Text Line Extraction in Graphical Documents using Background and Foreground Information

Partha Pratim Roy[1], Umapada Pal[2], Josep Lladós[1]

[1]Computer Vision Center, Universitat Autònoma de Barcelona, 08193, Bellaterra,Spain
[2]Computer Vision and Pattern Recognition Unit, Indian Statistical Institute, Kolkata – 108, India
Email: partha@cvc.uab.es

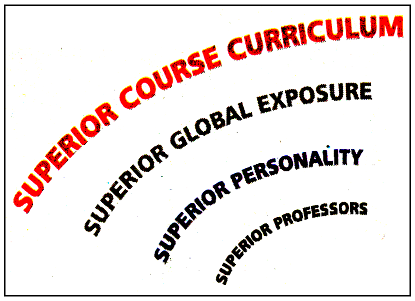
Abstract

In graphical documents (e.g. maps, engineering drawings), artistic documents etc., the text lines are annotated in multiple orientations or curvilinear way to illustrate different locations or symbols. For the OCR of such documents, individual text lines from the documents need to be extracted. In this paper, we propose a novel method to segment such text lines and the method is based on the foreground and background information of the text components. To effectively utilize the background information, a water reservoir concept is used here. In the proposed scheme, at first individual components are detected and grouped into character clusters in a hierarchical way using size and positional information. Next, the clusters are extended in two extreme sides to determine potential candidate regions. Finally, with the help of these candidate regions, individual lines are extracted. The experimental results are presented on different datasets of graphical documents, camera-based warped documents, noisy images containing seals, etc. The results demonstrate that our approach is robust and invariant to size and orientation of the text lines present in the document.

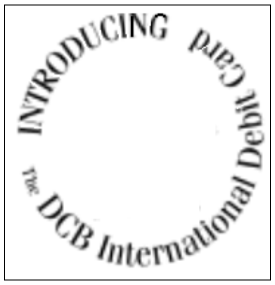
Key Words: Multi-oriented Text Line Segmentation, Artistic Documents, Graphical Document Analysis, Foreground-Background Information.

1. Introduction

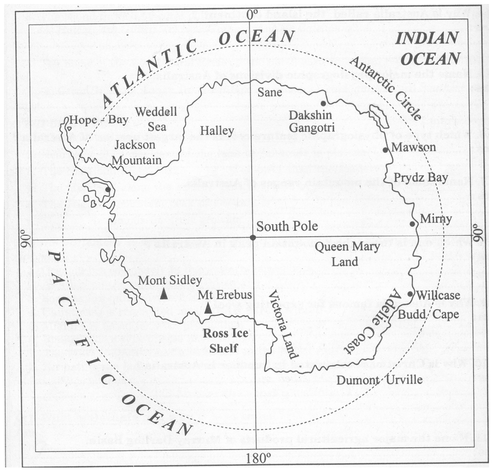
In Optical Character Recognition (OCR), the text lines in a document must be segmented before recognition. When the text lines in a document image are parallel to one another (single oriented documents) simple techniques like the projection profile, component nearest neighbour clustering method [2,4] etc. are good enough to segment individual lines. But there are many documents where text lines may be printed in several orientations (multi-oriented documents) or the text lines may be curved in shape. Examples of some such documents are shown in Fig.1. This paper deals with a novel technique to extract (segment) individual text lines from such multi-oriented or curved documents.



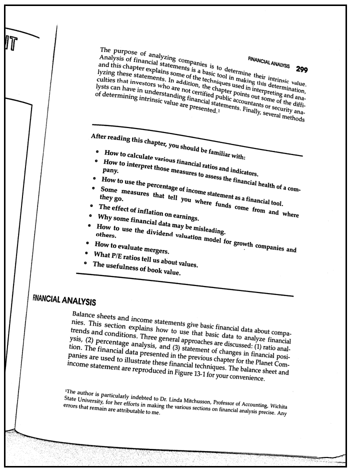
(a)



(b)



(c)



FINANCIAL ANALYSIS 299

The purpose of analyzing companies is to determine their intrinsic value. Analysis of financial statements is a basic tool in making this determination, and this chapter explains some of the techniques used in interpreting and analyzing these statements. In addition, the chapter points out some of the difficulties that investors who are not certified public accountants or security analysts can have in understanding financial statements. Finally, several methods of determining intrinsic value are presented.[1]

After reading this chapter, you should be familiar with:

- **How to calculate various financial ratios and indicators.**
- **How to interpret those measures to assess the financial health of a company.**
- **How to use the percentage of income statement as a financial tool.**
- **Some measures that tell you where funds come from and where they go.**
- **The effect of inflation on earnings.**
- **Why some financial data may be misleading.**
- **How to use the dividend valuation model for growth companies and others.**
- **How to evaluate mergers.**
- **What P/E ratios tell us about values.**
- **The usefulness of book value.**

FINANCIAL ANALYSIS

Balance sheets and income statements give basic financial data about companies. This section explains how to use that basic data to analyze financial trends and conditions. Three general approaches are discussed: (1) ratio analysis, (2) percentage analysis, and (3) statement of changes in financial position. The financial data presented in the previous chapter for the Planet Companies are used to illustrate these financial techniques. The balance sheet and income statement are reproduced in Figure 13-1 for your convenience.

[1]The author is particularly indebted to Dr. Linda Mitchusson, Professor of Accounting, Wichita State University, for her efforts in making the various sections on financial analysis precise. Any errors that remain are attributable to me.

(d)

Fig.1. Examples of documents containing multi-oriented and curved text-lines. (a) and (b) Text images from newspapers (c) Map image (d) Camera based warped image

There are many techniques to extract text lines from single oriented documents [13], but the published work on extraction of multi-oriented and curved text lines are few. Li et al. [24] proposed an approach for handwritten textline segmentation using level sets. Here, Gaussian filtering is used to estimate the probability density function (PDF) of pixel values and then level sets are initialized on these high PDF values. Growing and merging of level sets is then performed iteratively. Goto and Aso [1] proposed a local linearity based method to detect text lines in English and Chinese documents. In the method proposed by Hones and Litcher [3], line anchors are first found in the document image and then text lines are generated by expanding the line anchors. These methods cannot handle variable sized text, which is the main drawback of the methods. Loo and Tan [6] proposed a method using irregular pyramids for text line segmentation. This algorithm uses the inclusion of background information, concept of “closeness”, density of a word region, majority “win” strategy and the directional uniformity and continuity among words in a sentence. Recently, Bukhari et al. [23] proposed a line segmentation approach for camera based warped documents using active contour models. The segmentation technique uses several active contour models (baby snakes) and their convergence. Gatos et al. [12] proposed an algorithm based on text line and word detection for warped documents. The binary document is de-warped based on word rotation and translation according to upper and lower word baselines. Bai et al. [26] used a traditional perceptual grouping based algorithm for extracting curved line from logos and slogans. Chains of connected components are extended to the left and the right according to the local orientations of the text lines. Pal and Roy [5] proposed a head-line based technique for multi-oriented and curved text lines extraction from Indian documents containing Bangla and Devnagari scripts. Since the head-line feature is missing in English, this method can not be used for English line extraction. In other work Pal et al. [11] developed a system for English multi-oriented text line extraction estimating the equation of the text line from the character information. Main drawback of this method is that, it will not work for documents of curved text lines.

In graphical documents, however, text characters do not appear isolated. These documents usually contain many graphical lines besides text characters information and the text often touch/overlap with these long graphical lines. Detection of such text characters is difficult even there exist many text/graphics separation algorithms in literature [9, 16]. Text line segmentation techniques may provide useful information in this point of view. If some of the characters in a text line are missing, the remaining portion of text line (well-extracted characters) can direct us to locate the missing text characters. Thus, more emphasis can be provided to segment the touching/missing character.

Extraction of individual text lines from artistic or technical documents having multi-oriented or curved text is a difficult problem. In graphical documents, the components of text lines can be of different sizes and font styles. Again, the inter-character distance varies time to time for annotation purpose. For example, in maps shown in Fig.1(c), the inter character distance in the word “OCEAN” is different for annotation and it has different orientation in different location to annotate the graphical lines.

In this paper, we propose a line extraction technique from such artistic/technical documents. To handle documents of wide variations in terms of size, font, orientation, layout, etc., the proposed technique is based on the foreground and background information of the characters in a text line. Most researchers focus their attention only on foreground components and discard the background part (background part refers to those white areas of a binary image surrounding the actual black foreground). Here, we show how the background information can be utilized for multi-oriented line extraction.

The proposed approach is based on perceptual properties of text components. Text is a sequence of cavities following a regular distribution in a smooth path. These cavities/background information play an important role in our proposed method and guides our algorithm to extract individual lines from the documents containing multi-oriented and curved text lines. We use the background portion obtained between two consecutive characters of a line. To get this background portion we apply the water reservoir concept. Water reservoir is a metaphor to illustrate the cavity region of a component [7]. The use of background information and its computation through the water reservoir concept guides our proposed algorithm to segment text lines. In the proposed scheme, at first, individual components are detected and grouped into initial character clusters using their inter-component distance, size and positional information. Merging these initial character clusters, a proximity graph is created to have larger clusters.

Using inter-character background information, orientations of the extreme characters of a larger cluster are determined, and based on these orientations, two candidate regions are formed from the cluster. Finally, with the help of these candidate regions, individual lines are extracted.

The organization of the rest of the paper is as follows. A brief discussion on water reservoir concept used for line extraction is given in Section 2. The line extraction procedure is detailed in Section 3. We demonstrate our proposed algorithm on a variety of datasets including graphical documents and camera based documents in Section 4. Conclusions and future work are presented in Section 5.

2. Water Reservoir Concept

The water reservoir principle [7] is as follows. 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. Details of water reservoir and its different properties can be obtained in [7]. However, here we give a short description of different properties used in our proposed scheme to ease readability.

Top (Bottom) reservoir: By top (bottom) reservoirs of a component we mean the reservoirs obtained when water is poured from the top (bottom) of the component. A bottom reservoir of a component is visualized as a top reservoir when water is poured from top after rotating the component by 180°.

Left (Right) reservoir: If water is poured from the left (right) side of a component, the cavity regions of the component where water will be stored are considered as left (right) reservoirs. A left (right) reservoir of a component is visualized as a top reservoir when water is poured from the top after rotating the component by 90° clockwise (anti-clockwise).

Water reservoir area: The area of a reservoir is defined by the area of the cavity region where water will be stored. The number of points (pixels) inside a reservoir is computed and this number is considered as the area of the reservoir.

Water flow level: The level from which water overflows from a reservoir is called the water flow level of the reservoir (see Fig.2).

Reservoir surface width: The width of the reservoir at the flow-level of the reservoir is the reservoir surface width. In Fig.2, AC is the surface width of the reservoir.

Mid-Point of water flow surface: This is defined as the mid-point of the reservoir surface width. In Fig.2, M is the mid-point of the water flow surface.

Height of a reservoir: By height of a reservoir, we mean the depth of water in the reservoir.

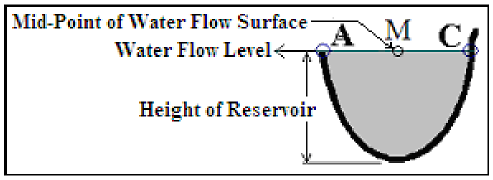
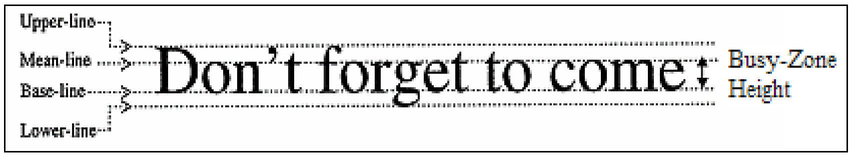


Fig.2. A top water reservoir and its different features are shown. Water reservoir portion is marked in grey.

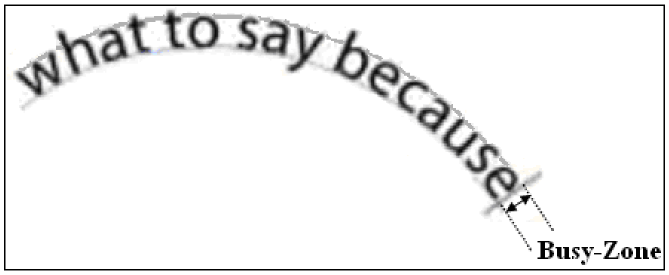
These background region based features obtained using the water reservoir concept help our line extraction scheme.

3. Text Line Extraction

An English text line can be divided into 3 zones: Upper zone, Middle zone or Busy zone and Lower zone. The Busy zone for an English text line is the zone between the mean-line and the base-line as shown in Fig.3(a). Busy zone is shown for curve line in Fig.3(b). This height information is important for the line extraction algorithm. It is used in our approach to determine the search zone (explained later) for the multi-oriented text line extraction.



(a)



(b)

Fig.3: Different parts of a text line shown in (a) straight line and (b) curve line.

In our line detection algorithm, we assumed that, the inter-character distance in a word of a line is smaller than the distance of this line from its neighbouring lines. The proposed line detection process is divided into 4 steps. These steps are a) Initial character clustering. b) Grouping of character clusters to form larger cluster. c) Candidate point selection of large clusters and d) Extension of large clusters for line extraction. The flow chart of text line segmentation is given in Fig.4. The documents are digitized in grey tone with 300 dpi. We apply Otsu binarization [28] method to convert a grey document into two-tone image and component labeling is performed on this binary image to detect individual components. The proposed scheme uses these connected components to extract the separate text lines. The main part of the line segmentation approach is illustrated within the dashed rectangle. In the following sections, we detail each step.

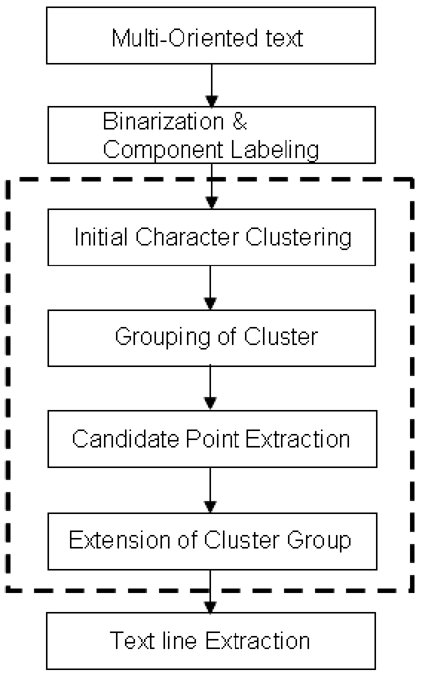

Multi-Oriented text
Binarization & Component Labeling
Initial Character Clustering
Grouping of Cluster
Candidate Point Extraction
Extension of Cluster Group
Text line Extraction

Fig.4. Flow chart of our approach for curved line extraction.

3.1. Initial Character Clustering

Our approach of line segmentation is similar to perceptual grouping. Text line is a sequence of similar sized character components following a regular distribution in a smooth path. The grouping of characters in text line is done in hierarchical ways in our approach. After detecting the text characters by connected component labeling, we group the neighbour characters in small clusters using the size and local linearity. Neighbour characters are selected using boundary growing algorithm which is performed as follows.

The external contour points of each character component are detected by a contour tracing algorithm. These contour points are expanded outwards iteratively by one pixel (8 pixel neighbour configuration) for boundary growing. Number of iteration to find nearest components is decided considering the size of the characters of the word. It is decided as Q = q x S_c. The multiple factor *q* is selected based on the type of dataset. We considered different values of q (e.g. 2,3,..8) in the experiment of different datasets to set q. In text documents, *q* is set to 4. In graphical documents (e.g. maps) where text characters are very sparse, this value is set to 6. These values are set based on the experiment. If after Q iterations of boundary growing, a character component does not touch other components, then no neighbour is found for the character.

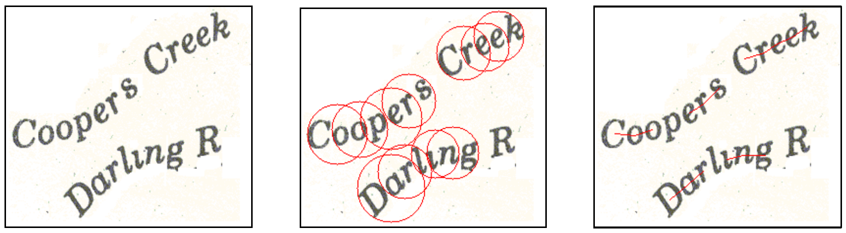
Next, the size (S_c) of individual components (characters) is computed for character clustering. The size S_c is calculated by finding the radius of a minimum enclosing circle of the component. For each component (say, D_1) we find its two nearest components using a boundary-growing algorithm (discussed later). Let the two nearest components of D_1 be D_2 and D_3. Also let, c_1, c_2 and c_3 be the centers of minimum enclosing circle (MEC) of the components D_1, D_2 and D_3, respectively. The components D_1, D_2 and D_3 will form a valid 3-character cluster if they satisfy both the size similarity and the local linearity. These are described as follows.

a) Size similarity: The size similarity is tested as follows. Let, H_{Di} be the size of a component D_i. The component D_1 will be similar in size to its neighbour component D_2 if

$$H_{D2}/1.5 \leq H_{D1} \leq 1.5 * H_{D2}$$

b) Local linearity: Local linearity (orientation) is tested based on the angular information of c_1, c_2 and c_3. If the angle $\angle c_2c_1c_3$ formed by the points c_2, c_1 and c_3 lies between 150º-180º, then we assume these 3 components are linear in nature. The angle between 2 neighbor components is measured and the range of the angle is considered between 0º to 180º. T_{ang} (150º-180º) is empirically chosen from experimental results for linearity detection. The components are neighbour when the features considering size and local linearity match and these two features are considered one by one. Here size is computed prior to local linearity feature.

Initial 3-character clustering of the image Fig.5(a) is shown in Fig.5(b). Each of these 3-character clusters is marked by circular ring. From the figure, it is to be noted that, there is no cluster among the components (o-p-e) and (r-l-i) because they fail to satisfy the angular criteria. The clusters (r-s-C), (s-C-r) and (n-g-R) are not formed because the inter-character spacing is large.



(a) (b) (c)

Fig.5. (a) Image shows different words from a map. (b) Initial clusters are marked by circular rings. (c) Large clusters are shown by lines.

3.2. Grouping of Initial Clusters

From the initial clustering, several 3-character clusters are obtained which will be grouped together by iterative growing. From the second text-line of Fig.5(b), we have different 3-character clusters such as (D-a-r), (a-r-l), (l-i-n) and (i-n-g). A graph analysis is performed to grow these clusters. A graph (G), described below is used here with all the components of different clusters, where components of all the clusters are considered as nodes. An edge between two component nodes is added if they are from the same initial cluster.

Formally, a graph G = (V, E) is created for growing these small 3-character clusters. Here, V = component and E = edges between components. Let there be T_s components in the document, thus, $|V| = T_s$. An edge, e_{ij} in E is formed by a component pair (C_i, C_j) if components C_i and C_j are neighbors.

Sometimes, by adding edges in such a way, 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 not considered in this step. For illustration, see Fig.6. Here, the cluster including the character ‘h’ of the word ‘Algorithm’ has two neighbours ‘t’ and ‘m’ from this word. Similarly, the character ‘s’ of word ‘analysis’ has neighbours ‘i’ from this word and ‘h’ from the word ‘Algorithm’. After grouping into initial clusters, ‘h’ will have 3 edges that are connected to its 3 neighbours (‘h-t’, ‘h-m’ and ‘h-s’). This node ‘h’ is marked for removal and is not considered for initial cluster grouping, since it may generate erroneous results in line extraction.

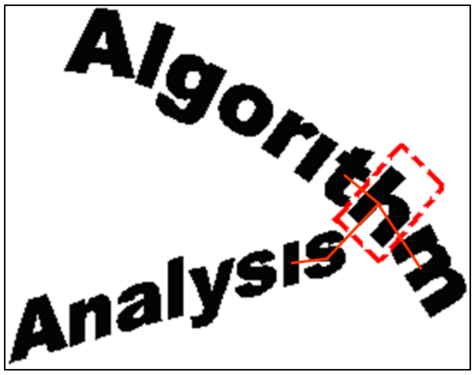


Fig.6. Example of text when a character (may) have 3 neighbours. Here, the character ‘h’ has 3 neighbouring characters ‘t’, ‘m’ and ‘s’.

The angle of each node (of degree 2) is calculated with respect to two connected nodes. If the angle is less than T_{ang}, this node is also marked for removal. For example, in Fig.5(c), the component ‘l’ of (D,a,r,l,i,n,g) does not satisfy the angle criterion with its neighbour components ‘r’ and ‘i’.

Thus, if for a node, either number of edges is greater than 2 or its angle with neighbour characters is less than T_{ang}, then the node is removed and all the edges of the corresponding node are deleted from G. Thus local linearity criterion between each of 3 character clusters is validated. Because of removal of some edges from the graph, G will be split into sub-graphs or a set of components. In each of these sets, we will have a chain of components which are linear in fashion. In other words, paths of characters are searched in the graph according to piece-wise linearity criterion. Each sub-graph is considered as a large cluster.

These large clusters represent different parts of text lines. These clusters are grown to merge other clusters in the same text line. Since, the clusters in the same line can be in different orientations, we use the cluster growing in local curvilinear direction. The continuation path of each cluster is followed by extending two extreme sides of the cluster. Hence, the orientation of the extension is computed from the extreme characters of the cluster group. For this extension purpose, we select candidate points from both ends of each cluster to follow the orientation. The selection of candidate points and the extension of the cluster are described in the following.

3.3. Candidate Point Selection

Using inter-character background information, orientations of the extreme characters of a cluster group are decided and based on these orientations, two candidate regions are formed from the cluster. For each cluster group we find one pair of characters from both of the two extreme sides of the cluster. Let, ‘NG’ and ‘IN’ be two such pairs of extreme characters of a cluster group and these pairs are shown in the 1st column of Fig.7. To find background information, the water reservoir concept is used. To do so, first convex hull of each character is found (a convex hull is shown in the 2nd column of Fig.7) and this is done to fill up the cavity regions of the character, if any. Next, the resultant components are joined (shown in 3rd column of Fig.7) by a straight line through their centres of MEC (Minimum Enclosing Circle). This joining is performed to make the character pair into a single component to find the water reservoir in the background part between them. It may be noted that, the line obtained by joining the centres of the MEC of two characters does not show the proper orientation of the characters always (e.g. when one character has ascender/descender parts and the other does not have ascender/descender parts) where as water reservoir concept can take care such characters properly. Because of this the water reservoir concept has been used here.

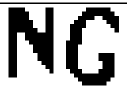
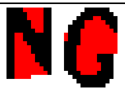
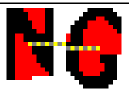
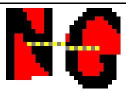
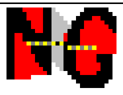
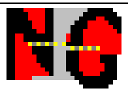
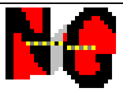
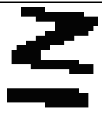
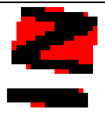
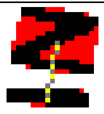
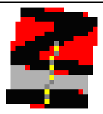
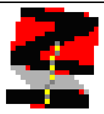
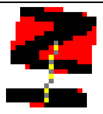
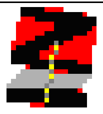
Two pairs of Components	Convex Hull of the components	Line joining through MEC centre	Reservoirs obtained in different directions			
			N_1-N_5	N_2-N_6	N_3-N_7	N_4-N_8
						
						

Fig.7. Water reservoir computation is shown on two extreme character pairs of the cluster “INTRODUCING” shown in Fig.10(a). The convex hull is marked by red and the water reservoir area is marked by light grey.

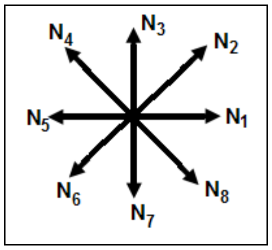


Fig.8. Directions of water reservoir computation.

Subsequently, the water reservoir area of this joined character is computed in 8 directions at 45° intervals as shown in Fig.8. Computation of the water reservoir when water is poured in the N_7 (N_3) direction is equivalent to the computation of top (bottom) reservoir. The areas of water reservoirs in direction N_3 and its opposite direction N_7 are added to get the total background area with respect to the orientation N_3N_7. This is done for other orientations N_1N_5, N_2N_6 and N_4N_8. The orientation, in which the maximum area is found, is detected and water flow-lines of the corresponding reservoirs are computed. The mid-points of the water flow-lines (discussed earlier in Section 2) of the two reservoirs are the candidate points. For illustration, see the last four columns of Fig.7, where water reservoirs in four orientations (N_1N_5, N_2N_6, N_3N_7 and N_4N_8) are shown. From the figure, it can be noted that, the maximum reservoir area is found for the pair ‘NG’ in N_3N_7 (top-bottom) directions whereas for the pair ‘IN’, the maximum reservoir area is obtained in the N_1N_5 (left-right) direction. This is because of the different orientations of these two extreme pairs of the cluster “INTRODUCING”.

For a pair, we get two candidate points and an orientation. The candidate points obtained from the character pairs ‘NG’ and ‘IN’ are shown in Fig.9. The orientation is obtained from the directions in which we get a maximum reservoir. For example, if we get a maximum reservoir area in the N_3 and N_7 directions, then N_3N_7 is the orientation. This orientation represents the direction of the extreme characters of a cluster, and it helps us to extend the cluster group for text line extraction. Generally, cluster groups should be extended in perpendicular to this orientation to obtain its neighbouring clusters of a text line. We also compute the distance between two candidate points and this distance gives the height information of the busy zone of the text line. The number of water reservoir direction is tested with different values (4, 8 and 12) of discrete angles in the experiment. If we use more direction for water reservoir than 8 directions, then computation time will be more. From our experiment we noted that water reservoir with 8 directions solves our purpose.

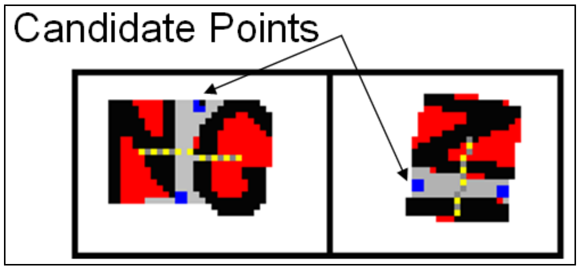
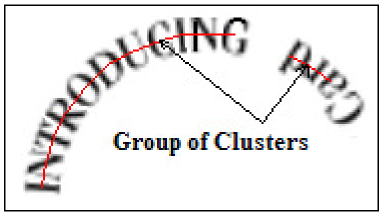


Fig.9: Candidate points are marked by dot. Red zones denote convex hull and light grey zones denote the water reservoir area.

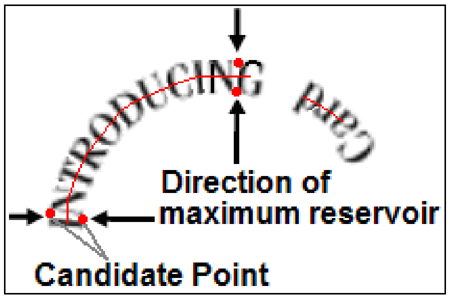
3.4. Extension of Cluster Group

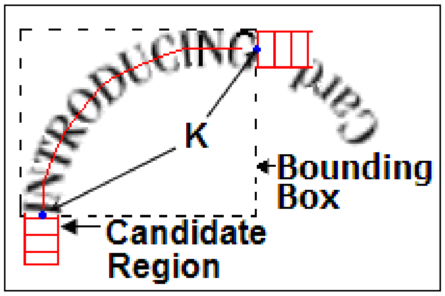
For each extreme character pair of a cluster group, we can know its candidate points and orientation. Let two candidate points of one extreme side of a cluster be M_1 and M_2 and its orientation be D. The distance between M_1M_2 is noted and this distance is called height info (*HI*). We find a line which is perpendicular to D and passing through the mid-point of M_1M_2. Let this line be XY and we call it the estimated line of extension. Note that, this line is detected based on the information of the extreme characters of a cluster and as a result, our line extraction scheme gives good results. Now, the line XY is extended in an outward direction until it reaches the bounding box of the cluster group. The point where the extended line meets the bounding box is noted and we call it a key point. This is done for the other parts of the other extreme sides of the cluster and we detect another key-point. So, for a cluster we have two key-points. Key points of the cluster 'INTRODUCING' are shown in Fig.10(c) and marked by 'K'.

Now, for each extreme side of a cluster we have the following: (a) estimated equation of the line of extension and (b) a key-point (K). For cluster extension to get individual line, we generate a candidate region for each key point of a cluster. Candidate region generation is performed as follows.



(a)





(b) **(c)**

Fig.10. Example of candidate region detection from a cluster. (a) Two clusters “INTRODUCING” and “Card” are shown. (b) Candidate points of the two extreme pairs of characters for the cluster “INTRODUCING” are shown. (c) Key points and candidate regions of cluster “INTRODUCING” are shown. Key points are marked by ‘K’. The candidate region is marked by hatched-line box.

Generation of candidate regions: Estimated equation of line of extension is extended up to a distance $TR = q \times HI$ in the outwards direction from the key point K. The path obtained during this extension is called as *extended candidate path* of K of the cluster. The value of q is discussed in Section 3.1. A rectangular mask of length $q \times HI$ and width HI is placed on the extended candidate path in such a way that the mask will be divided into two halves by the extended candidate path. This region of the rectangular mask is considered as the candidate region of the cluster for the key point K. In a similar way, the candidate region of the cluster for the other key point is computed. Candidate regions for two key points of a cluster “INTRODUCING” are shown in Fig.10(c) and they are marked by hatched line.

The reason to choose a *TR* equal to $q \times HI$ is as follows. In printed text *HI* generally represents the busy-zone height of a text line. So, if we assume *TR* as $q \times HI$ it is most likely that a part of the neighboring cluster will fall in one of the candidate regions of the current cluster.

Line Extraction: Candidate regions are used for line extraction and line extraction is performed as follows. Let α be the set of candidate clusters and isolated characters of an input image. 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 candidate regions (*CR*): left and right *CR*. Next, we check whether there exists any extreme character of a cluster or individual component whose portion falls in these *CR*s of the line-group *L*. Suppose, from α, we get a cluster (say, Nc) of which an end character falls in *CR*. If the *HI* of that end character of Nc and that of corresponding *CR* is similar and their corresponding orientation are also similar, then we include it in *L*. The candidate regions of the cluster Nc are noted and the *CR*s of *L* are modified based on the new cluster Nc. The *CR*s are modified as follows. If the right (left) candidate region of Nc falls in the left (right) *CR* of *L* then the left (right) candidate region of Nc is assigned as left (right) *CR* of *L*. Sometimes a portion of some isolated characters of α may also fall in the *CR*s. We consider such isolated characters for inclusion in *L*. If a portion of an isolated component falls in any *CR* of *L*, the component will be selected to join the line-group *L*, if the size similarity and linearity condition discussed in Section 3.1 are satisfied. If an isolated character is included in *L*, then the corresponding *CR* is updated considering the included character as the extreme character of the line-group. The extension of this line-group continues at both sides, till it does not find any component/cluster in any *CR*, or it reaches the border of the image.

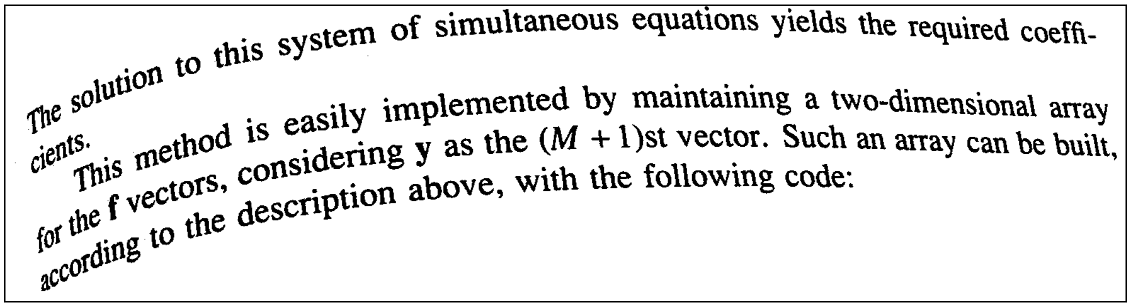
When several clusters are found in the candidate region, we check size and orientation similarity for each cluster with that of candidate region of *L*. The cluster having best matching in terms of size and orientation is selected to be included. To decide the best matching cluster, at first, the cluster having minimum height difference is chosen. If the chosen cluster satisfies the local linearity properly with that of *L* then the chosen cluster is considered as best matching cluster. Algorithmic steps of the extension of clustering group of the proposed scheme are given in Algorithm 1.

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 *T* number of line-groups if there are *T* text lines in a document. Finally, the punctuation symbols/small components which were left before are included into the nearest text lines according to the proximity. In pre-processing of line extraction, small symbol components ('.', ',' etc.) and background noise are filtered out based on their aspect ratio and pixel density. To include these punctuation symbols/small components in the line-group of their respective text lines we use a boundary growing technique. A punctuation symbol/small component is grown along its border until it touches any component of a line. A component is included to the text line to which it touches first during its boundary growing. Different results of the process are described in Fig.11.

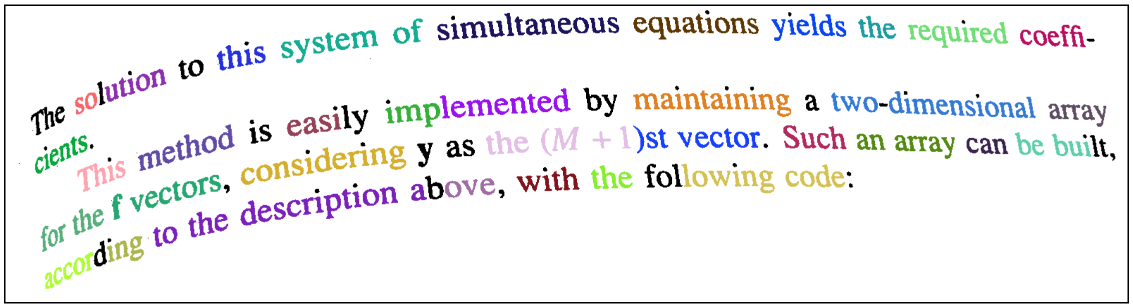
Algorithm 1: Extension of Cluster Group (G_c).

Step 1: Let M_1 and M_2 be the candidate points for one extreme side of a cluster (G_c) and D be the orientation of line obtained by joining M_1 and M_2. Let *HI* be the distance between M_1 and M_2. An equation XY is estimated perpendicular to D and passing through the mid-point of M_1M_2.
Step 2: The intersection point of XY and bounding box of G_c is detected to find the Key-Point (K) as shown in Fig.10(c).
Step 3: XY is extended up to length $TR=q\times HI$ in the outwards direction from the key point K, q is the multiplying factor of neighbor search area to get candidate region.
Step 4: A Candidate Region (*CR*) of rectangular size ($TR\times HI$) is considered at K as shown in Fig.10(c).
Step 5: If extreme character of another cluster falls in *CR*, we check the height and orientation information of the new cluster with respect to *CR*. If their sizes are similar and orientation of the candidate points of the new cluster is also similar to that of orientation D of G_c, then the cluster is included with G_c. When several clusters fall in the *CR*, the best matching cluster is considered for inclusion. Instead of a cluster, if an isolated character falls in *CR*, then it will be included in G_c if the character has similar size and local linearity with respect to G_c. Update G_c to get new M_1, M_2

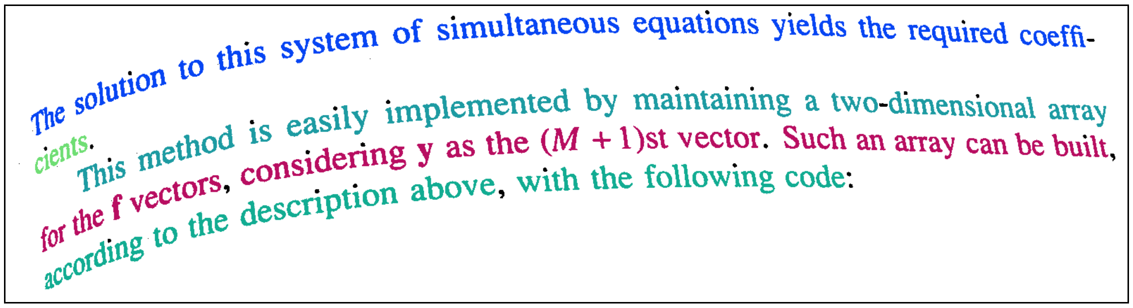
and D from the resultant cluster. If no cluster is found to include with G_c then `exit'. Else, goto Step 1.


The solution to this system of simultaneous equations yields the required coeffi-
cients.
This method is easily implemented by maintaining a two-dimensional array
for the $\mathbf{f}$ vectors, considering $\mathbf{y}$ as the $(M+1)$st vector. Such an array can be built,
according to the description above, with the following code:

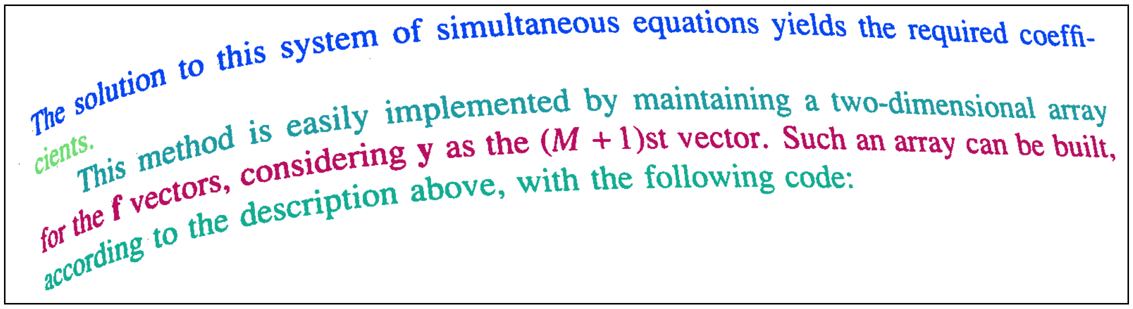
(a)


The solution to this system of simultaneous equations yields the required coeffi-
cients.
This method is easily implemented by maintaining a two-dimensional array
for the $\mathbf{f}$ vectors, considering $\mathbf{y}$ as the $(M+1)$st vector. Such an array can be built,
according to the description above, with the following code:

(b)


The solution to this system of simultaneous equations yields the required coeffi-
cients.
This method is easily implemented by maintaining a two-dimensional array
for the $\mathbf{f}$ vectors, considering $\mathbf{y}$ as the $(M+1)$st vector. Such an array can be built,
according to the description above, with the following code:

(c)


The solution to this system of simultaneous equations yields the required coeffi-
cients.
This method is easily implemented by maintaining a two-dimensional array
for the $\mathbf{f}$ vectors, considering $\mathbf{y}$ as the $(M+1)$st vector. Such an array can be built,
according to the description above, with the following code:

(d)

Fig.11. Different steps of our proposed algorithm are shown with a given camera based warped document shown in (a). (b) Large clusters. (c) Large clusters are merged to form text lines using water reservoir concept. (d) Other punctuations/small components are included in their respective text lines.

4. Experimental Results: Different Case Studies

For the experimentation of the present work, we considered real data from maps, newspapers, magazines etc. as well as synthetic data generated through the computer. The scanned text images that are obtained in gray scale are transformed into binary images. We have used a histogram based global binarization algorithm [28] to convert the data image 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, leading to undesired effects on the system. Since our method is based on cavity, so if there is a broken part, our method may not work. Hence we used a method to join the broken parts using the algorithm due to Roy et al. [27]. Since we consider in our work only small broken part, if there is a small broken part in the component this algorithm can join that part and our proposed method works well. We have done a quantitative experiment on this and noted that our method could not join 0.74% of the broken characters. We also noted that our method wrongly joins 0.32% cases.

We have used different datasets for our experimental results computation. To get the idea of the font sizes, we considered 10, 12, 16, 20, 26 and 30 point-size characters for the experiment. The results are discussed in the following sections. Some of the datasets have ground truth and some do not have ground truth. If ground truth is not available, then the result is checked manually by viewing the results. To check visually whether a text line is extracted correctly or not, we render all components that are clustered in an individual line by a single color. The colors for different text lines are selected with a random function generator.

4.1 Text Documents

Our experiments involve text documents which are obtained both from scanner and camera. The scanned dataset is taken from documents where text lines are parallel to each other and the text lines are horizontal. We considered 50 such documents to test. The camera based documents are taken from the CBDAR 2007 document image dewarping contest [14]. This document dataset consists of images captured with a hand held camera in uncontrolled environments. From this dataset, we considered 30 images containing text data. Some of them include graphical diagrams also. As our system can accept graphical diagrams, we used these documents without any other pre-processing and evaluated our approach on these documents. Segmented results of some of the images are shown in Fig.12. Detailed quantitative results obtained from our different dataset (text documents, geographical maps, engineering drawings and seal documents) are given in Fig.17.

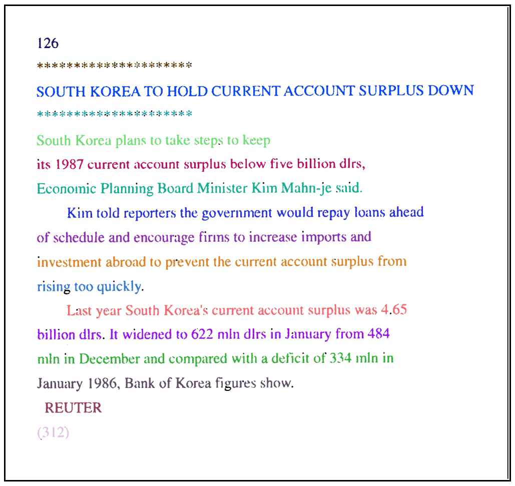
The errors occurred mainly due to the presence of broken text components. If characters in a text line are broken, and those broken components cannot be joined through pre-processing [27], then our approach may fail. In this case, neighborhood component selection will not be proper due to the size similarity feature. Hence, direction from the water reservoir concept cannot determine the candidate region properly and errors occur.

4.2 Graphical Documents

We have considered 20 different real geographical maps and 6 engineering drawing images to test our method. Images were digitized by a flatbed scanner at 300 dpì. They contain text lines of different scale and orientation. We have used one dataset of synthetic maps which are generated using an automatic system described in Delalandre et al. [22]. The backgrounds (graphical lines, boundaries etc.) of these maps are taken from a few real maps and the text words of country/river names are placed in the foreground using a set of rules. We show two different test images of the same background map in Fig.13(a) and (b). The graphical long lines i.e. geographical borders and rivers shown in these maps are kept fixed. The text portions are randomly placed and oriented to generate different synthetic documents.

Some real geographical maps are selected from historical archive. See Fig.13(c), where a portion of the document is shown. These images contain text printed in dark colour compared to the background. To obtain the text image, we get foreground information by converting the color image to a gray-level one and apply a threshold to this image for segmenting. The conversion to gray level is performed by transforming the RGB color model to YIQ model, where the luminance channel (Y) represents the gray-level image. It is achieved by the conversion equation,

$$Y = 0.299\times R + 0.587\times G + 0.114\times B$$



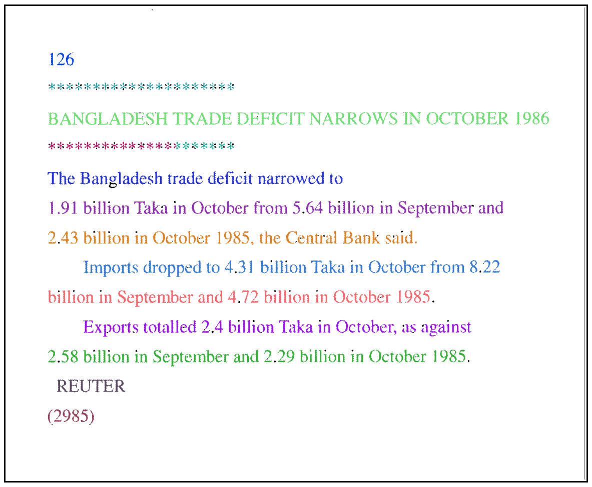
126

****

SOUTH KOREA TO HOLD CURRENT ACCOUNT SURPLUS DOWN

****

South Korea plans to take steps to keep
its 1987 current account surplus below five billion dlrs,
Economic Planning Board Minister Kim Mahn-je said.
Kim told reporters the government would repay loans ahead
of schedule and encourage firms to increase imports and
investment abroad to prevent the current account surplus from
rising too quickly.
Last year South Korea's current account surplus was 4.65
billion dlrs. It widened to 622 mln dlrs in January from 484
mln in December and compared with a deficit of 334 mln in
January 1986, Bank of Korea figures show.
REUTER
(312)



126

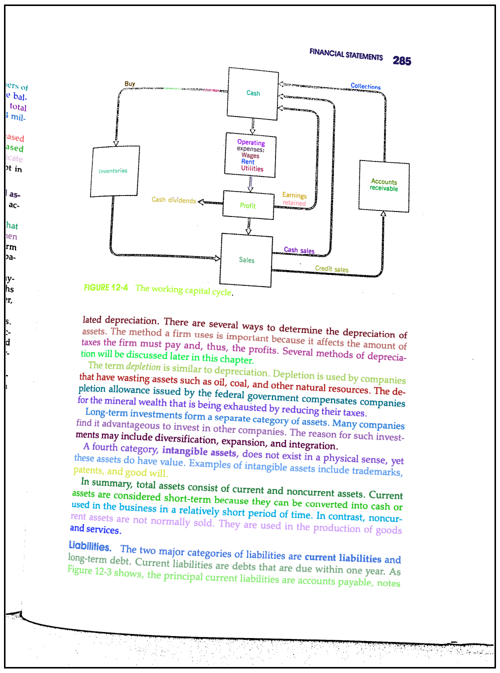
****

BANGLADESH TRADE DEFICIT NARROWS IN OCTOBER 1986

****

The Bangladesh trade deficit narrowed to
1.91 billion Taka in October from 5.64 billion in September and
2.43 billion in October 1985, the Central Bank said.
Imports dropped to 4.31 billion Taka in October from 8.22
billion in September and 4.72 billion in October 1985.
Exports totalled 2.4 billion Taka in October, as against
2.58 billion in September and 2.29 billion in October 1985.
REUTER
(2985)

(a) (b)



FINANCIAL STATEMENTS 285

Buy
Cash
Collections
Operating expenses: Wages Rent Utilities
Inventories
Accounts receivable
Cash dividends
Profit
Earnings retained
Sales
Cash sales
Credit sales

FIGURE 12-4 The working capital cycle.

lated depreciation. There are several ways to determine the depreciation of assets. The method a firm uses is important because it affects the amount of taxes the firm must pay and, thus, the profits. Several methods of depreciation will be discussed later in this chapter.

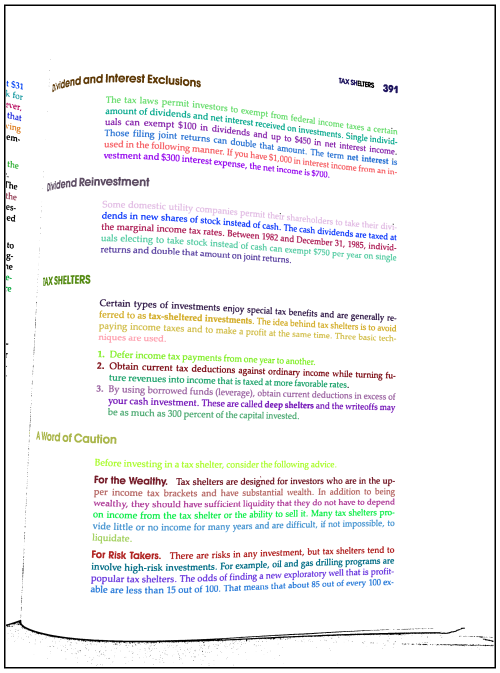
The term *depletion* is similar to depreciation. Depletion is used by companies that have wasting assets such as oil, coal, and other natural resources. The depletion allowance issued by the federal government compensates companies for the mineral wealth that is being exhausted by reducing their taxes.

Long-term investments form a separate category of assets. Many companies find it advantageous to invest in other companies. The reason for such investments may include diversification, expansion, and integration.

A fourth category, **intangible assets**, does not exist in a physical sense, yet these assets do have value. Examples of intangible assets include trademarks, patents, and good will.

In summary, total assets consist of current and noncurrent assets. Current assets are considered short-term because they can be converted into cash or used in the business in a relatively short period of time. In contrast, noncurrent assets are not normally sold. They are used in the production of goods and services.

Liabilities. The two major categories of liabilities are **current liabilities** and long-term debt. Current liabilities are debts that are due within one year. As Figure 12-3 shows, the principal current liabilities are accounts payable, notes



Dividend and Interest Exclusions — TAX SHELTERS 391

The tax laws permit investors to exempt from federal income taxes a certain amount of dividends and net interest received on investments. Single individuals can exempt $100 in dividends and up to $450 in net interest income. Those filing joint returns can double that amount. The term **net interest** is used in the following manner. If you have $1,000 in interest income from an investment and $300 interest expense, the net income is $700.

Dividend Reinvestment

Some domestic utility companies permit their shareholders to take their dividends in new shares of stock instead of cash. The cash dividends are taxed at the marginal income tax rates. Between 1982 and December 31, 1985, individuals electing to take stock instead of cash can exempt $750 per year on single returns and double that amount on joint returns.

TAX SHELTERS

Certain types of investments enjoy special tax benefits and are generally referred to as **tax-sheltered investments**. The idea behind tax shelters is to avoid paying income taxes and to make a profit at the same time. Three basic techniques are used.

1. Defer income tax payments from one year to another.
2. Obtain current tax deductions against ordinary income while turning future revenues into income that is taxed at more favorable rates.
3. By using borrowed funds (leverage), obtain current deductions in excess of your cash investment. These are called **deep shelters** and the writeoffs may be as much as 300 percent of the capital invested.

A Word of Caution

Before investing in a tax shelter, consider the following advice.

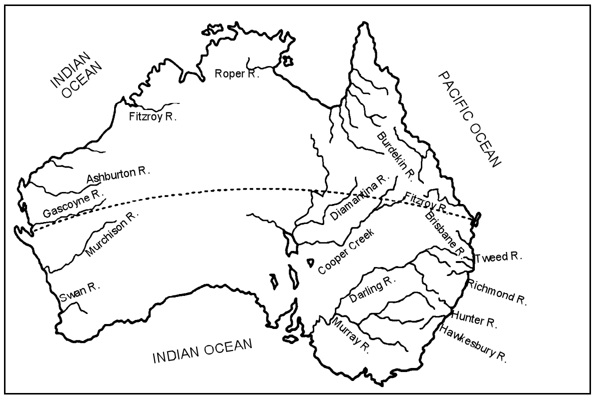
For the Wealthy. Tax shelters are designed for investors who are in the upper income tax brackets and have substantial wealth. In addition to being wealthy, they should have sufficient liquidity that they do not have to depend on income from the tax shelter or the ability to sell it. Many tax shelters provide little or no income for many years and are difficult, if not impossible, to liquidate.

For Risk Takers. There are risks in any investment, but tax shelters tend to involve high-risk investments. For example, oil and gas drilling programs are popular tax shelters. The odds of finding a new exploratory well that is profitable are less than 15 out of 100. That means that about 85 out of every 100 ex-

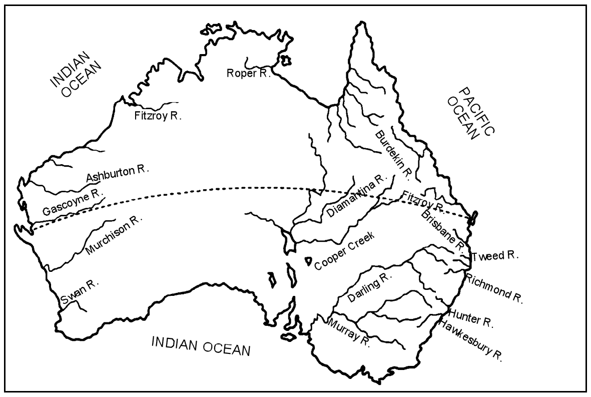
(c) (d)

Fig.12. Line segmentation result in (a) & (b) Normal text document. (c) & (d) Camera-based warped text documents.

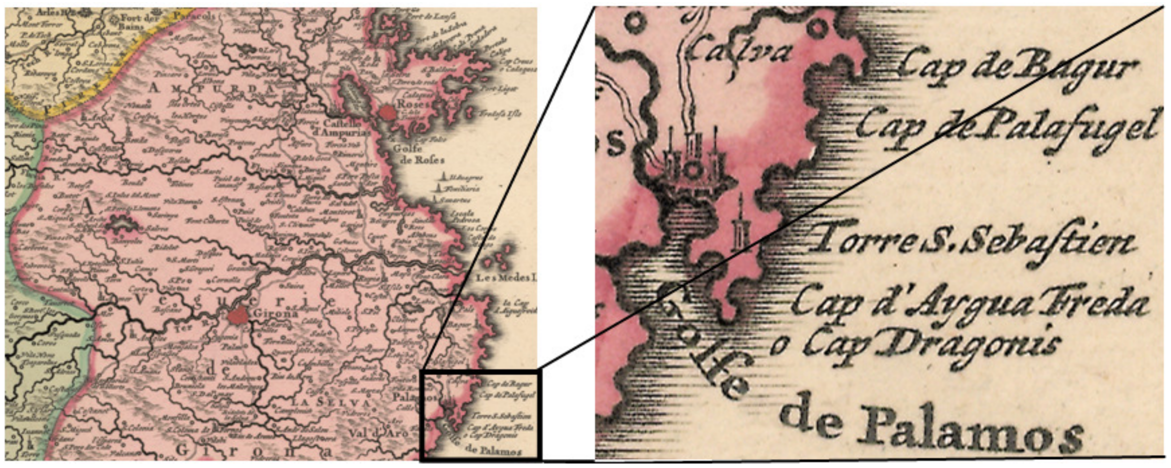
In maps, long graphical lines touch or overlap with text in some places. To extract all the text components from graphical documents, we need to remove long graphical objects that are present in the image as shown in Fig.1(c). In the literature [9, 16], different algorithms for extraction of these text characters are explained. Connected component analysis on the image will be useful for extracting isolated characters in the document. For each connected component, we use a minimum enclosing boundary box which describes the height and width of the character shape. The text components are separated analyzing the rectangular size (size of bounding box) and the area of the connected components. For this purpose histogram of the bounding box size of the components is generated. Through a correct threshold selection (*T*) 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 ×max(A_{mp}, A_{avg}), where A_{mp} and A_{avg} are the mode and average of the bounding box size. The value of 'n' was set to 3 empirically according to Tombre et al. [9]. This resultant image is fed to our system for line segmentation.



(a)



(b)



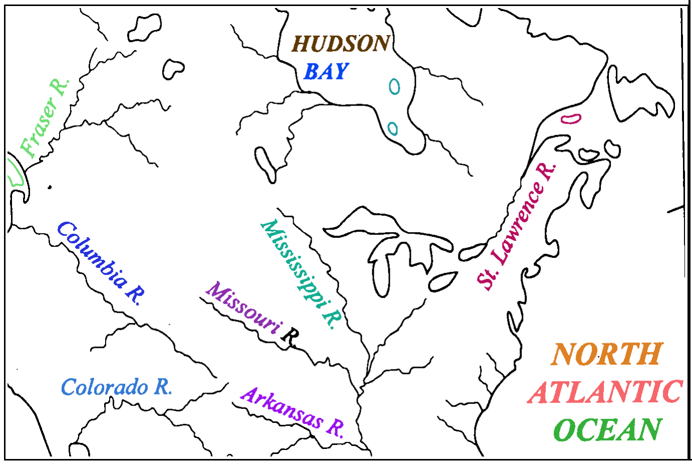
(c)

Fig.13. (a) and (b) Synthetic Maps of the same graphical background but different foreground. (c) A real Historical Map showing its curvilinear text lines in a zoomed window.

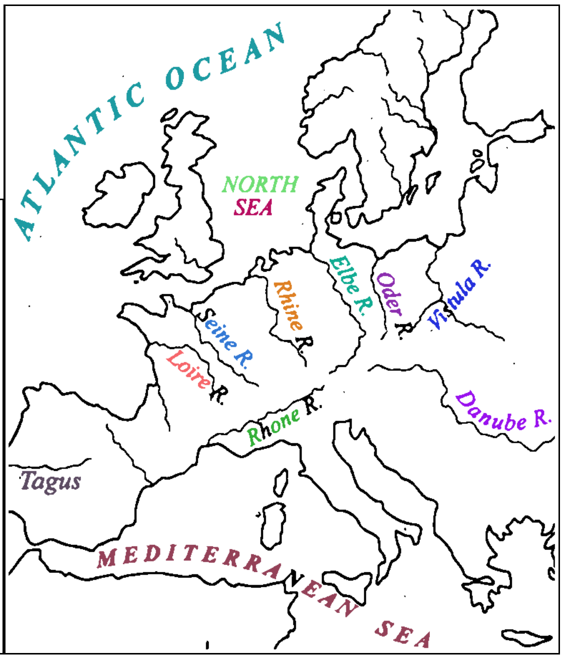
As mentioned before, there may exist some small graphical objects which are not text components. Also, some text components which are touched by graphical objects may be removed by the size threshold criterion. Due to this reason, we may not extract all the text characters by the text/graphics separation approach mentioned earlier. The text characters which touch graphical lines are missed out from the text lines extracted in the final result. To get an idea of our text line extraction results in graphical documents, some of the images are shown in Fig.14. Most of the errors in maps occurred due to missing of character components in their corresponding lines. It is noticed that, if some characters are in middle of a word, then our algorithm can extract the rest of the characters and group them in a single line. For example in Fig.14(b) the word "MEDITERRA_EAN SEA" is recovered though some of its characters like `N' in the middle are missing. We can use this extracted text line information to obtain the missing characters. Using some post-processing steps such as top-bottom line fitting of the text-line, the missing characters of the words could be found. We get some false positives in maps due to the presence of small components (non-text) in a line. In the engineering drawing dataset, the errors occurred when words were located very close together. As we are not considering the graphical information which is separated from the text portion, the boundary/box information of the text is ignored. Due to this reason, the words from different text blocks are combined together.

4.3 Seal Documents

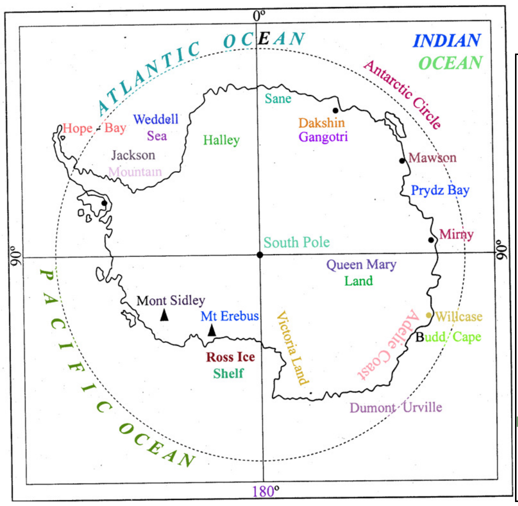
For experiments with real noisy data, documents containing seals were considered (see Fig.15). These were collected from different sources, for example historical documents, postal letters, official documents from universities etc. In these documents, text lines are in different orientations and they overlap with seal many times. We tested a database of 50 documents containing seals of English characters. The documents were digitized by a flatbed scanner at 200 dpi. In our database there are mainly 3 types of seal shapes, e.g. circular, elliptical and rectangular. Text information is annotated in straight or curve way according to the shapes of seals. The seals are posted in different orientations and in different locations. Frequently, there is missing of seal information due to noise or overlapping signatures. Sometimes, due to stamping on to text documents, many additional text characters of the document itself are also present inside the seal region.



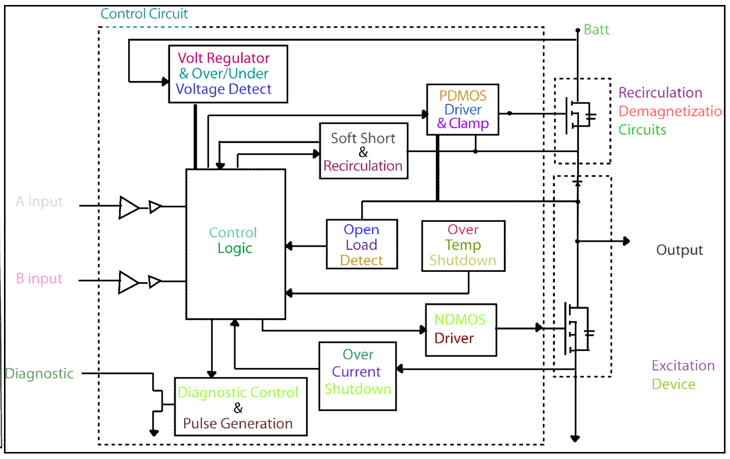
(a)



(b)



(c)



(d)

Fig.14. (a), (b), (c) Text line results from three different maps, (d) Engineering drawing are shown.

The characters in a string may touch frequently due to background noise, which creates problems for the extraction of isolated text characters. When more than one characters in a string touch, they make a touching component. The size of a touching component is usually bigger than an isolated component. As our text line segmentation approach is performed by clustering the character components based on their size and positional information, touching components may not be clustered in the text line because of the size difference of touching components and isolated characters. To tackle this problem, the touching components are decided first using the number of valley formed in these components. For the touching components, the orientations of each extreme side are used for character clustering. To decide a component as touching or isolated, we checked the number of concavity regions of each component. If the number of concave regions in a single component is more than 4, then we assume that the component is a touching component. Here, the height information (height of minimum enclosing rectangle) is considered for checking size similarity criteria mentioned in Section 3.1 for character clustering.

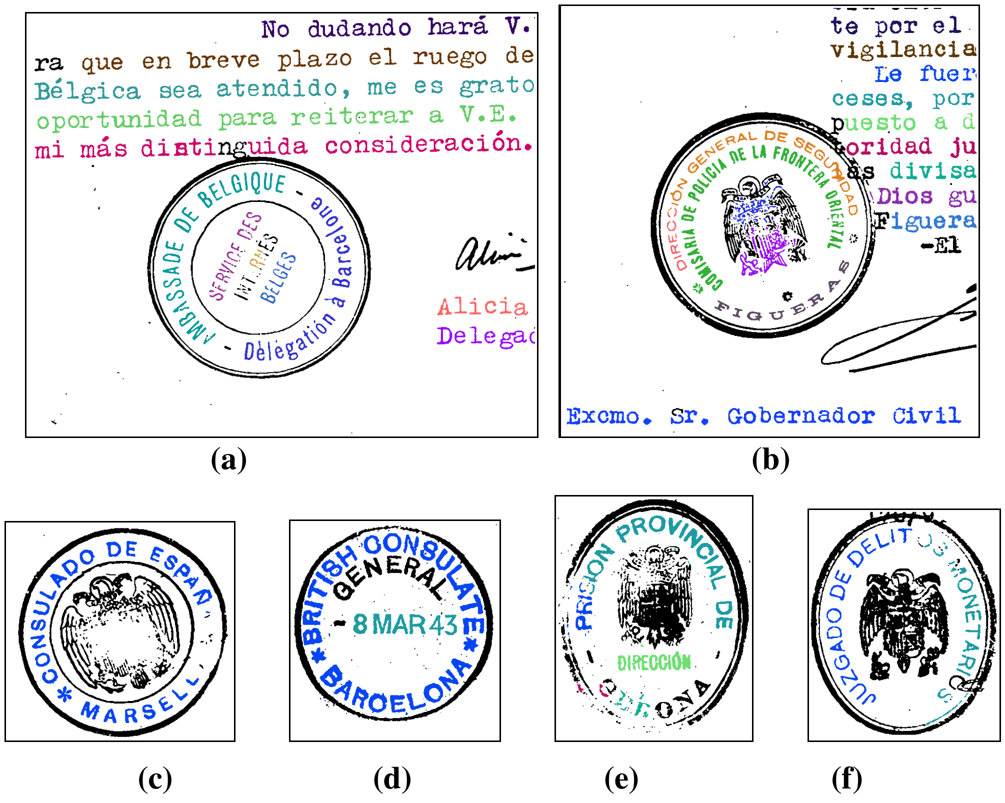
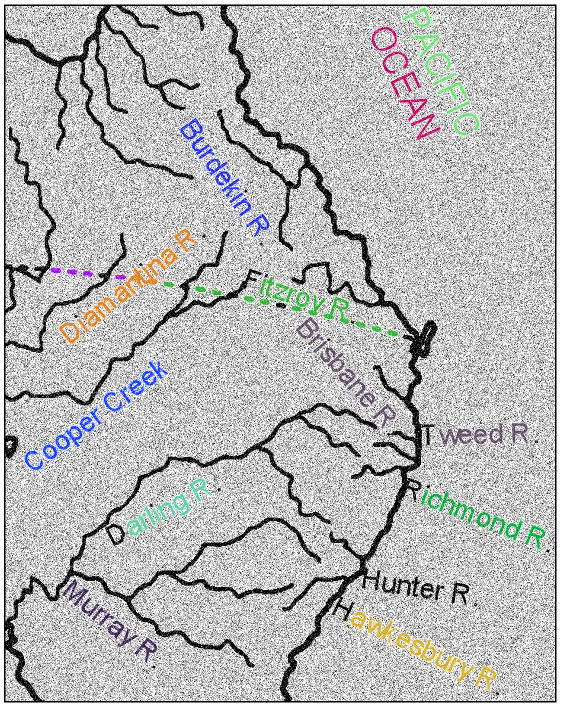
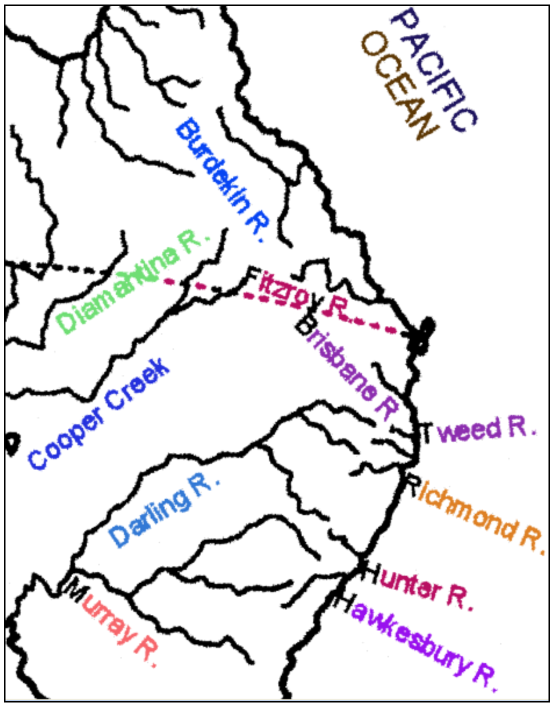

No dudando hará V.
ra que en breve plazo el ruego de
Bélgica sea atendido, me es grato
oportunidad para reiterar a V.E.
mi más distinguida consideración.
AMBASSADE DE BELGIQUE - Délégation à Barcelone -
SERVICE DES INTÉRÊTS BELGES
Alicia
Delegad
te por el
vigilancia
Le fuer
ceses, por
puesto a d
oridad ju
más divisa
Dios gu
Figuera
-El
DIRECCIÓN GENERAL DE SEGURIDAD
COMISARIA DE POLICIA DE LA FRONTERA ORIENTAL
FIGUERAS
Excmo. Sr. Gobernador Civil
(a)
(b)
CONSULADO DE ESPAÑA * MARSELLA
BRITISH CONSULATE GENERAL * BARCELONA *
8 MAR 43
PRISION PROVINCIAL DE GERONA
DIRECCIÓN
JUZGADO DE DELITOS MONETARIOS
(c)
(d)
(e)
(f)

Fig.15. (a)-(f) Line segmentation results obtained from different seal documents.

4.4 Results with synthetic noise

We have tested our approach with the documents added with synthetic noises. A set of graphical documents are degraded with Gaussian noise of different noise levels (10%, 20% and 30%). For lines segmentation in noisy documents, our approach is as follows. In the binarized image we apply a Gaussian smoothing technique to remove some of the noises. Since our method is based on cavity, so if there is a broken part, our method may not work. Hence we used a method to join the broken parts using the algorithm due to Roy et al. [27]. If there is a small broken part in the component this algorithm can join the broken part and our proposed method work well. Next we use our approach for line segmentation. To get an idea of such line segmentation results, we show a document image in Fig.16(a) where the document is added with 20% Gaussian noise. We have also tested method with Kanungo noise with parameter (0.9, 0.9, 0.5, 0.2, 0.0, 1.0) and result is provided in Fig.16(b).





(a) **(b)**

Fig.16. Line segmentation results are shown on a document after addition of (a) Gaussian noise and (b) Kanungo Noise.

4.5 Discussion and Error Analysis

To give an idea about different ranges of accuracy of the system on different types of documents, we divide the accuracy into three categories: (a) 100% (b) 95-99.9% and (c) <95%. The accuracy of the 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 $(M \times 100)/N\%$. So if all components of a text line are extracted correctly by the proposed algorithm, we say that the accuracy for the line is 100%.

Fig.17 shows the overall result of line segmentation using our approach. To evaluate the performance of the system with the retrieved text lines, we use common ratio of precision (P) and recall (R). The precision measures the quality of the extracted lines in terms of the ability of the system to include only relevant text lines in the result. For a given retrieval result, the precision measure (P) is defined as the ratio between the number of relevant extracted lines and the number of retrieved lines. Whereas recall measures the effectiveness of the system in retrieving the relevant text lines. The recall (R) is defined as the ratio between number of relevant retrieved lines to the total number of relevant lines in the collection. Precision and recall are computed as follows.

$$\text{Precision} = \{\text{relevant line}\} \cap \{\text{retrieved line}\}/\{\text{retrieved line}\}$$

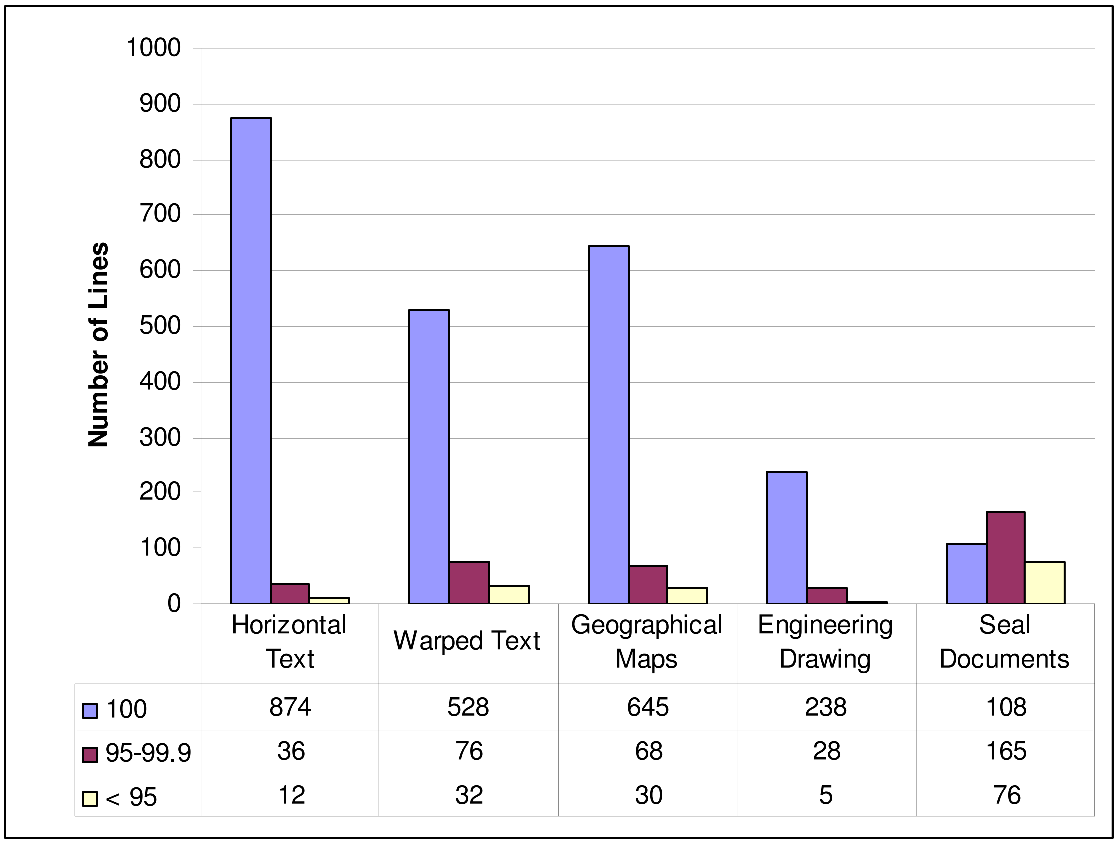
$$\text{Recall} = \{\text{relevant line}\} \cap \{\text{retrieved line}\}/\{\text{relevant line}\}$$

We show in Table 1 the precision recall accuracy of different datasets obtained from our experiments. The precision recall performance is computed with the text lines where components in text lines are more than 95%.

Table 1. Precision Recall measure of our approach obtained in different datasets.

	Horizontal Text	Warped Text	Geographical Maps	Engineering Drawing	Seal Documents
Precision	99.12%	97.42%	98.48%	95.0%	88.93%
Recall	98.70%	94.97%	95.96%	98.15%	78.22%

In our proposed methodology of text line separation, the assumption was taken that the text is present in foreground layer of the documents. Thus, the images were binarized by Otsu threshold to convert the image into two-tone images. If the image is in reverse contrast, i.e. black background and white foreground, our present method can not handle. Our method, thus, depends on efficiency of binarization approach. Also, our system will not work well with color information in the document. We have converted color images to binary images using a typical color to gray image conversion technique and used the binary images for different experiments. More focused color segmentation methods, precisely for documents can be useful for better results.



	Horizontal Text	Warped Text	Geographical Maps	Engineering Drawing	Seal Documents
🟦 100	874	528	645	238	108
🟥 95-99.9	36	76	68	28	165
🟨 < 95	12	32	30	5	76

Fig.17. Text line segmentation accuracy of our proposed method from different documents. Y-axis represents the number of lines and X-axis represents the accuracy.

Most of the errors in our approach are due to over-segmentation of the characters in the text line. We show in Fig.17, the text line segmentation accuracy in different dataset used in our experiment. In the datasets of normal scanned text documents, the text lines are segmented very well. In CBDAR dataset, the warped lines are segmented properly and separated from graphical portions if they exist. The orientation of text line information can be useful in future for de-warping the document. In seal document dataset, text line segmentation accuracy is affected mainly due to broken text components in the seal and the presence of long graphical lines over text regions. Also, the text lines in these documents are not separated by large inter-line distance. Sometimes, more than one character clusters appear in the candidate region. Thus, error occurs when a cluster end is very close to another cluster, which is a member of a different text line. Hence, text lines are over segmented. From the quantitative analysis, we noted that only 1.12% errors occur because of such situation. To reduce such errors, we have considered height and orientation information. In maps, when the characters of a word are distantly spaced, then our line extraction method sometimes fails to join the cluster ends due to threshold selection. In future we plan to work to take care of such errors.

Instead of the few drawbacks that occur mainly from connected components detection, our line segmentation approach can handle documents where the characters are sparse and dense. As the algorithm depends on the background information for growing of the text lines, the method is invariant to character font and style. Also, another advantage of the method is that, it does not depend on the orientation of the text lines. We have tested it in graphical documents containing maps, electrical diagrams, etc. The extracted text line can be useful for post-processing to find missing/touching characters in graphical documents. We have demonstrated the robustness of the approach with a range of documents from degraded text documents to complicated seal document.

5. Conclusion

We have presented a new approach for text line segmentation. A key characteristic of our approach is the use of foreground and background information. The background information is taken care using Water Reservoir approach, which guides our process to follow the text line. The approach is based on perceptual laws of text characters in a line and it is easy to apply.

We have demonstrated our approach of invariance towards rotation, scale, affine to real world images. We demonstrated the robustness of this approach with different dataset with high degree of curvilinearity. For this purpose, we considered camera based documents, graphical documents (map, engineering drawing) and seal documents. Our experiment shows that the approach can be used without de-warping or rotating the documents. One of the significant advantages of the proposed method is its flexibility. Our scheme is independent of font, size and style of the text characters.

References

[1] H. Goto and H. Aso, "Extracting curved lines using local linearity of the text line", International Conference on Document Analysis and Recognition (IJDAR), vol.2 pp. 111-118, 1999.

[2] G. Nagy, S. Seth and M. Viswanathan, "A prototype document image analysis system for technical journals", Computer, vol.25, pp. 10-22, 1992.

[3] F. Hones and J. Lichter, "Layout extraction of mixed mode documents", Machine vision and applications, vol.7, pp.237-246, 1994.

[4] L. O'Gorman, "The document spectrum for page layout analysis", Transactions on Pattern Analysis and Machine Intelligence, vol. 15, pp. 1162-1173, 1993.

[5] U. Pal and P. P. Roy, "Multi-oriented and curved text lines extraction from Indian documents", IEEE Trans. on SMC - Part B, vol.34, pp.1676-1684, 2004.

[6] P.K.Loo and C.L.Tan, "Word and sentence extraction using irregular pyramid", Workshop on Document Analysis System (DAS), pp. 307-318, 2002.

[7] U. Pal, A. Belaïd and Ch. Choisy "Touching numeral segmentation using water reservoir concept" Pattern Recognition Letters, vol.24, pp. 261-272, 2003.

[8] P. P. Roy, U. Pal and J. Lladós, "Touching Text Character Localization in Graphical Documents Using SIFT", Revised Selected Papers of Workshop on Graphics Recognition (GREC), Lecture Notes in Computer Science (LNCS), pp. 199-211, 2010.

[9] K. Tombre, S. Tabbone, L. Peissier, B. Lamiroy, and P. Dosch. "Text /graphics separation revisited", Workshop on Document Analysis System (DAS), LNCS, Springer-Verlag, London, UK, vol. 2423, pp. 200-211, 2002.

[10] F. Kimura and M. Shridhar, "Handwritten numeral recognition based on multiple algorithms", Pattern Recognition, vol. 24, pp. 969-983, 1991.

[11] U. Pal, S. Sinha and B. B. Chaudhuri, "English Multi-Oriented Text Line Extraction", Image Analysis, Springer Verlag, Lecture Notes on Computer Science (LNCS-2749), pp. 1146-1153, 2003.

[12] B. Gatos, I. Pratikakis, and K. Ntirogiannis. Segmentation based recovery of arbitrarily warped document images. In Proceedings of International Conference on Document Analysis and Recognition (ICDAR), 2007.

[13] L. Likforman, A. Zahour and B. Taconet. "Text Line Segmentation of Historical Documents: a Survey". International Journal on Document Analysis and Recognition (IJDAR). pp. 123 – 138, 2007.

[14] F. Shafait and T. M. Breuel. "Document image dewarping contest". In 2nd international Workshop on Camera-Based Document Analysis and Recognition (CBDAR). Curitiba, Brazil. Sep. 2007.

[15] OpenCV library : http://sourceforge.net/projects/opencvlibrary/

[16] L.A. Fletcher and R. Kasturi. "A Robust Algorithm for Text String Separation from Mixed Text/Graphics Images". IEEE Transactions on Pattern Analysis and Machine Intelligence, vol (10), pp. 910-918 , 1988.

[17] H.C. Park, S.Y. Ok, Y.J. Yu and H.G. Cho. "Word Extraction in Text/Graphic Mixed Image Using 3-Dimensional Graph Model". International Journal on Document Analysis and Recognition (IJDAR), vol (4), pp. 115 130 , 2001.

[18] C.L. Tan and P.O. Ng. "Text extraction using pyramid". Pattern Recognition, vol (31), pp. 63-72 , 1998.

[19] S. He, N. Abe and C. L. Tan. "A clustering-based approach to the separation of text strings from mixed text/graphics documents". International Conference on Pattern Recognition (ICPR), pp. 706-710, 1996.

[20] M. Burge and G. Monagan. "Extracting Words and Multi Part Symbols in Graphics Rich Documents". International Conference on Image Analysis and Processing (ICIAP), 1995.

[21] M. Deseilligny, H. Le Men and G. Stamon. "Characters string recognition on maps, a method for high level reconstruction". International Conference on Document Analysis and Recognition (ICDAR), pp. 249-252, 1995.

[22] M. Delalandre, T. Pridmore, E. Valveny, E. Trupin and H. Locteau. "Building Synthetic Graphical Documents for Performance Evaluation". Workshop on Graphics Recognition (GREC) , Lecture Note in Computer Science (LNCS), vol (5046), pp. 288-298 , 2008.

[23] S. S. Bukhari, F. Shafait, and T. M. Breuel. "Segmentation of curled textlines using active contours". In 8th IAPR Workshop on Document Analysis Systems (DAS), pp. 270–277, 2008.

[24] Y. Li, Y. Zheng, D. Doermann, and S. Jaeger. "Script independent text line segmentation in freestyle handwritten documents". IEEE Transactions on Pattern Analysis and Machine Intelligence, 30(8):1313–1329, 2008.

[25] A. Clavelli and D. Karatzas. "Text segmentation in colour posters from the Spanish civil war era". International Conference on Document Analysis and Recognition (ICDAR), pp. 181-185, Spain, 2009.

[26] N. N. Bai, K. Nam and Y. Song. "Extracting curved text lines using the chain composition and the expanded grouping method". Document Recognition and Retrieval XV, USA, 2008.

[27] K. Roy, U. Pal and B. B. Chaudhuri, "A System for Joining and Recognition of Broken Bangla Numerals for Indian Postal Automation". In Proc. of 4th Indian Conf. on Computer Vision, Graphics and Image Processing, pp 641-646, 2004.

[28] N. Otsu, "A threshold selection method from grey level histogram". IEEE Trans on SMC, vol (9), pp.62-66, 1979.