A Fast Multifunctional Approach for Document Image Analysis

Abhishek Gattani, Maitrayee Mukerji and Hareish Gur
Newgen Software Technologies Limited
agattani@newgen.co.in, maitrayee@newgen.co.in, hareish@newgensoft.com

Abstract

Collinear arrangement of objects (such as, text elements or continuous lines) is integral part of any office document image, whether structured or unstructured. The ability to analyze such an organization of objects thus provides the basic and important building block for a plethora of image analysis applications. Most Hough Transform-based line detection approaches do not furnish line widths and other measurements, and are computationally expensive for large-sized images. Other approaches often deploy a filter or morphological operation as a pre-processing step, which introduces reverse noise pattern while attempting to solve the cleaning problem in generalized manner. We propose an algorithm for fast, accurate, efficient and customizable detection of lines, which returns complete description of lines without having to apply an image pre-processing or conditioning step. Our approach, furthermore, allows simultaneous removal/ reproduction of lines, which is invariably used in the later phases of image analysis for higher-level interpretation and matching. The speed and flexibility of the approach presented here makes it serve as a multifunctional building block for a variety of document image analyses. The integration of this approach as a building block for diverse application areas have been implemented and explained.

1. Introduction

The physical control of recorded information began in the late nineteenth century when the universe of what we now call information management was defined by paperwork management. Since then, automation of paper-based processes has been propelled in different directions by various contemporary technologies of the day. But ever since business computers first saw the light of day, researchers have continuously focused their attention on solving the omnipresent problem of image analysis - for all kinds of paper-based applications. While many researches have contributed in advancing older techniques such as Hough Transform (HT), Skeletonization, Contouring, etc., there are others who have striven hard to make intractable problems be solved with pragmatic approaches, under constrained but real-life conditions. Herein, we present one such model for fast document image analysis using a generic line-detection algorithm, which replaces the basic component labeling approach with more meaningful higher-level solution for typical office document images [2,3,4]. The line detection approach in our model differs from other common approaches such as HT and recursive morphological operations in the execution speed, line information, response to real-life conditions and coupling depth and ease with other document image analyses to follow. We suggest that this algorithm be used in analysis applications, such as, document classification and separation, flexible table recognition, form registration, feature extraction, layout analysis for OCR formatting, skew angle analysis, line/ bar pattern detection, raster to vector conversion of lines, and the likes.

A typical recognition methodology comprises six steps: image formation, conditioning or pre-processing, labeling, grouping, extraction, and matching [4]. Of these, the latter five constitute a canonical decomposition of the recognition problem. Our proposition addresses labeling, grouping and extraction concerns, and serves as a foundation for application-dependent matching operations. Our model identifies and exploits the fact that lines and co-linear collection of objects (largely text) form the useful part of a typical office document. Hence our model presents line detection as a base module, which can be efficiently coupled with higher operations. It finds all interpretations of lines that satisfy the defined constraints. The conditioning step deployed before labeling by many researchers use filters or morphological operators to correct random unpatterned variations that affect measurements. However, while attempting to suppress noise, this step also introduces, reverse noise or errors. Our method, instead, alleviates the problem of noise by a selective compensation consideration, rather than general correction process. The proposed algorithm is extensible to detecting analytic shapes or even arbitrary geometric

Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR'03)
0-7695-1960-1/03 $17.00 © 2003 IEEE

models. In this paper, however, we focus of our attention only on the detection of hypothesized straight lines.

2. Common Approaches

Line segmentation methods can be widely grouped into HT-based, thinning-based, contour-based, run-graph-based, mesh-pattern-based and pixel-tracking-based. Most line segmentation methods assume that the image of the text does not change much and that lines are well separated. Since HT converts a global detection problem in image space into peak detection problems it can deal with noise, gaps and partial breakage even in complicated backgrounds. However, line detection for document images does not always have to necessarily deal with complicated backgrounds in practical solutions [6]. HT is also not suitable for application on large-size images usually because of its well-known weaknesses of time-inefficiency, the difficulty of choosing a proper threshold to distinguish short lines from noise and the missing line width. Most HT-based approaches are not able to detect line width. An abundant number of improved HT methods e.g. gradient-based HT, randomized HT [12], probabilistic HT [7] and sampling HT [9] have been proposed with the objective of decreasing computational time.

Li Xingyuan et. al. [11] proposed a Robust Method for Unknown Forms Analysis to deal with noise and disconnectivity. However, though their line segment extraction algorithm fills small breaks of form lines and line extraction filters out erroneous line segments, the extracted lines are far from ideal. Furthermore, for noisy document images opening and closing morphological operations are performed thus loosing information of the disconnectivity that forms a line. The morphological operations also increase the processing time.

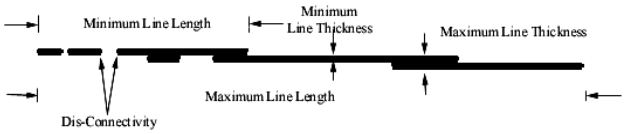
Palvidis et. al. [8] introduced a method based on adjacency graph that related connected component labeling to thinning. They used the approach for finding structures similar to skeletons, but because the method was dependent on the orientation, it could not be used to obtain good approximation of the medial lines or the end lines.

The existing methods of line detection have time complexity order ranging from linear to quadratic. Our approach operates with linear time complexity and offers greater amount of line information than any other known approach, thereby making it also serve as a model for fast document image analysis.

3. Hypothesizing Lines

3.1 Defining Line Constraints

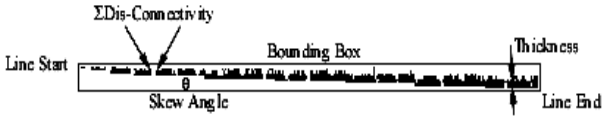
The specifications of a line are passed in a *hyposet* that describe the line constraints. The *hyposet* typically contains the minimum and maximum length, minimum and maximum thickness, maximum aspect ratio**,** *disconnectivity* threshold of the hypothesized line and orientation. Aspect ratio implies the ratio of the length (*longer side*) to the breadth (*shorter side*) of a line segment. *Disconnectivity* threshold is the minimum distance between two line segments for them to be treated disjoint.



"Figure 1. Hypothesizing Line"

3.2 Defining Line Description

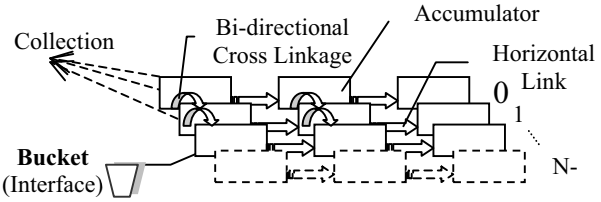
The algorithm returns the following information for each line segment detected: bounding box co-ordinates, precise start, end, thickness, skew, line priority based on heuristics and amount of white connected. See figure 2. All this information provides great flexibility to the higher document image analysis that follow.



"Figure 2. Line Information Returned"

4. Concept and Data Structure

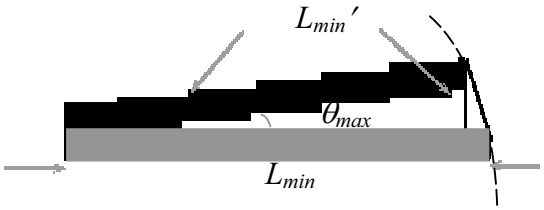
A 2-D *Cross-linked* LIFO list forms the basis of the algorithm presented. An *accumulator* is a repository for black runs collected. For each scan row there is a (LIFO) linked list containing aggregations of *accumulators* for black runs collected (*in that row*). Aggregations of *accumulators* are linked in both dimensions to yield *collections*. Each *collection* can have one or more *accumulators*. However, *a collection* can have one and only one *bucket*. A *bucket* serves as the information storehouse for a *collection*.



"Figure 3. 2-D Cross-linked LIFO"

5. Algorithm

Each row is traversed byte-wise (*one byte being a collection of eight pixels*) while skipping white pixels. Based on the specification of the hypothesized lines received from the user contiguous black run lengths in each row are collected in *accumulators* and put in a Last In First Out (LIFO) linked list. The line specifications received from the user are scaled down when validating an *accumulator* to see if it has the potential to be a part of a *collection*. Hence, validation occurs at two levels namely; the *accumulator* level and the *bucket* level. The reason for scaling down is that the users line hypothesis is ignorant of the skew. As can be seen in Figure 4, the skewed line suffers from the jagged effect resulting in reduction in the size of the run-lengths that form a part of it.



"Figure 4: Scaling Down User Line Hypothesis"

Upon further investigation using Bresenham's algorithm it can be fairly approximated that the length of the new run-lengths that form the part of the skewed line gets divided by the opposite side of the right triangle formed. For, instance, given minimum length of the line to be extracted (L_{min}), *disconnectivity* threshold (C_{max}), skew angle tolerance (θ_{max}) and the new scaled down threshold (L_{min}') is determined using

$$L_{min}' = L_{min} - C_{max} / (L_{min} . \text{Sin}(\theta_{max})); \quad \text{if } L_{min} . \text{Sin}(\theta_{max}) > 1$$

$$= L_{min} - C_{max} \; ; \quad \text{if } L_{min} . \text{Sin}(\theta_{max}) \leq 1$$

It may be noted that with current advances in ADF (Auto Document Feeder) technology, skew greater than ± 3 degrees is not found in the scanned documents.

For each scan row, a black run length is accumulated. If the LIFO linked list for that scan row is devoid of any *accumulators*, indicating that no black runs are collected. Hence, the very first *accumulator* containing information of the run-lengths is augmented. Note: for every new *accumulator* created a *bucket* is also created and initialized with the information of the *accumulator*. If the linked list contains *accumulators* then the last *accumulator* is fetched. The fetched *accumulator's* end co-ordinate is compared for *disconnectivity* with the new runs collected. If the two *accumulators* decide to be disjoint, the fetched *accumulator* is validated for being in the range [L_{min}', L_{max}]. Any *accumulator* not meeting the validation criteria is deleted. The new *accumulator* is augmented to the linked list. If connectivity was required then the fetched *accumulator* and the new *accumulator* are merged together resulting in one *accumulator* and one *bucket*. The number of pixels used to connect is updated in the merged *accumulator*. Hence, *disconnectivty* information is captured as a by-product. *Accumulators* of the current and previous scan rows are compared to determine which *accumulators* form parts of the same *collection*. A look-behind in the vertical direction implies the vertical *disconnectivty*. In short, two parallel lines separated by one pixel gap are considered as one with a look-behind of *two*. Coming back to look-behind of one, the *accumulator*'s of the adjacent rows are compared and cross-linked, if they overlap. The comparison is not iterative instead parallel since the last compared *accumulator's* forms that start of the next comparison. The task of cross-linking may seem trivial but cross-linkages lead to four scenarios, which have to be taken care. For each scenario, the data structure complements the algorithm optimally since now the LIFO linked list becomes a bi-traversal cross-linked map. These scenarios effectively deal with all the possible situations that can take place when cross-linking nodes across rows.

5.1 Case one

-Current and previous accumulators are NOT part of any collection. When current and previous accumulators overlap their buckets are merged and bounding box, start, end, thickness, skew, disconnectivty, etc are updated. A flag is set to imply that they now are of the same collection. Note: In Fig. 5, 6, and 7, gray denote current rows and black the previous rows.

"Figure 5: Illustrating Case one"

5.2 Case two

-Current accumulator is NOT a part of collection but previous accumulator is. The previous *accumulator* is traversed down to the *bucket* of the *collection* to which it belongs. The current *accumulator* is compared for overlap, and joined to the *collection* of the previous *accumulator*.

"Figure 6: Illustrating Case two"

5.3 Case three

-Previous accumulator is NOT a part of line and current accumulator is part of line. Case analogous to previous, except the situation is reverse.

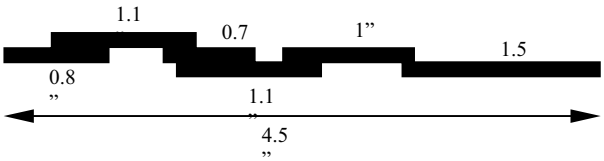
5.4 Case four

-Current accumulator and Previous accumulators are both part of line. In this case current as well as previous belong to some collection. Hence, these collections need to be merged. The current *accumulator* and previous *accumulator* are traversed down to get their respective *buckets*. The two buckets are merged if they overlap.

"Figure 7: Illustrating Case four"

5.5 Thickness

A *bucket* (B_i) has n *accumulators* where *i* is the row index. In Figure 8, all *accumulators* have unit widths and are labeled by their length in inches.



"Figure 8: Determining Line Thickness"

The thickness is given by T_{acc} / T_{buc} where $T_{acc}=\sum_{i=0..n-1} Length(Acc_i)$ and $T_{buc}= Length\ (B_i)$.

The *buckets* are then validated to fit the criteria for the hypothesized lines imposed by the user. Its interesting to note that our approach also allows to simultaneously remove/ reproduce a line *accumulator* by *accumulator* by traversing the 2-D Cross-linked Map depicted in Figure 3. The skew in the line is determined from the *bucket* since the line start, end and thickness are known.

Typically, noise is a random unpatterned *collection* of small pixels that affect measurements. Therefore, the detection process is not affected by noise gaps falling outside the *disconnectivity* threshold. For noise gaps occurring within the threshold, noise quantization could be done and stored in the *accumulator*. This noise quantization could be ratio of the number of black pixels in an *accumulator* to the number of connections that were made. An *accumulator* having a higher ratio is more likely to be a noise pattern and can be rejected from the *collection.*

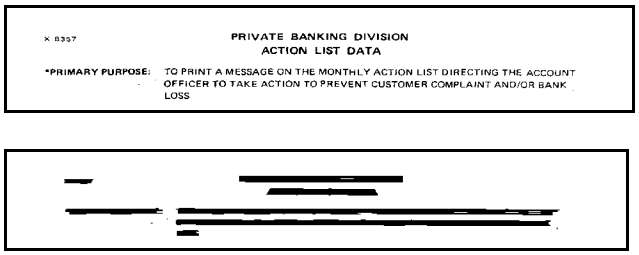
This algorithm follows linear time complexity and provides enough line information that greatly augments its purpose as a building block. The algorithm provides for advanced line removal and unlike other approaches it removes a line pixel-wise ensuring preservation of other data that overlaps with the line's bounding box. Moreover our approach speaks of a unified model for fast multifunctional document image analysis. The input to the model is a typical office document containing set of line segments, which are then detected using the aforementioned algorithm. The result is a cross-linked hierarchy of buckets, each with priority and line information. A higher document image analysis can select all possible buckets based on spatial locality, then make a selection based on the heuristics, assign priorities to the selection and select buckets, which have the maximum priority value for further analysis. The above description of the algorithm is general; it omits the details needed for an implementation.

6. Applications

Many image analysis applications including document classification and separation, flexible table recognition, form registration, feature extraction, layout analysis for OCR formatting, skew angle analysis, line/ bar pattern detection, raster to vector conversion of lines, etc., will get benefited by using this approach as a basic building block for the respective solutions. Let us describe one way of using it as a model.

6.1. Document Classification and Separation

The proposed algorithm, combined with modified A* Search Algorithm [1] can be used for comparing and classifying documents at spatial layout level [5] without OCR and with real-time performance. For all kind of documents that need to be separated, their line features are extracted and stored along with their interrelationships, as reference. The classifier algorithm can then speedily recognize any other document having the same or very similar spatial layout. Taking the detected line data for both the reference image and the current image, the classifier algorithm runs the Modified A* Algorithm to return the *confidence of similarity*. This approach also works for office documents that do not contain horizontal or vertical lines. Textual lines, when *smeared*, can be treated as a line on which the above-described approach works without requiring any further customization. Smearing an image is similar to applying the Constrained Run Length Analysis algorithm [10].



"Figure 9. Original & Smeared Images"

7. Experiments and Conclusions

Over 870 documents comprising structured, semi-structured and unstructured contents were scanned at resolution of 200 and 300 Dots Per Inch. Three different applications (Classification and Separation; Flexi-Table Identification; and Forms Registration) integrated a common implementation of the proposed algorithm to detect horizontal and vertical lines, and utilized the line information returned for higher level decision making in different ways. It was observed that the lines meeting the hypothesis were always detected as expected. Our 'C' language implementation took 0.2 sec on an average to detect horizontal lines on a Pentium III 800 MHz, Windows NT system. Further, the higher-level image analyses speeded up significantly. The worst and the best case executions took 0.81 sec and 0.06 sec, respectively.

Unlike most other approaches such as Hough Transform (and its variations), pixel tracking, contouring, etc., the presented approach meets multiple objectives of complete line description, customizability, simultaneous line removal/ reproduction and computational efficiency.

This approach is not limited to the presence of continuous graphical lines since it also works for roughly collinear collection of objects such as text elements. Many other published approaches use filters or morphological operators in the pre-processing stage to correct random unpatterned variations and thus affect measurements. The proposed algorithm, by contrast, is robust advert to noise tolerance in practical situations, since it works on selective compensation considerations rather than generalized image correction process. Further, noise validation rules can be defined by associating appropriate parameters with *accumulator*s for higher-level noise handling. The integrated performance of the algorithm has demonstrated its potential as an important building block for a range of fast and accurate solutions for document image analysis problems.

8. References

[1] Uli Bohnacker, Schacht Johannes and Yucel Tulug, "Matching Form Lines Based on a Heuristic Search", *Proc. 4th International Conference on Document Analysis and Recognition*, Ulm, Germany, 1997, pp. 86-90.

[2] Danielsson, P.E., "An improved segmentation and coding algorithm for Binary and Non-binary images", *Research and Development*, IBM J, Vol. 26, 1982, pp. 698-707.

[3] Dillencourt, M.B., H.Samet, and M. Tammininen, "General approach to Connected-Component Labelling for Arbitrary Image Representations", J.ACM Vol 39, No.2, 1992, pp. 253-280.

[4] Harlick Robert M., Linda and G.Shapiro. *Computer and Robert Vision Volume I*, Addison Wesley, Reading, MA, 1992.

[5] Hu Jianying, Ramanujan Kashi, Gordon Wilfong. "Document Image Layout Comparison and Classification". *Proc. 5th International Conference on Document Analysis and Recognition*, Bangalore, India, 1999 pp. 197-200.

[6] J. Illingworth and J. Kittler, "A survey of the Hough transform", CVGIP, Vol. 44.87-116, 1988.

[7] N. Kiryati, Y.Eldar and A.M. Bruckstein, "A probabilistic Hough Transform", *Pattern Recognition*, Vol 24, No. 2, 1991 pp. 303-316.

[8] Pavlidis, T, *Algorithms for Graphics and Image Processing,* Computer Sci. Press, New York, 1982 pp. 195-214.

[9] P. K. Ser and W. C. Siu, "Sampling Hough algorithm for the detection of lines and curves", in *Proceedings of IEEE International Symposium on Circuits and Systems*, Hough, vol .5, 1992, pp. 2497 –2500.

[10] Wahl M. Friedrich, Kwan Y. Wong, Richard G.Casey, "Block Segmentation and Text Extraction in Mixed Text/ Image Document", *Computer Graphics and Image Processing*, Vol 20, 1982, pages 375-390.

[11] Li Xingyuan, David Doermann, Weon-Geun Oh, Wen Gao, "A Robust Method for Unknown Forms Analysis", *Proc. 5th International Conference on Document Analysis and Recognition*, Bangalore, India, 1999, pp. 531-534.

[12] L. Xu and E.Oja, "Randomized Hough Transform (RHT): basic mechanisms, algorithms, and computational complexities", CVGIP: *Image Understanding*, Hough, Vol. 52, No. 2, 1993, pages 131-154.