Area of each CC; Lower half shows the histogram of the area of CCs; b) Spatial distribution of characters of text

Now, compute the area of each CC from I_{th}. Upper half of Fig.-2a shows the area correspond to each component of Fig.-1a. The histogram based on the area of those CCs is formed (see lower half of Fig.-2a). The histogram has one highest peak and the threshold (T_{h1}) for extracting components for the *first* layer lies at valley on the right side of it. CCs whose areas are less than T_{h1}, belong to first layer. CCs lying in the *middle* layer is grouped by another threshold, T_{h2} computed by taking the average area of all components excluding the components from the first layer. Middle layer components have got area in the range of T_{h1} to T_{h2} and components whose areas are crossing the threshold t_{h2} have been put to the last layer. For example see Fig.-1) which fairly supports these basis for forming three different layers.

The CCs of *first* layer are removed because most of these are noises. The *middle* layer is analysed because the components are isolated and have to be grouped together forming words or part of a word. As noted earlier few text portions may be grouped to the last layer as they are touching a bigger component. After grouping the connected components for each portion of text contained in *middle* layer, there is a need to get back the texts which may have been, inadvertently, put to the *last* layer during layer formation. This requirement can be observed from Fig-1c and 1d where portions of the texts have gone to the last layer as they are touching big components (lines in this case). Similarly, small lines, circles and similar components have been placed, wrongly, in the middle layer.

2.2 Analysis of Middle Layer and Distance Profile Generation

The characters belonging to a word are grouped based on the Distance Profile (DP). Before elaborating the DP we draw the attention of some characteristics of the text. The characters forming a word (most of the cases) are isolated



a

b

Fig. 3. a) Horizontal and vertical DP for i^{th} component, (here i=50); b) Modified last layer, Fig.-1d; text and text like components are retained

components. The main body of a character is largely confined within the base-line to top-line while the extended portion (outliers) coming out of the body is confined either between the upper-line and top-line or between the base-line and bottom-line (see Fig.-2b). Thus the top most points and bottom most points of each letter of a text reside on near vicinity of the top and base lines, respectively.

The analysis of the middle layer CCs is aimed at grouping discrete characters to a single unit (words in this case). This grouping is done by generating a DPs for each component which is a measure of the horizontal and vertical distances of the other components from the concerned component. At first for a component we find the topmost and bottom most points and drawing a straight line through these two points. Now we draw a straight line perpendicular to the first line and passing through the bottom most point. The second line is considered to be the base line for that particular component. Next, we draw perpendiculars to this base line from the bottom most points of all other components. The perpendicular distances thus obtained reflect the vertical distances of all other components with respect to the chosen component. Similarly, the distance of each perpendicular line projected on the base line reveal the horizontal distances of all other components with respect to the first component. Fig.-3a shows DP for a component; upper half shows the vertical distances while the lower half shows the horizontal distances. We generate the DPs for each and every component.

A group is formed around a component by taking all other components whose horizontal and vertical distances are less than or equal to lengths of major, m_1 and minor, m_2 axis of the concerned component, respectively. Finally, a regrouping among these groups have been done to find the chain connected set of components which is likely to be a word(s) or part of it.

2.3 Analysis of *Last* Layer

The last layer consists ideally the big components; mostly lines. Some of these lines may have touched a character and dragged the character to the last layer as an integral part of its own. The objective of analysing the last layer is to get back those characters from last to middle layer. First compute the nodes that can be formed from each component. A pixel is considered to be a node if it has one or more than two neighbours.The terminal points as well as the junction points are considered as nodes according to this logic. Note that for a thinned component most of the pixels are a part of a continuous run and are neither the terminal pixel or the junction pixel and they will have exactly two neighbours; which may be termed as left and right or top and bottom. We keep the node pixels of all the components removing all non-node pixels. In the next step form a weighted graph by connecting each node to all other nodes.

Edge weight (W_{ij}) between two nodes (v_i and v_j) is computed as per Equation-1 where number of neighbors at node v is $N(v)$ and $d(v_i, v_j)$ is the Euclidean distance between two nodes, v_i and v_j. The node density will be more in junction points (e.g., where a character is touching a line) and the edges will get more weights for short edges connecting closely spaced node cluster (junction points). This graph is partitioned using the edge weight as threshold to remove edges with low weight. Note that the edge weights will exponentially reduce and the threshold value may be set easily observing the knee point [9].

$$W_{ij} = \begin{cases} 1 & \text{if} & v_i = v_j \\ \frac{1}{d(v_i,v_j)}[1 - \frac{1}{N(v_i)+N(v_j)}] & \text{if} & N(v_i) \text{ and } \\ & & N(v_j) \neq 1 \\ 0 & \text{Otherwise} & \end{cases} \tag{1}$$

Consider a partition and project all the nodes back to the component(s) in last layer. If there is a continuity between any two nodes we keep that part of the thinned component of the last layer. This way the characters (with some extraneous part) which have been put to last layer may be segmented from the usual line like components of the last layer (compare Fig.-3b with Fig.-1d).

2.4 Combining the Missing Components

The characters have grouped to form words in middle layer. We also have isolated the characters connected to line(s) and were pushed to this layer truncating some of the words of the middle layer. These are to be combined now and for that purpose we have generated search strip from the extreme points of the components in the middle layer as elaborated next.

The extrema points (top-left, top-right, bottom-left, bottom-right, left-top, left-bottom, right-top and right-bottom) of all components from each subset (group of components) in the middle layer are computed first. Now we connect the top-left and bottom-left most extreme points of the leftmost component of a group; the perpendicular lines passing through the top-left and bottom-left

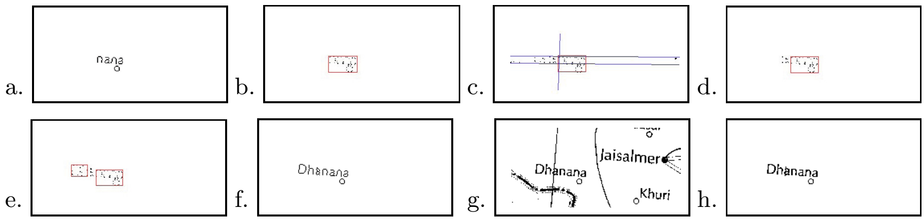


Fig. 4. a) A group of CCs considered from middle layer, b) Extreme points of the considered portion ('nana') of a word; c) Search strip; d) Extrema points grouped within the strip, e)After regrouping, marker-point set for a text; f) Associated set of connections among marker points (used as marker image, I_c); g) Original binarized input map image (used as mask image, I_{bw}), h)Reconstructed text, $R_C(I_c)$

Algorithm 1. Text Reconstruction from initial connections, I_c among marker-points of a set

---

Require: Marker, (I_c) and Mask, (I_{bw}).
Ensure: Reconstructed Text, $R_C(I_c)$.
1: Initialize h_1 to be the marker image, I_c
2: Create the structuring element: B
3: $Area_1$ = Total number of marker pixels in h_1; $C_1 = 0$
4: **Repeat** the following:
$h_{k+1} = (h_k \bigoplus B) \bigcap I_{th}$; $Area_{k+1}$=Total number of marker pixels in h_{k+1}
$C_{k+1} = Area_{k+1} - Area_k$
Until $C_{k+1} \geq C_k$
5: $R_C(I_c) = h_{k+1}$
6: **return** $R_C(I_c)$;

---

point on the connecting line specifies a band and has been extended across the image. Exactly the same thing is done on the rightmost component of a group. The two bands would roughly overlap and would make our search strip. All the extrema points of all components from middle and last layer within this search strip are stored together. The actions described are shown in Fig.-4a to c.

Next, we form two graphs using; i) all the extrema points of the components of the group and ii) extrema points under the search strip including the points of the group (that may not exclusively confined within the strip) and put edge weight using the Euclidean distance between the nodes. From the first graph we divide the edge weights into ten equally spaced bins and select the bin as threshold which has got maximum number of edge weights. Now from the second graph delete edges and corresponding nodes which are more than the selected threshold. This way we get a set of groups possibly enhanced with extreme points coming from other groups of middle and last layers. We combined these enhanced groups by merging them with common members (extreme points). Finally, a conditional morphological reconstruction (see Algorithm-1), using the extrema points of the groups, is done to form the text masks.

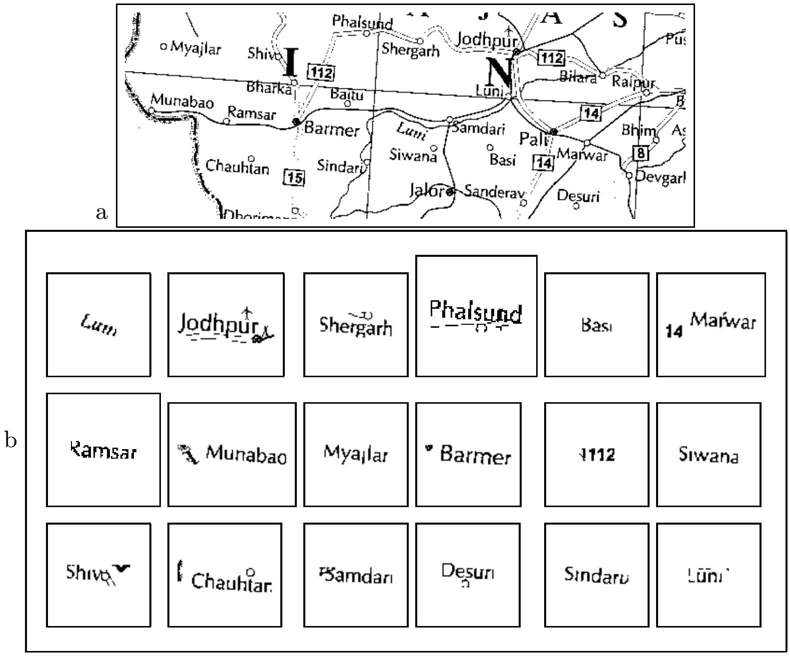


Fig. 5. a) A portion of a map image; b) Text images segmented (few are shown)

Table 1. Performance of the proposed method

No. of Map Images	True positive(%)	MDP(%)	IDP(%)	False Positive(%)
117	86.05	10.28	3.46	11.28

3 Dataset Details and Experimental Result

We have created our own dataset (BESUMAPS) with a variety of map images collected from sources like *websites*, *map books*[1] and *land surveyors*. The collected paper maps are scanned at 200 dpi in TIFF format. Each of the test map image consists of texts with myriad background consists of various intensity values, orientations, overlapping objects, intersected lines etc. At present the population in the BESUMAPS is 157 out of which 117 are in English and rest in different Indian languages including *Bangla*. The texts (place names etc.) have been manually extracted for each map and a ground-truth is made available. Fig.-5a shows a portion of a map which has been used as input to the proposed approach, while Fig.-5b shows the output.

Overall performance is shown in Table-1 for maps in English from BESUMAPS. *True Positive* (TP), *False Positive* (FP), *Incomplete Detection Percentage* (IDP) and *Mixed Detection Percentage* (MDP) are used for performance measure. TDP

[1] Oxford School Atlas, Oxford University Press.

indicates the extractions where texts have only been identified while MDP indicates texts with extraneous components. IDP gives an account of the texts (words) where we have missed a portion of the text (a character or part of it) and FP counts the extracted component which contain no text. Considering TP and MDP together the results are certainly encouraging. FP is rather high due to the presence of many small components which when grouped exhibits text-like characteristics.

4 Conclusion

In maps the major problem is that the lines do not merely touch the character; they pass through them and make them an integral part of the texts. This is an inherent problem and may be difficult to separate with the available algorithms generally used to solve the so called "touching character" problem. We are currently working towards a practical solution to dissociate the strongly bonded lines (or similar components) from the characters for the improved performance of our extractor. In the context of non-availability of well known map image databases and the small size of our database we are consolidating our efforts on creating a comprehensive database along with the ground truth. This would enable us to compare our proposed algorithm with the other algorithms already reported in the literature.

References

1. Tofani, P., Kasturi, R.: Segmentation of text from color map images. In: Proceedings of the Fourteenth International Conference on Pattern Recognition, vol. 1, pp. 945–947 (August 1998)
2. Li, L., Nagy, G., Samal, A., Seth, S., Xu, Y.: Cooperative text and line-art extraction from a topographic map. In: Proceedings of the Fifth International Conference on Document Analysis and Recognition, ICDAR 1999, pp. 467–470 (September 1999)
3. Cao, R., Tan, C.L.: Text/graphics separation in maps. In: Proceedings of the 4th International Workshop on Graphics Recognition Algorithms and Applications, pp. 245–254 (2001)
4. Roy, P.P., Vazquez, E., Lladós, J., Baldrich, R., Pal, U.: A system to segment text and symbols from color maps. In: Liu, W., Lladós, J., Ogier, J.-M. (eds.) GREC 2007. LNCS, vol. 5046, pp. 245–256. Springer, Heidelberg (2008)
5. Roy, P., Llados, J., Pal, U.: Text/graphics separation in color maps. In: International Conference on Computing: Theory and Applications, ICCTA 2007, pp. 545–551 (2007)
6. Pezeshk, A., Tutwiler, R.L.: Improved multi angled parallelism for separation of text from intersecting linear features in scanned topographic maps. In: ICASSP, pp. 1078–1081 (2010)
7. Pezeshk, A., Tutwiler, R.L.: Automatic feature extraction and text recognition from scanned topographic maps. IEEE T. Geoscience and Remote Sensing 49(12), 5047–5063 (2011)

8. Biswas, S., Das, A.K.: Text segmentation from scanned land map images using radon transform based projection profile. In: 2011 International Conference of Soft Computing and Pattern Recognition (SoCPaR), pp. 413–418 (October 2011)
9. Biswas, S., Das, A.K.: Fuzzy graph modeling for text segmentation from land map images. In: Proceedings of the Eighth Indian Conference on Computer Vision, Graphics and Image Processing, ICVGIP 2012, pp. 75:1–75:7 (2012)
10. Biswas, S., Das, A.K.: Text extraction from scanned land map images. In: IEEE/OSA/IAPR International Conference on Informatics, Electronics & Vision, Dhaka, pp. 231–236 (May 2012)