Document Layout Analysis using Multigaussian Fitting

Laiphangbam Melinda
School of Computer and
Information Sciences
University of Hyderabad
India, 500046
Email: mel.laiphangbam20@gmail.com

Raghu Ghanapuram
School of Computer and
Information Sciences
University of Hyderabad
India, 500046
Email: raghughanapuram@gmail.com

Chakravarthy Bhagvati
School of Computer and
Information Sciences
University of Hyderabad
India, 500046
Email: chakcs@uohyd.ernet.in

***Abstract*—This paper proposes a novel technique for layout analysis of documents with complex Manhattan layouts. The technique is designed for Indic script newspapers and works on many types of documents not necessarily with Indic scripts with Manhattan layout. The main idea behind the algorithm is to categorise the physical elements of a document into noise, text, titles and graphics based on their heights. A histogram of heights is computed from the bounding boxes of connected components and a multigaussian fit is used to discover optimal split points between the categories. The gaussian with the highest peak is assumed to correspond to running text. Running text regions are grouped into blocks using nearest neighbour analysis. These initial regions are further refined using a second-level classification of the other elements into graphics, light-coloured text on a dark background, and graphical separators. The resulting layouts show accuracies comparable to some of the best and most popular algorithms such as MHS (winner of ICDAR-RDCL2015 competition) and PRImA's Aletheia (tool developed by PRImA Research Lab). Results of testing on many Indic script newspapers and other documents, and comparison with Aletheia and MHS on ICDAR dataset show its performance. Our initial results on an Indic document dataset show high performance in identifying running text ($>98\%$) with an accuracy of 82% on identifying the other elements. Ground truth data for the Indic script newspaper documents is being generated for a more extensive quantitative testing. The strength of our algorithm is that it requires only one parameter - the number of gaussians to fit the height histogram data and is therefore easy to automate and adapt to many documents.**

***Index Terms*—Document Layout Analysis, Bounding Boxes, Height Histogram, Multigaussian, Nearest Neighbor.**

I. Introduction

Document Layout Analysis is the process of recognising the arrangement of physical and logical elements of a printed document. Physical elements are paragraphs of text, graphics such as photos, line drawings, decorations and columns of text such as tables. Logical elements are titles, sub-titles, captions, figures, sections, sub-sections, headers, footers, column-separators etc. Optical Character Recognition (OCR) is used to convert text into electronic form. While OCR for many European and Asian scripts is considered mature and commercially viable, layout analysis is still an open research problem, as seen from the ICDAR competition in 2015 [1].

Layout or arrangement of the documents ranges from simple to extremely complex. Documents like novels have a simple layout with usually a single column text, with few graphics, titles and others. They may have graphical section separators, headers and footers. Progressively complex layouts include technical papers, text-books, magazines, newspapers, fliers and brochures. Technical papers and text-books contain more graphics, more font sizes, tables and sometimes two columns when compared to novel layout. Newspapers and Magazines layouts contain multi-column text, a higher number of graphics, tables and variations in text styles. Documents such as fliers and brochures contain very complex free form layout.

Many methods exist in literature such as RLSA [2], X-Y cut [3], Whitespace analysis [4], DOCSTRUM [5], Voronoi [6] etc for handling page layout. X-Y cut [3] is a well known top-down approach whereas RLSA [2], DOCSTRUM [5] and Voronoi [6] are popular bottom-up approaches. In top-down approach, the document image is split repeatedly till homogeneous regions are obtained. In bottom-up approach, smaller regions are repeatedly grouped into larger regions such as words, lines, paragraphs etc. Top down approaches work well for Manhattan layouts but generally have difficulty with Non-Manhattan layouts. Bottom-up approaches are considered better for handling complex layouts. Hybrid approaches combine top-down and bottom-up methods. Many variation of such basic methods have also been proposed [7]–[14] for handling complex layouts. In ICDAR2009 Page Segmentation competition [15], four different methods were submitted in which Fraunhofer Newspaper Segmenter (FNS) method showed the highest overall performance. In ICDAR-RDCL2015 competition [1], MHS [16] method achieved the highest overall performance among the four submitted methods which also include FNS, the winner of 2009 Page Segmentation competition.

In literature, layouts are broadly classified into Manhattan and non-Manhattan layouts. Manhattan layouts arrange elements in a grid form and the element boundaries are straight lines parallel to the edges of the documents. Newspapers are a good example of complex Manhattan layouts. Non-Manhattan layouts may be seen in brochures and magazines. The elements are placed in arbitrary orientations with non-linear boundaries. Non-Manhattan layouts present a great challenge to layout analysers and this paper deals only with Manhattan layouts.

2379-2140/17 $31.00 © 2017 IEEE
DOI 10.1109/ICDAR.2017.127

Though MHS [16], [17] works well for the ICDAR dataset, however, fails on newspaper documents which have multi-column, narrow gap between line which results in merged regions due to morphology operation. Also, different layout regions are generated with different smoothing kernel.

Layout analysis of Indic script documents has not been studied extensively [18]–[21]. Indic scripts fall into two main categories: those that have a headline joining all the characters of a word and those without. The north Indian scripts generally have a headline while the scripts from south and west India do not have a headline. The algorithm proposed in this paper is for scripts without headlines. Although, we developed the algorithm initially on Meetei Mayek (MM) script, our results demonstrate that the algorithm is more general and applicable to a variety of scripts such as Odia, Kannada, Tamil, Malayalam and Telugu as well as English. MM is used for Manipuri language from state of Manipur in North-East India. It does not use a headline but has several similarities to North Indian scripts in the use of modifiers. Results on English newspapers and magazines show that our algorithm is comparable in performance to some of the popular layout analysis algorithms for English.

We propose here a method based on height histogram of the connected components for handling complex layout and also with less parametric dependence as compared to MHS. This paper is organised as follows, section II presents the proposed work, section III analyses the experiments and section IV discuses the conclusion and future work.

II. Proposed Work

A document image contains mainly three elements: running text, titles and graphical objects. Running text or body text is the primary component and forms the main logical unit of a document. The layout is organised for easy reading of running text. Generally the font size of a running text is fixed for a document. Titles are also text but the font sizes are large, may be multicoloured with decorative fonts. Usually the font size is not fixed and there may be subtitles and other variations. Graphics are the most variable in size and can range from large photographs and line drawings to small section and column separators, headers and footers. Thus, the simplest recognisable element is the body text with its fixed font size.

The intuition is that if we compute a histogram of the heights of the different components in the document, we should get three peaks one for each of the major element viz. body text, graphics and titles. In practice, the height histogram shows many more peaks due to variations in graphics sizes, title sizes, irregular spacing between elements, etc. There is a need to preprocess the histogram to identify the peaks robustly. We also make the number of peaks as a parameter to our algorithm. If a document is logically simple and consists only of text, titles and graphics, three peaks are sufficient. If there are more complex elements such as subtitles, variable graphics, the number of peaks may be four or five. In our experiments, we found that there are only two peaks for novels, and three to five peaks for newspapers.

The proposed algorithm has the following steps:

- Binarization - using Sauvola's adaptive document image binarization [22] as it handles variations in illumination and noise.
- Connected Components(CCs) and Bounding Boxes(BBs) - Using OpenCV findContours() function.
- BBs analysis through histogram - Height histogram is smoothed using a Gaussian function to make it easier to find more stable peaks and valleys.
- Multigaussian fit
- Bin classification
- Classification of non-text

The last three steps which form the heart of the algorithm are explained below.

A. Multigaussian fit

The number of categories and hence the number of peaks in the height histogram depends on the complexity and the variety of elements used in the document. Therefore, the number of gaussians n is given as input to this step. *Note that this is the only parameter required by our algorithm and varies from two to five*. Once the desired number of Gaussians are fit to the height histogram, we find *split points* between them using Equation (1). In this example, we assume that three Gaussians with parameters $\mu_1, \mu_2, \mu_3, \sigma_1, \sigma_2$ and σ_3 are fit to the data.

$$SP = (a_1 \pm a_2)(a_3) \tag{1a}$$

where,

$$a_1 = \mu_2\sigma_1^2 - \mu_1\sigma_2^2 \tag{1b}$$

$$a_2 = \sigma_1\sigma_2\sqrt{(\mu_1 - \mu_2)^2 + 2(\sigma_2^2 - \sigma_1^2)\log(\frac{\sigma_2}{\sigma_1})} \tag{1c}$$

and

$$a_3 = \frac{1}{\sigma_1^2 - \sigma_2^2} \tag{1d}$$

The last split point is computed from the last bin using Equation (2) where $\overline{h}$ is the mean height

$$SP_n = 3\sigma_n + \overline{h} \tag{2a}$$

and,

$$\overline{h} = \frac{\sum(heights * frequencies)}{\sum frequencies} \tag{2b}$$

The outputs of multigaussian fit are the split points for differentiating between document elements.

B. BIN classification

1) Text classification: Once the optimal SPs are obtained, the gaussian distributions can be classified into bins. The first gaussian distribution contains small elements such as punctuation and noise. In Indic scripts, it also includes modifiers that are not connected to the characters above or below. The bin with the highest peak (usually the second) is assumed to represent running text. All the components that correspond to bins other than the running text are filtered out for this step. Running text (RT) blocks are formed by merging the nearest BB. The nearest BB is found by computing top and bottom

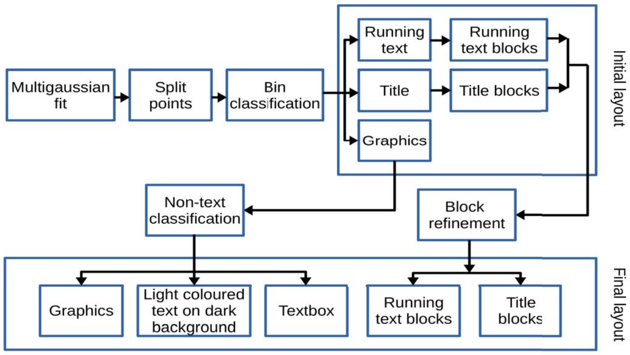


Fig. 1. Block diagram of the proposed algorithm

Algorithm 1 Bin Classification

Input: Binarized image

Output: Bin_i where $i = 1, ..., n$ and n is the number of bins

```
procedure SplitPoints($I_{binarize}$)
    for all $CC_i \in I_{binarize}$ do
        $height_i \leftarrow height(BB(CC_i))$
    end for
    $H \leftarrow \{height_i\}$
    $Hist(H) \leftarrow Histogram(H)$
    $Hist_s(H) \leftarrow SmoothHistogram(Hist(H), \sigma)$
    Initial guess $P_0 \leftarrow \{\mu_1, \sigma_1, \mu_2, \sigma_2, \mu_3, \sigma_3\}$
    Best fit $P \leftarrow curvefit(P_0, Hist_s(H))$

    $SP_k \leftarrow intersection(G_k, G_{k+1})$    ▷ where $k = 1, ..., n-1$
    $SP_n \leftarrow 3\sigma_n + meanHeight$    ▷ last split point
    $Bin_i = \{BB_j | SP_k < H_j < SP_{k+1}\}$    ▷ where $j = 1, ..., n$ and $n$ is the number of components
return $Bin_i$
end procedure
```

Next Nearest Neighbor (NNN) distance in Equation (3). Once the nearest block is found, it is merged with the previous block to build a bigger block. Merging threshold is set one less than the higher split point and running text blocks are obtained by computing connected components.

$$D^v_{it} = dist\left(BB_i, BB^t_i\right) \tag{3a}$$

and

$$D^v_{ib} = dist\left(BB_i, BB^b_i\right) \tag{3b}$$

where, D^v_{it} is the top vertical distance,
D^v_{ib} is the bottom vertical distance,
BB^t_i is the top vertical NNN,
BB^b_i is the bottom vertical NNN and
$dist$ is the Euclidean distance

$$RT_i = \begin{cases} merge\left(BB_i, BB^t_i\right), & if\ D^v_{it} < D^v_{ib} \\ merge\left(BB_i, BB^b_i\right), & if\ D^v_{it} > D^v_{ib} \end{cases} \tag{3c}$$

After completing the layout of running text, we pick the next higher bin (usually the third) as titles. Title blocks are formed from title bin elements using a similar approach of merging nearest neighbours.

2) Blocks refinement: There may be an overlap between adjacent running text and title blocks after the initial analysis in the previous step. If there are more than two regions that overlap, then they are merged into a single large overlapped area. The majority class of the components within this region is used to merge minority components with the majority class. If the dominant components belong to one of the two overlapping blocks, then the other block is adjusted.

C. Classification of non-text

1) Graphics (G): The bin containing the largest height components is considered as graphics bin. The graphical components can be true graphics such as photographs and drawings, text boxes (i.e., text regions enclosed by a box) and light text on dark and complex backgrounds. These are now correctly classified.

Horizontal Projection Profile (HPP) is used to discriminate between graphics and text boxes. Text boxes show HPP with deep valleys and sharp peaks corresponding to interline spaces and textlines. Graphics do not show such large variations. The half-peak width of HPP is computed and if greater than the split-point given in Equation (2), then it is a graphics block; otherwise, it is a text box.

2) Light coloured text on dark background (TDB): Minimum Bounding Rectangle (MBR) is used for determining light coloured text on dark or complex backgrounds. We reduce the size of the minimum bounding rectangle by an amount based on the split points on either side of the bin to which the BB belongs. The number of black pixels in the maximum and minimum bounding boxes are calculated. Bounding rectangle is identified to be dark background text if the difference in the number of black pixels in maximum and minimum BBs is comparable to the number of pixels in the removed area.

3) Text Box (TB): Bounding rectangles are reduced from each side by SP_1 as threshold. HPP is computed for all bounding boxes of the reduced bounding rectangle. Height at the mid point is computed for all peaks and if the computed heights falls within text bin then it is a text box.

The algorithm requires only one parameter i.e., number of gaussians to be fit to the height histogram. This enables us to derive split points that separate the major elements. By analysing these elements separately, we eliminate confusing line-spacings, font size variations and column widths and get robust layouts. We wish to improve this algorithm further by using a Gaussian Mixture Model that allows for finding the optimal number of Gaussians automatically. It is hoped that such an approach removes even the one parameter that is needed now.

III. Results

We implemented our algorithm with OpenCV in C++ and Python for fitting the Gaussians to the height histogram. The

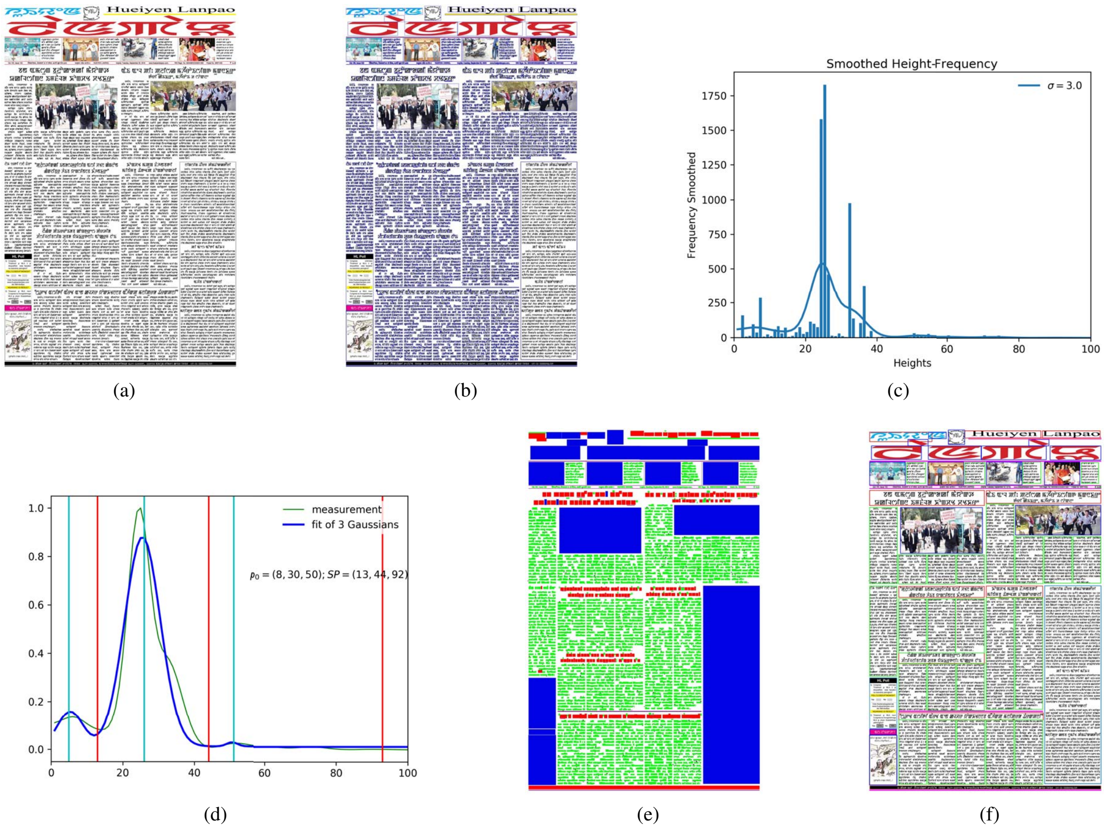


Fig. 2. Image showing step by step result of Proposed method on Meetei Mayek (MM) newspaper image (a) Input image, (b) Bounding boxes, (c) Smoothed height histogram, (d) Multigaussian fit, (e) Bin classification and (f) Final layout

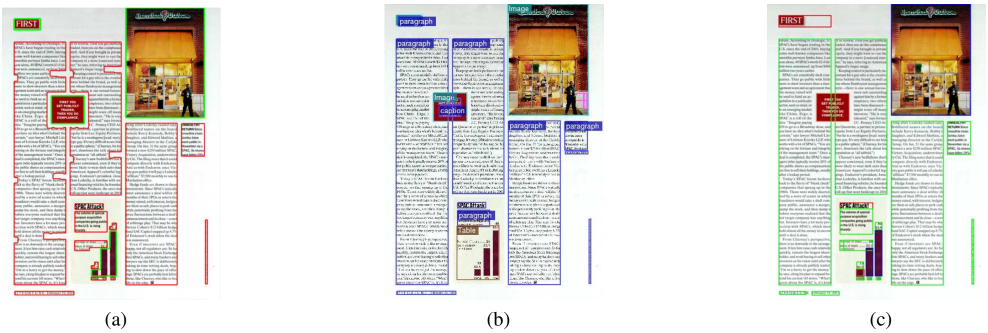
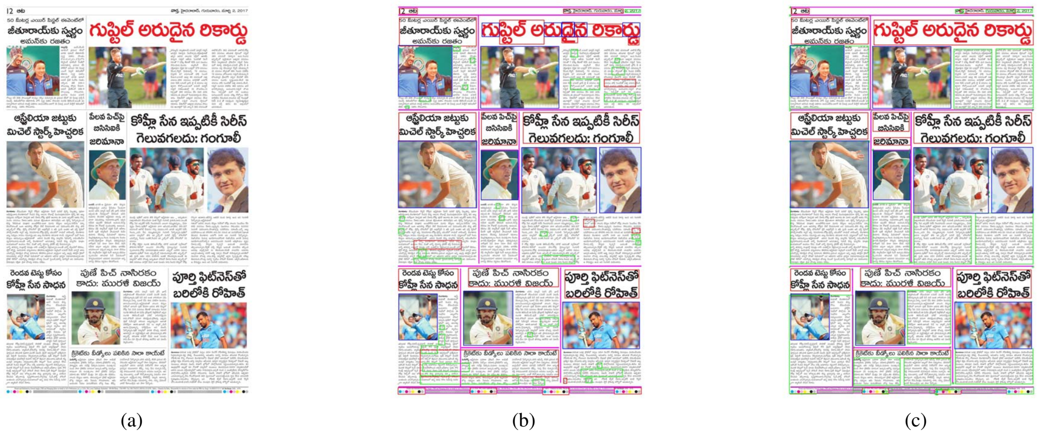


Fig. 3. Results on ICDAR-RDCL2015 dataset (a) MHS method- outlines in red and green denotes text and graphic, (b) PRImA's Aletheia and (c) Proposed method - outlines in red, green, blue and salmon denotes title, text, graphic and text on dark background

documents are scanned at 300 dpi and the average size of a newspaper image is 4000×6000 pixels and 2000×3000 pixels for books. A total of 510 documents have been collected consisting of 375 newspaper images, 120 book images and 25 ICDAR [1] documents. Our data includes six Indic scripts: Meetei Mayek (MM) (120 + 46), Telugu (108), Kannada (48), Tamil (50), Malayalam (35), Odia (34) apart from 44 English newspapers.

Results on a Meetei Mayek newspaper are shown in Figure 2. Figure 2(b) shows the extraction of bounding boxes. Figure 2(c) shows the smoothed height histogram with a Gaussian of $\sigma = 3$. Figure 2(d) shows three Gaussians fit to the histogram in which P_0 gives the mean values of the three bins and SP are the split points. Figure 2(e) shows the classification of bins where different colors are used for each bin identification - red, green, blue and purple denotes titles, running text, graphics and modifiers. Figure 2(f) shows the final layout result where the outlines in red, green, blue, salmon, cyan and magenta



(a) (b) (c)

Fig. 4. Image showing layout with different number of Gaussians (a) Input Image, (b) With 3 Gaussians and (c) With 4 Gaussians

TABLE I
F-MEASURE OF OUR ALGORITHM ON TWO DATASETS

F-measure	Indic(MM)	ICDAR-RDCL2015
Text	98.04	93.42
Non-text	81.49	87.65

TABLE II
EVALUATION RESULTS OF VARIOUS SCRIPT

Script	No. of Newspaper Documents	Success rate (%)
MM	46	98.91
Telugu	108	86.11
Tamil	50	84
Kannada	48	97.92
Malayalam	35	82.86
Odia	34	91.18
English	44	90.91

denotes titles, running text, graphics, text on dark background, text boxes and line separators.

Figures 3(a-c) compare our results with MHS [17] and PRImA's Aletheia on ICDAR dataset. For Meetei Mayek script, we have generated ground truth using PRImA's Aletheia ground truth generation tool [23]. It also shows the limitation of our algorithm which finds only rectangular blocks. However, it may be seen that we are able not only to analyse the layout but classify the different elements correctly. Note that Aletheia classifies light text on dark background as an image while ours classifies it as Text on Dark Background (TDB). MHS does not classify the document elements.

F-measures (which can be considered as weighted harmonic mean of detection rate and recognition accuracy) of our algorithm on MM and ICDAR datasets are shown in Table 1. Table 2 summarises the evaluation results on the entire newspaper dataset and for 120 book images, we achieved 82.5%. We achieve an F-measure of 98% on text and 82% on non-text elements on our Indic script document dataset which is superior to that obtained with Tesseract used in Aletheia tool (Figure 5). On the English documents in the ICDAR dataset, we achieve almost the same performance as Aletheia (93% for text and 87% for non-text) although the primary target for our algorithm is Indic script newspapers and books.

A more interesting document, a newspaper in Telugu language, is shown in Figure 4. A feature of the Telugu language is that the running text contains characters with two distinct sizes: basic characters and characters with modifiers attached to them. The lower modifiers are also often the same size as running text. All these lead to multiple peaks in the height histogram for running text and requires 4 or even 5 Gaussians to be fit. The layout analysis with 3 Gaussians and 4 Gaussians are shown in Figures 4(b) and (c). It may be seen that the layout in Figure 4(c) is correct.

A. Discussion

In this paper, we proposed a new layout analysis algorithm primarily for documents with complex Manhattan layouts and especially for Indic scripts. The algorithm, however, is general enough to give performance comparable to popular tools for English documents too. There are a few limitations of the proposed approach. The first is that it deals only with rectangular blocks. However, as the approach is bottom-up, it may not be difficult to extend it for polygonal blocks as required in the example shown in Figure 3.

A second limitation is that it may have difficulty in handling light *running* text on dark backgrounds. Often such text is found in footers in magazines, gets merged with running text and causes failure in correctly determining column widths. There are two ways of handling it. The first is to detect the reverse coloured text and remove it while finding text blocks and paragraphs. In the second case, especially in newspapers, there may be other text which is correctly analysed. In such a case, any component whose width is larger than that of a paragraph can be identified as reverse coloured text or a graphical line separator.

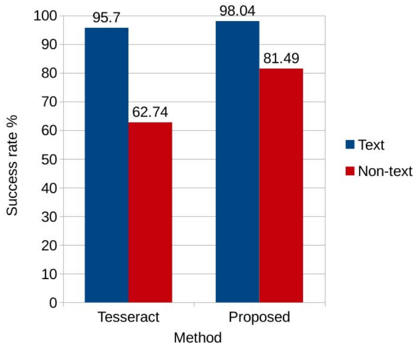


Fig. 5. F-measure of our method and Tesseract method in our dataset

IV. Conclusion

We proposed an algorithm for layout analysis that is based on height histogram of the connected components and multigaussian fitting. Our algorithm is simple and easy to automate as it requires only one parameter - the number of gaussians to fit the height histogram data - and therefore may be adapted to many types of documents such as newspapers, books and magazines. In future work, we are going to use GMM (Gaussian Mixture Models) with EM (Expectation-Maximization) to improve the performance and possibly eliminate even the one parameter. Our results on Indic script documents show high accuracy in layout analysis as well as in classifying the various document elements. Results on English documents show that the algorithm gives a performance comparable to some of the best available tools.

Acknowledgment

The authors acknowledge Professor Atul Negi for valuable suggestions, comments and feedback on the initial versions of the algorithm.

References

[1] A. Antonacopoulos, C. Clausner, C. Papadopoulos, and S. Pletschacher, "ICDAR2015 competition on recognition of documents with complex layouts-RDCL2015," in *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pp. 1151–1155, IEEE, 2015.

[2] F. M. Wahl, K. Y. Wong, and R. G. Casey, "Block segmentation and text extraction in mixed text/image documents," *Computer Graphics and Image Processing*, vol. 20, no. 4, pp. 375 – 390, 1982.

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

[4] H. S. Baird, "Background structure in document images," in *In Advances in Structural and Syntactic Pattern Recognition*, pp. 17–34, World Scientific, 1992.

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

[6] K. Kise, A. Sato, and M. Iwata, "Segmentation of page images using the area voronoi diagram," *Comput. Vis. Image Underst.*, vol. 70, pp. 370–382, June 1998.

[7] J. Ha, R. M. Haralick, and I. T. Phillips, "Recursive x-y cut using bounding boxes of connected components," in *Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 2) - Volume 2*, ICDAR '95, pp. 952–, IEEE Computer Society, 1995.

[8] T. M. Breuel, "Two geometric algorithms for layout analysis," in *Proceedings of the 5th International Workshop on Document Analysis Systems V*, DAS '02, (London, UK, UK), pp. 188–199, Springer-Verlag, 2002.

[9] J. Xi, J. Hu, and L. Wu, "Page segmentation of chinese newspapers," *Pattern Recognition*, vol. 35, no. 12, pp. 2695 – 2704, 2002. Pattern Recognition in Information Systems.

[10] H.-M. Sun, "Page segmentation for manhattan and non-manhattan layout documents via selective crla," in *Proceedings of the Eighth International Conference on Document Analysis and Recognition*, ICDAR '05, pp. 116–120, 2005.

[11] S. Ferilli, T. M. A. Basile, and F. Esposito, "A histogram-based technique for automatic threshold assessment in a run length smoothing-based algorithm," in *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, DAS '10, pp. 349–356, ACM, 2010.

[12] N. Nikolaou, M. Makridis, B. Gatos, N. Stamatopoulos, and N. Papamarkos, "Segmentation of historical machine-printed documents using adaptive run length smoothing and skeleton segmentation paths," *Image Vision Comput.*, vol. 28, pp. 590–604, Apr. 2010.

[13] K. Chen, F. Yin, and C.-L. Liu, "Hybrid page segmentation with efficient whitespace rectangles extraction and grouping.," in *ICDAR*, pp. 958–962, IEEE Computer Society, 2013.

[14] M. Felhi, S. Tabbone, and M. V. O. Segovia, "Multiscale stroke-based page segmentation approach," in *Document Analysis Systems (DAS), 2014 11th IAPR International Workshop on*, pp. 6–10, IEEE, 2014.

[15] A. Antonacopoulos, S. Pletschacher, D. Bridson, and C. Papadopoulos, "Icdar 2009 page segmentation competition," in *2009 10th International Conference on Document Analysis and Recognition*, pp. 1370–1374, July 2009.

[16] T. A. Tran, I.-S. Na, and S.-H. Kim, "Hybrid page segmentation using multilevel homogeneity structure," in *Proceedings of the 9th International Conference on Ubiquitous Information Management and Communication*, p. 78, ACM, 2015.

[17] T. A. Tran, I. S. Na, and S. H. Kim, "Page segmentation using minimum homogeneity algorithm and adaptive mathematical morphology," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 19, no. 3, pp. 191–209, 2016.

[18] A. R. Chaudhuri, A. K. Mandal, and B. B. Chaudhuri, "Page layout analyser for multilingual indian documents," in *Language Engineering Conference, 2002. Proceedings*, pp. 24–32, Dec 2002.

[19] P. B. Pati, S. S. Raju, N. Pati, and A. Ramakrishnan, "Gabor filters for document analysis in indian bilingual documents," in *Intelligent Sensing and Information Processing, 2004. Proceedings of International Conference on*, pp. 123–126, IEEE, 2004.

[20] P. Dasigi, R. Jain, and C. V. Jawahar, "Document image segmentation as a spectral partitioning problem," in *2008 Sixth Indian Conference on Computer Vision, Graphics Image Processing*, pp. 305–312, Dec 2008.

[21] V. Singh and B. Kumar, "Document layout analysis for indian newspapers using contour based symbiotic approach," in *Computer Communication and Informatics (ICCCI), 2014 International Conference on*, pp. 1–4, IEEE, 2014.

[22] J. Sauvola and M. Pietikinen, "Adaptive document image binarization," *PATTERN RECOGNITION*, vol. 33, pp. 225–236, 2000.

[23] C. Clausner, S. Pletschacher, and A. Antonacopoulos, "Aletheia - an advanced document layout and text ground-truthing system for production environments," in *2011 International Conference on Document Analysis and Recognition*, pp. 48–52, Sept 2011.