Language-Independent Text Lines Extraction Using Seam Carving

Raid Saabni[1,2]
[2]*Triangle R&D Center,*
Kafr Qara, Israel
saabni@cs.bgu.ac.il

Jihad El-Sana[1]
[1]*Department of Computer Science,*
Ben-Gurion University of the Negev
Beer Sheva,Israel
el-sana@cs.bgu.ac.il

***Abstract*—In this paper, we present a novel language-independent algorithm for extracting text-lines from handwritten document images. Our algorithm is based on the seam carving approach for content aware image resizing. We adopted the signed distance transform to generate the energy map, where extreme points indicate the layout of text-lines. Dynamic programming is then used to compute the minimum energy left-to-right paths (seams), which pass along the "middle" of the text-lines. Each path intersects a set of components, which determine the extracted text-line and estimate its hight. The estimated hight determines the text-line's region, which guides splitting touching components among consecutive lines. Unassigned components that fall within the region of a text-line are added to the components list of the line. The components between two consecutive lines are processed when the two lines are extracted and assigned to the closest text-line, based on the attributes of extracted lines, the sizes and positions of components. Our experimental results on Arabic, Chinese, and English historical documents show that our approach manage to separate multi-skew text blocks into lines at high success rates.**

***Keywords*-Seam Carving; Line Extraction; Multilingual; Signed Distance Transform; Dynamic programming; Handwriting.**

I. Introduction

The large collections of handwritten historical manuscripts existing in libraries, museums, and private houses around the world are valuable human heritage. The rising interest in these collections and the recent effort to digitize them reveal interesting problems, which call for theoretical and applied research in Historical Document Image Analysis. These include document image binarization, writer identification, page layout analysis, keyword searching, indexing, and script recognition. These procedures are essential in helping scholars easily access and analyze digital copies of historical documents. However, the low quality of these document images, the lack of constraints on page layout, and the complexity of handwriting, pose real challenges for processing such document images automatically. The success in optical character recognition techniques on printed scripts cannot be easily carried to handwritten documents. Nevertheless, modest tasks, such as keyword searching and spotting, are already in use for average quality documents.

Text line extraction algorithms aim to determine the letters and words along a text line on an image document. It is an important practice often used in various handwriting analysis procedures, such as word-spotting, keyword searching, and script recognition. For example, keyword searching often requires determining the text-lines. In addition, segmenting text blocks into distinct text lines is vital for script recognition [17], [16], [15], [5], [11]. It is easy to determine the text lines in machine-printed documents, as they are usually parallel and often have the same skew. Density histograms, projection profiles, and Hough transform are often enough to reveal text lines in machine-printed document images. On the contrary, determining the text lines in handwritten historical documents is a challenging task for various reasons: the variability of skew between the different text-lines and within the same text-line; spaces between lines are narrow and variable; components may spread over multiple lines or overlap and touch within consecutive lines; and the existence of small components, such as dots and diacritics (e.g., Arabic script), between consecutive text-lines.

Several text-line extraction methods for handwritten documents have been presented. Most of them group connected components using various distance metrics, heuristics, and adaptive learning rules. *Projection profile*, which was initially used to determine text-lines in printed image documents, was modified and adapted to work on sub-blocks and stripes [21], [13], [22], [9].

In this paper we present a language independent global method for automatic text line extraction. The proposed algorithm computes an energy map of the input text block image and determines the seams that pass across text lines. The crossing seam of a line, l, marks the components that make the letters and words along l. These seams may not intersect all the components along the text line, especially vertically disconnected components; e.g., a seam may intersects the body of the letter *"i"* and misses the dot. This is handled by locally labeling and grouping the components that formulate the same letter, word-part, or word. The component collection procedure may require parameter adjustments that may differ slightly from one language to the other, and mainly depend on the existence of additional strokes – their expected location and size.

In the rest of this paper we briefly review related works,

1520-5363/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDAR.2011.119

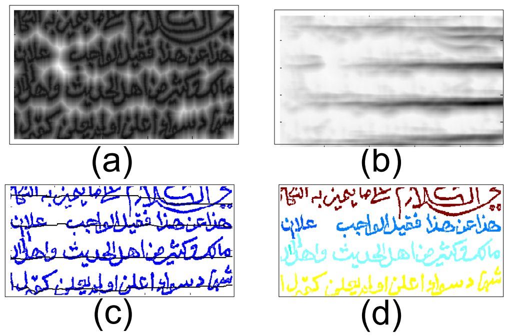

Figure 1. (a) Calculating a Signed Distance Map of a given binary image, (b) Calculating the energy map of all different seams, (c) Finding the seam with minimal energy cost and (d) Extracting the components that intersect the minimal energy seam.

describe our approach in detail, and present some experimental results. Finally we conclude our work and discuss directions for future work.

II. Related Work

Text-line extraction methods can be divided roughly into three classes: top-down, bottom-up, and hybrid. Top down approaches partitions the document images into regions, often recursively, based on various global aspects of the input image. Bottom-up approaches group basic elements, such as pixels or connected components, to forms the words of a line. The hybrid schemes combine top-down and bottom up procedures to yield better results.

A. Top-down approaches

Projection Profiles [21] along a predetermined direction are usually used in top-down approaches to determine the paths separating consecutive text-lines. Shapiro *et al.* [22], applied a Hough transform to determine the predefined direction to compute the *Project Profile*. Hough transform was also used by Likforman-Sulem *et al.* [12] to generate the best text line hypothesis in the Hough domain and check the validity of the hypothesis in the image domain. He and Downton [9] presented the *RXY* cuts, which relies on projections along the X and the Y axes, resulting in a hierarchical tree structure. Several approaches [3], [9], [25] use *Projection Profile* on predefined sub blocks of the given document image to handle multi-skew. These global methods often fail to segment multi-skew (fluctuating) document images. An Adaptive Local Connectivity Map (ALCM) was presented in [26] for text-line location and extraction, which can be directly applied on gray-scale images. Thresholding the gray scale(ALCM), reveals clear text-line patterns as connected components. Shi *et al.* [18] presented a text-line extraction method for handwritten documents based on ALCM. They generate ALCM using a steerable direction filter and group connected components into location masks for each text line, which are used to collect the corresponding components (on the original binary document image). Nicolaou and Gatos [14] used local minima tracers, to follow the white-most and black-most paths from one side to other in order to shred the image into text line areas. Alaei *et al.* [1]. presented a novel approach for unconstrained handwritten text-line segmentation addressing different languages such as Persian, German, Greek, English and others. They use a new painting technique to smear the foreground portion of the document image and by that enhances the separability between the foreground and background. Furthermore, the authors presented some novel techniques to overcome the problems of overlapping and touching components.

B. Bottom-up approaches

Various approaches rely on grouping techniques to determine text line in document images, while applying applying heuristic rules [6], learning algorithms [24], nearest neighbor [8], and searching trees [20]. In contrast to the machine printed document images, simple rules such as nearest neighbor does not work for handwritten documents. The nearest neighbor often belongs to the next or previous text line, which necessities additional rules for quality measurement to determine the quality of the extracted text lines. The approaches on this category require the isolation of basic building elements, such as strokes and connected components, and often find it difficult to separate touching component across consecrative text rows

Gorman [8] presented a typical grouping method, which rules are based on the geometric relationship among k-nearest neighbors. Kise *et al.* [10] combine heuristic rules and the Voronoi diagrams to merge connected components into text lines. Nicola *et al.* [20] use the artificial intelligence concept of the production system to search for an optimal alignment of connected components into text lines. The minimal spanning tree (MST) clustering technique was used in [19] to group components to text lines. Proximity, similarity, and direction continuity were used to iteratively construct lines by grouping neighboring connected components [6]. Recently, a few methods were presented using Level-set techniques for line extraction [23], [4].

III. Our Approach

Human ability to separate text blocks into lines is almost language independent. They tend to identify lines by collecting basic elements and/or components into groups and then analyze the shape, size, and location of these groups with respect to the adjacent elements. The spaces between adjacent lines and the concentration of ink along the lines play a major role in separating text-lines. These observations have motivated most line extraction approaches to search for the path that separates consecutive text-lines with minimal crosses and maximal distance from the script components.

Our novel approach to separating text blocks into lines was inspired and built upon the seam carving work [2], which resizes images in a content-aware fashion. The core of our approach is a line extraction procedure (See Figure 1), which starts by computing an energy map of the input image based on the signed distance transform (Figure 1(a)). It then uses dynamic programming to compute the minimum energy path, p, that crosses the image from one side to its opposite, as shown in Figure 1(b,c). The path p resembles a text line in the document image. Finally, it collects the components along the computed path p, which formulate the words of that line (Figure 1(d)). The line extraction procedure is executed iteratively until no more lines remain. In our current implementation we assume the input document image is binary.

Next we discuss in detail the three main steps of the line extraction procedure: generating an energy map, computing the minimal energy path, and collecting the component along the path.

A. Preprocessing

It is possible to detect over-average-height components before segmenting the text into lines, but determining a component that vertically stretches over multi-lines requires line estimation and extraction.

We calculate the average height of the connected components and classify them (according to their height) into four categories: additional strokes, ordinary average components, large connected components, and vertically touching components. Additional strokes are identified as the small components; components that include ascenders and/or descenders are classified as large components; and the components which are significantly higher than ordinary and large connected components are classified as touching components.

The classification is performed by comparing to the average height of the components. The classification is not rigid, i.e., components may switch category after line extraction. In the preprocessing step, connected components, which were labeled as touching components, are split vertically in the middle. The list of these components is passed to the post-processing phase, which draws the final decision based on the extracted lines – a suspected touching component may actually be an ordinary large component with ascender/descender. The small components (additional strokes) are reconsidered with respect to the computed line region to decide their final position.

B. Energy function

Avidan and Shamir [2] discussed several operators for calculating the energy function to determine pixels with minimum energy for content-aware resizing. They suggested the gradient operator (see Equation 1) to generate the energy map, $E(I)$, for an image I and showed that removing pixels with low energy lead to minimal information loss.

$$E(I) = \left| \frac{\partial(I)}{\partial x} + \frac{\partial(I)}{\partial y} \right| \tag{1}$$

Typical line extraction approaches seek paths that separate text lines in a document image, which is usually performed by traversing the "white" regions between the lines or the medial axis of the text (the "black" regions), respectively. The separating paths are perceived as seams, in seam carving terminology, with respect to some energy function. We have found the energy functions presented in the seam carving work inappropriate for text line extraction, mainly because the applications are different.

The search for a separating path (polyline) that lies as far as possible from the document components motivated adopting the distance transform for computing the energy map. Local extreme (minima and maxima) points on the energy map determine the separating path. This scheme also requires maintaining a range of possible horizontal directions to prevent seams (paths) from jumping across consecutive lines. Even though, the seams often jump across consecutive lines, mainly when the local skew is close to diagonal or when there is large gaps between consecutive components on the same row. It also fails to handle touching components along consecutive lines, as they act as barriers and prevent the progress of the seam along the between-lines white region. To overcome these limitations we search for seams that pass along the medial axis of the text lines.

To search for seams that pass along the medial axis of the text lines, i.e., cross components within the same text line, we use the *Signed Distance Transform*[SDT] in computing the energy map. In SDT, pixels lying inside components have negative values and those lying outside have positive values. Following the local minima on that energy map generates seams that pass through components along the same text-line.

C. Seam Generation

We define a horizontal seam in a document image as a polyline that passes from the left side of the image to the right side. Formally, let I be an $N \times M$ image, we define a horizontal seam as shown in Equation 2, where x is the mapping, $x : [1 \ldots m] \rightarrow [1 \ldots n]$. For $k = 1$ the resulting seam is 8-connected and for $k > 1$ the resulting seam may not be connected. Note that seams in content-aware resizing are connected in order to maintain the uniform rectangle grid of the image, when removing the seam pixels.

$$S = \{x(i), i\}_{i=1}^{m}, \forall\ i, \mid x(i) - x(i-1) \mid \leq K. \tag{2}$$

Let $E(I)$ be the distance transform based energy map, the energy cost, $e(s)$, of a horizontal seam (path) s is defined by Equation 3. The minimal cost seam, s_{min}, is defined as the seam with the lowest cost; i.e., $s_{min} = \min_{\forall\ s}\{e(s)\}$.

$$e(S) = e(\{x(i), i\}_{i=1}^{m}) = \sum_{i=1}^{m} E(x(i)) \tag{3}$$

Dynamic programming is used to compute the minimal cost seam s_{min}. The algorithm starts with filling a 2D array, $SeamMap$, which has the same dimension as the input image document. It initializes the first column of the cell map, $SeamMap$, to the first column of the energy map image; i.e., $SeamMap[i, 1] = E(e)[i, 1]$. It then proceeds iteratively and computes the rest of the columns from left to right and top-down using Equation 4. Elements out of ranges of the array $SeamMap$, are excluded from the computation.

$$SeamMap[i, j] = \tag{4}$$
$$2E(i, j) + min_{l=-2}^{2}(SeamMap[i + l, j - 1])$$

The resulting array, $SeamMap$, describes the energy cost of the left-to-right paths, which start from the left side and ends at the right side of the image. The algorithm determines the minimal cost path by starting with the minimal cost on the last column and traversing the $SeamMap$ array backward – from right-to-left.

D. Component Collection

The computed minimal cost path (seam) intersects the main components along the medial axis of the text line, but may miss off-the-baseline small satellite components, which usually consist of delayed strokes and small components, such as dots and short strokes. In addition, touching components across consecutive lines are treated as one component and assigned to the first intersecting path. Our component collection algorithm manages to handle almost all these cases correctly.

For an input minimal energy seam $s = \{x(i), i\}_{i=1}^{m}$, the collection component algorithm performs three main steps. In the first step it defines an empty component list, c_l, it then determines the components that intersect the seam s and adds them to the component list, c_l. The components in c_l represent the text row, r_s, spanned by the seam s and used to determine the upper, u_r and lower, l_r, boundary of the text line. We refer to the region between the two boundaries as the *row region*. The mean and standard deviation of the height of a row region is measured and used to filter touching elements across consecrative lines. The over-sized vertical components – their height being significantly above the average height – are classified as touching components and split in the middle. Small satellite components that intersect the row region are handled in two different phases. Components which major fraction (above 50%) falls within a row region, are assigned to the text row r_s, spanned by the seam s (note that this also includes components that fall entirly within the row regions). Finally the row region is marked as processed region.

This procedure does not collect small components beyond the row region, since correct assignment requires the existence of the two bounding row regions. For this reason, for each computed seam (except the first one), we determine whether it is adjacent to a marked region (already processed row region). In such a case, we distribute the unclassified components between the two adjacent row regions based on their distance from adjacent row regions; i.e., each component is assigned to the closest row region.

IV. Experimental Results

Several evaluation methods for line extraction algorithms have been reported in the literature. Some, evaluate the results manually, while others use predefined line areas to count misclassified pixels. Connected components are also used to count the number of misclassified components within the extracted lines [18]. We also measure the classification of connected components to evaluate our algorithm. We have manually generated the ground truth for the test set images by adding information about the lines existing in these pages as groups of word-parts (main part and additional strokes). The results of the algorithm are evaluated by counting the number of the classified and misclassified components in the processed data sets.

We have evaluated our system using 40 Arabic pages from Juma'a Al-Majid Center (Dubai), ten pages in Chinese, and 50 pages taken from the ICDAR2007 Handwriting Segmentation Contest dataset [7] including English, French, German, and Greek. The images have been selected to have multi-skew and touching lines. The 40 Arabic pages include 853 lines and $24,876$ word-parts, the 50 pages from the ICDAR2007 contest have 967 lines. Using the Arabic set, Only 9 lines were extracted with misclassified components; i.e., 98.9% lines were extracted correctly. The number of misclassified word-parts (additional strokes in extreme cases were not considered) was 312, which is 1.25% of the $24,876$ word-parts. In a post-processing step, we have used average height, orientation, and average component size (along text line) to reclassify components. Around 63% of misclassified components were reclassified correctly. All the 86 touching components from consecutive lines in the tested dataset were split correctly in the post processing step, see Figures 3 and 4 . We have received similar results with the other 50 pages from the ICDAR2007 contest. Only 12 lines were extracted incorrectly which is 98%. Generally, misclassification occurs when the extracted seam jumps from one line to its neighbor, which can be easily detected during line extraction and corrected at a post processing level.

We have implemented two known systems to evaluate the performance of our algorithm. The first system was presented by Nicolaou and Gatos [14], shreds an image to lines using tracers to follow the white-most and black-most

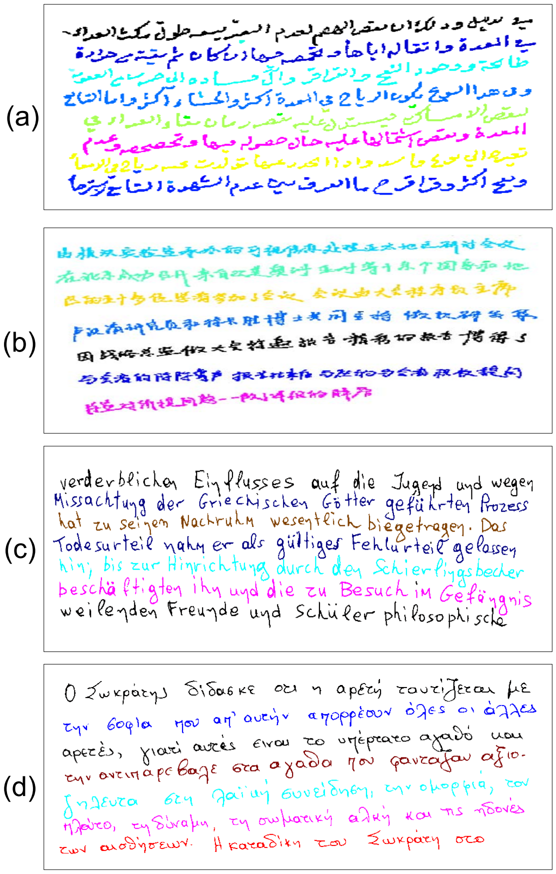

Figure 2. Random samples from the tested pages: (a) Arabic, (b) Chinese, (c) German and (d) Greek. The extracted lines are shown in different colors.

	Nicolaou09	Shi09	Our System
ICDAR2007	98.1%	98.2%	98.6%
Dubai+	97.85%	98.1%	98.9%

Table I

RESULTS ACHIEVED BY OUR SYSTEM (THE THIRD COLUMN) COMPARED TO THE SYSTEM PRESENTED IN [14](THE FIRST COLUMN) AND THE SYSTEM PRESENTED IN [18](THE SECOND COLUMN). THE FIRST ROW COMPARES RESULTS OF THE THREE SYSTEMS ON A SUBSET OF THE ICDAR2007 [7] TEST SET AND THE SECOND ROW USES OUR PRIVATE COLLECTION, WHICH IS BASED ON DOCUMENTS FROM JUMA'A ALMAJED CENTER, DUBAI.

paths. In the second system by Shi *et al.* [18], generates ALCM using a steerable direction filter and groups connected components into location masks for each text line. The three systems (including our system) were tested using the data sets described in this section and have yielded similar results, see table I.

V. Conclusion and Future Work

The scope of future work includes a hybrid scheme that utilizes the med-text-row seams and the between-row seams. We believe that such an approach can leverage the advantages of the two types of seams. Currently, the signed distance map is recomputed after each line extraction. Extracting lines from slices with adaptive orientation could save the recalculation of the signed distance map and improve the runtime performance of the algorithm. Extending the proposed approach to work directly on gray level images is an additional direction. In this case, a hybrid scheme can be used to find separators between lines and by that determine the area of each text line.

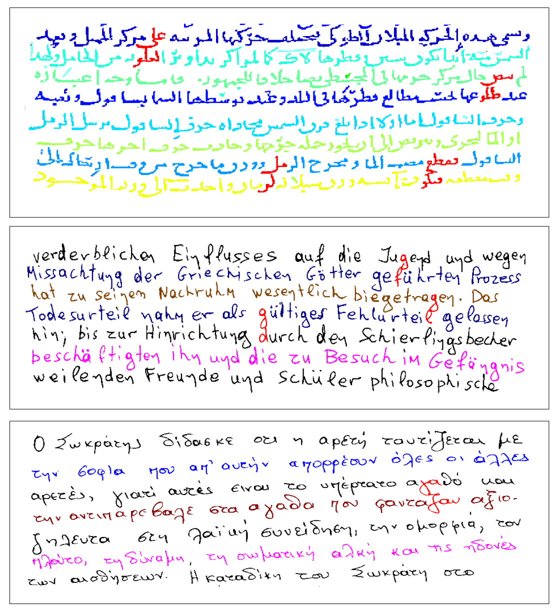

Figure 3. Documents from different languages with fluctuating lines, the components in red are touching components determined during the line extraction procedure.

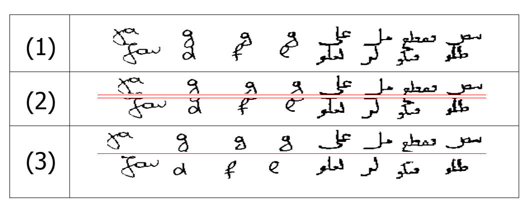

Figure 4. We can see in this figure the original touching component in line (1), the primary splitting in line (2), while in line (3) we can see the desired results.

VI. Acknowledgments

This research was supported in part by The Israel Science Foundation grant no. 1266/09 and DFG-Trilateral Grant no. 8716. We would like to thank the reviewers for their insightful comments which led to several improvements in the presentation of this paper.

References

[1] A. Alaei, U. Pal and P. Nagabhushan, "A new scheme for unconstrained handwritten text-line segmentation", *Pattern Recogn.*, 44:917–928, April 2011.

[2] S. Avidan and A. Shamir, "Seam carving for content-aware image resizing", *ACM Trans. Graph.*, 26(3):10, 2007.

[3] E. Bruzzone and M. Coffetti, "An algorithm for extracting cursive text lines", *in ICDAR 99: Proceedings of the Fifth International Conference on Document Analysis and Recognition. IEEE Computer Society*, 1999, pp 749.

[4] S. S. Bukhari, F. Shafait and T. M. Breuel, "Script-Independent Handwritten Textlines Segmentation Using Active Contours", *ICDAR*, 2009, pp 446–450.

[5] F. Chen, L. Wilcox and D. Bloomberg, "Word spotting in scanned images using hidden Markov models", *Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on*, 27-30 April 1993, volume 5, pp 1–4vol.5.

[6] L. forman Sulem and C.Faure, "Extracting text lines in handwritten documents by perceptual grouping", *in Advances in handwriting and drawing:a multidisciplinary approach .Winter Eds, Europia,Paris*, pp 117135, 1994.

[7] B. Gatos, A. Antonacopoulos and N. Stamatopoulos"ICDAR2007 Handwriting Segmentation Contest"", Proceedings of the 9th International Conference on Document Analysis and Recognition (ICDAR'07), Curitiba, Brazil, pp:1284-1288, September 2007.

[8] L. Gorman, "The document spectrum for pagelay-out analysis", *IEEE Trans. Pattern Analysis and Machine Intelligence.*, 15(11):11621173,, 1993.

[9] J. He and A. C. Downton, "User-Assisted Archive Document Image Analysis for Digital Library Construction", *ICDAR '03: Proceedings of the Seventh International Conference on Document Analysis and Recognition*, 2003, pp 498, Washington, DC, USA. IEEE Computer Society.

[10] K. Koichi, S. Akinori and I. Motoi, "Segmentation of page images using the area Voronoi diagram", *Comput. Vis. Image Underst.*, 70(3):370–382, 1998.

[11] A. Kolcz, J. Alspector, M. Augusteijn, R. Carlson and G. V. Popescu, "A Line-Oriented Approach to Word Spotting in Handwritten Documents", *Pattern Analysis and Applications*, 3(2):153 – 168, june 2000.

[12] L. Likforman-Sulem, A. Hanimyan and C. Faure, "A Hough based algorithm for extracting text lines in handwritten documents", *Document Analysis and Recognition, International Conference on*, 2:774, 1995.

[13] S. Nagy and S. Stoddard, "Document analysis with expert system", *Procedings of Pattern Recognition conference in practice II*, 1985.

[14] A. Nicolaou and B. Gatos, "Handwritten Text Line Segmentation by Shredding Text into its Lines", *ICDAR '09: Proceedings of the 2009 10th International Conference on Document Analysis and Recognition*, 2009, pp 626–630, Washington, DC, USA. IEEE Computer Society.

[15] T. Rath and R. Manmatha, "Word image matching using dynamic time warping", *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, 18-20 June 2003, volume 2, pp II–521–II–527vol.2.

[16] T. M. Rath, R. Manmatha and V. Lavrenko, "A search engine for historical manuscript images", *Annual ACM Conference on Research and Development in Information Retrieval*, pp 369–376, 2004.

[17] C. H. S. N. Srihari and H. Srinivasan, "A Search Engine for Handwritten Documents", *Document Recognition and Retrieval XII, San Jose, CA,Society of Photo Instrumentation Engineers (SPIE)*, pp pp. 66–75, January 2005.

[18] Z. Shi, S. Setlur and V. Govindaraju, "A Steerable Directional Local Profile Technique for Extraction of Handwritten Arabic Text Lines", *Document Analysis and Recognition, International Conference on*, 0:176–180, 2009.

[19] A. Simon, J.-C. Pret and A. P. Johnson, "A Fast Algorithm for Bottom-Up Document Layout Analysis", *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(3):273–277, 1997.

[20] S.Nicolas, T.Paquet and L.Heutte, "Text line segmentation in handwritten document using a production system", *in IWFHR04: Proceedings of the Ninth International Workshop on Frontiers in Handwriting Recognition (IWFHR04)*, 2004, pp 245 250.

[21] T.Pavlidis and J.Zhou, "Page segmentation by white streams", *1st Int. Conf. Document Analysis and Recognition. (ICDAR) Int. Assoc. Pattern Recognition*, 1991, pp 945–953.

[22] S. Vladimir, G. Georgi and S. Vassil, "Handwritten document image segmentation and analysis", *Pattern Recogn. Lett.*, 14(1):71–78, 1993.

[23] L. Yi, Z. Yefeng, D. David and J. Stefan, "Script-Independent Text Line Segmentation in Freestyle Handwritten Documents", *LAMP-TR-136/ CS-TR-4836/ UMIACS-TR-2006-51/ CFAR-TR-1017*, December 2006.

[24] Y.Pu and Z.Shi, "A natural learning algorithm based on hough transform for text lines extraction in hand written documents", *in Proceedings sixth International Workshop on Frontiers of Handwriting Recognition*, 1998, pp 637 646.

[25] A. Zahour, B. Taconet, P. Mercy and S. Ramdane, "Arabic Hand-Written Text-Line Extraction", *ICDAR*, 2001, pp 281–285.

[26] S. Zhixin, S. Srirangaraj and G. Venu, "Text Extraction from Gray Scale Historical Document Images Using Adaptive Local Connectivity Map", *ICDAR '05: Proceedings of the Eighth International Conference on Document Analysis and Recognition*, 2005, pp 794–798, Washington, DC, USA. IEEE Computer Society.