DOCUMENT LAYOUT ANALYSIS SYSTEM

Andrei Alexandru ALDEA [1*]
Radu Gabriel CORIU [2]
Ştefan-Vlad PRAJICĂ [3]
Răzvan-Ştefan BRÎNZEA [4]
Costin-Anton BOIANGIU [5]

ABSTRACT

The need to process large amounts of printed physical data has led to the development of automated solutions for scanning and converting such documents into an editable text format. Following the layout analysis process, the different areas (blocks) of the document can be labeled by content - text, image, tables. Such an analysis of the document is referred to as geometric analysis. A different approach is that of a logical layout analysis, or semantic analysis, in which text blocks are labeled according to their role inside the document - titles, footnotes etc. Identifying sections correctly, numbering pages and arranging them in the correct order are standard requirements for OCR.

KEYWORDS: *document layout analysis, area Voronoi diagrams, voting system, whitespace cover*

1. INTRODUCTION

Layout analysis - the process of identifying and labeling the regions of interest contained in scanned text documents - is a prerequisite for documents intended to be converted to electronic format with optical character recognition algorithms. Even if this task is simpler than image segmentation, it still poses difficulties for background structure analysis.

There are two main approaches to layout analysis [1]:

- bottom-up: iteratively parses a document based on its pixel information. Generally, these approaches first parse the document into black and white regions. These regions are then grouped into words, rows, and then into text blocks.
- top-down: Approaches of this type attempt to iteratively separate the document into columns and blocks based on whitespace and geometric information.

[1*] corresponding author, Engineer, Ubisoft Romania, Bucharest, Romania, andrei.sateanu@gmail.com
[2] Engineer, Sparkware Technologies Romania, Bucharest, Romania, radu.gabrielc@gmail.com
[3] Engineer, Tangoe Romania, Bucharest, Romania, stefan-vlad.prajica@my.fmi.unibuc.ro
[4] Engineer, "Politehnica" University of Bucharest, Bucharest, Romania, razvan.brinzea@gmail.com
[5] Professor PhD Eng., "Politehnica" University of Bucharest, Bucharest, Romania, costin.boiangiu@cs.pub.ro

This paper presents an approach for document layout analysis system, featuring three algorithms - one *top-down* and two *bottom-up* algorithms, and a voter used to aggregate the generated results. Due to their error masking capability, voting algorithms are used in a wide range of commercial and research applications.

2. RESEARCH STANDARD AND APPROACHES

To begin with, we consider a recent development in document layout analysis, described by Breuel in [2].

Traditional layout analysis methods generally begin by attempting a global and complete segmentation of the document in distinct geometric regions corresponding to entities such as columns, titles, and paragraphs, using proximity, texture, or white space. Although these sections can then be processed individually with good results, obtaining a segmentation that correctly mirrors the document's layout is a task that is very difficult to generalize.

The implementation depicted in [2] uses accurate and optimal algorithms combined with robust statistical models to model and analyze the layout of the pages.

The first step is to determine the background structure of the pages by assuming that there is a collection of rectangles in the plane, bounded by a given bounding box. The main idea is to find a rectangle that maximizes $Q(T)$ (where Q is the evaluation function, often just the area of the rectangle) among all possible bounding rectangles, without overlapping any rectangle in the plane. Figure 1 illustrates a partial application of this idea.

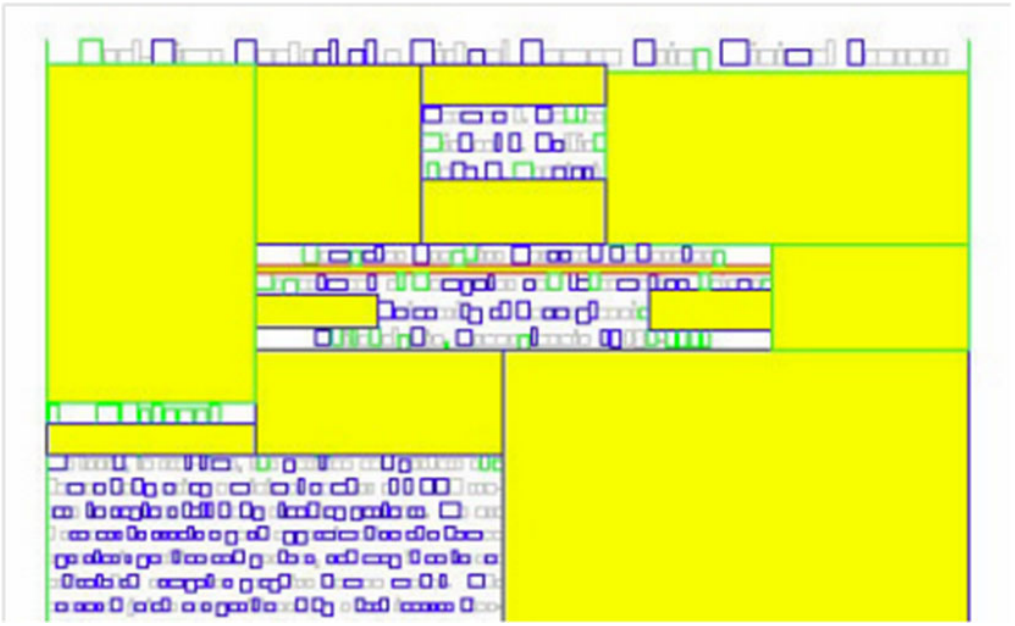


Figure 1. Partial coverage of white space using the greedy algorithm described in [2]

However, this step is insufficient in determining useful background information about the page. Next, a set of evaluation criteria will be used to determine the regions of white space that are “meaningful” (those that separate text). The requirement is that the identified rectangles are bounded by a minimum number of connected components on each side.

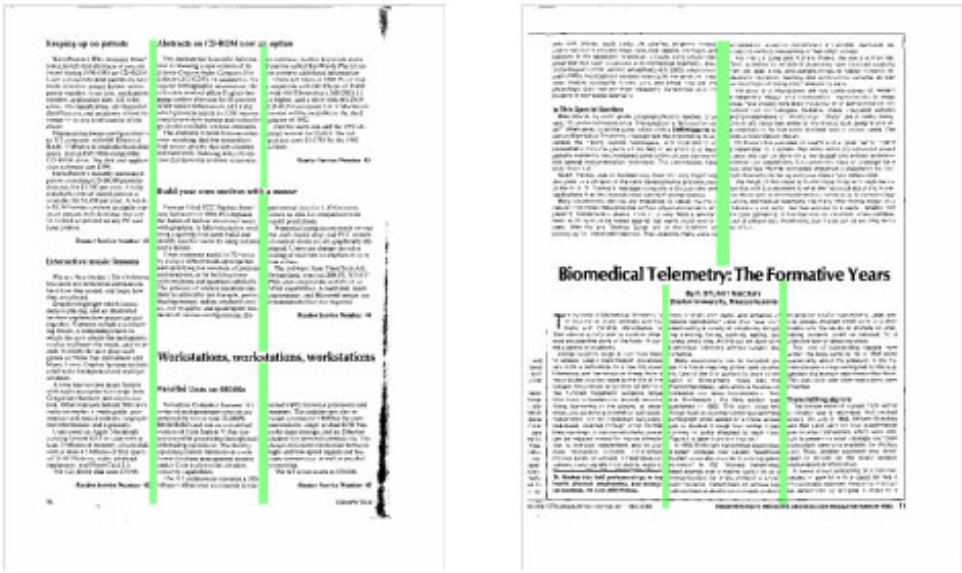


Figure 2. Automatically identified column delimiters

The previous idea can be used to identify white space aligned with the document’s axes , but an algorithm for identifying arbitrary whitespace rectangles can eliminate the need to correct the rotation of the page. It works somewhat analogously to the previous algorithm, receiving a list of foreground shapes as input (polygons or bounding boxes of words or characters) and returning the maximum white space rectangles that do not overlap and which meet certain parameter requirements (orientation, width, height, position).

The next step is to identify text lines, which can often prove problematic for complex layouts with multiple columns of varying widths. The idea is to use the column delimiters detected during the earlier processing steps as "obstacles" for a global *branch-and-bound* text line detection algorithm. This approach yields far better results compared to those of the traditional global or local methods.

Finally, determining the correct order in which the document blocks are read depends not only on the geometric layout of the document, but also on the linguistic and semantic content. A general approach is to use the following two ordering criteria, as demonstrated in Figure 3:

1. The line segment *a* precedes *b* if their *x*-coordinate regions overlap and the line segment *a* is above the line segment *b* inside the page.

2. The line segment *a* precedes *b* if *a* is entirely to the left of *b* and there is no line segment *c* whose y-coordinate is not between *a* and *b* and the *x*-coordinate area overlaps both *a* and *b*.

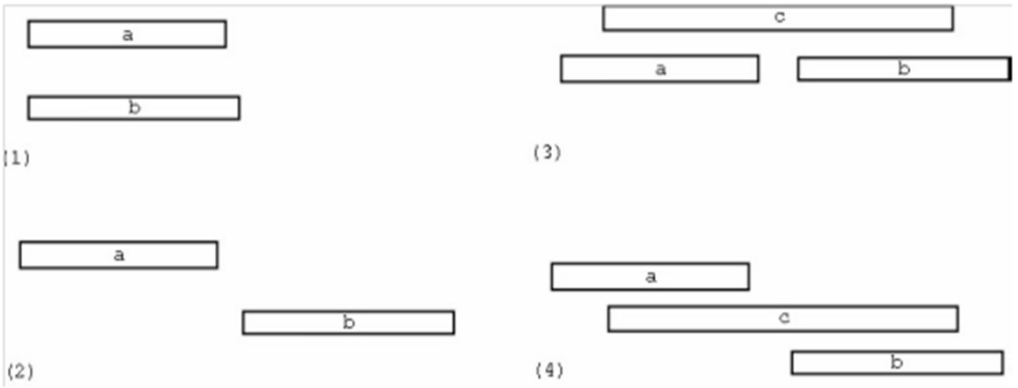


Figure 3. Topological sorting of text lines

3. VOTING-BASED LAYOUT ANALYSIS SYSTEM

In the following section, we propose a system for reliable document layout analysis, using three different algorithms and combining their results to better identify regions of interest and the emplacement of text areas within the document.

3.1. Overview

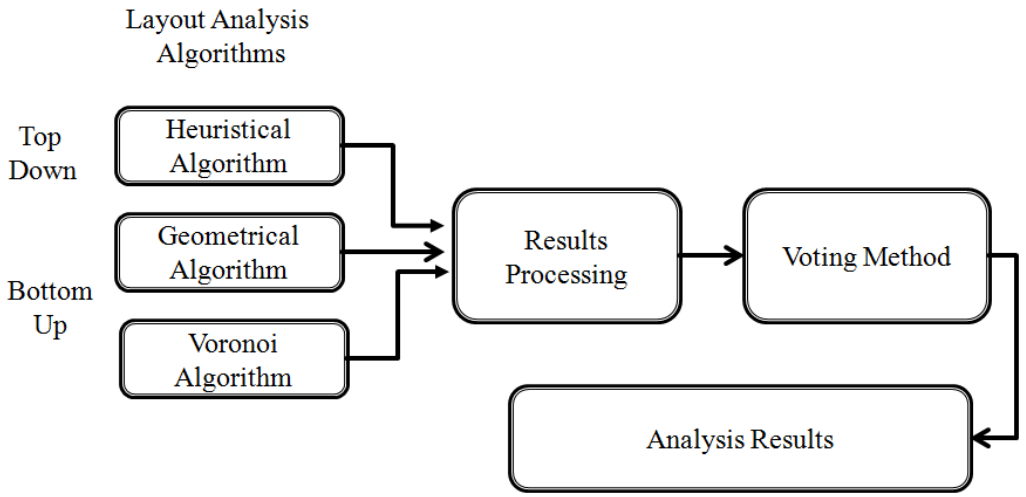


Figure 4. Overview of the layout analysis system - the voter aggregates the results of the layout analysis algorithms

3.2. Heuristic algorithm

The first algorithm uses a top-down approach which attempts to detect text through morphological modifications of the document, while also eliminating images. Based on some simple observations on multiple scanned document, we defined the algorithm, as presented below.

A Canny transformation is applied to the grayscale image to detect edges and remove color differences. The unnecessary details of the full image are discarded by downscaling to a maximum size (width or height) of 256 pixels. After experimenting on multiple samples, this value has proven sufficient to obtain good results, while significantly speeding up the processing. After downscaling, we apply a dilate-erode transformation with a 3x3 square kernel to merge small elements into bigger blocks.

To eliminate images, we added a preprocessing step in which we use OpenCV's *findContours* to obtain a hierarchy of shapes within the document. Then, the tree of contours is searched bottom-up and masks are generated. These masks are used to calculate the entropy of the different areas of the scanned document. Each region with an entropy of more than 2.5 is considered an image and discarded (set to 0). This matches the intuition that a region containing text generally has only 2 colors, which can be represented on one bit, but may sometimes have different more than that. Thus, the threshold of 2.5 yields good results.

After the image removal step, we move on to the actual detection of the layout (text paragraphs). A binarized, downscaled version of the image is intersected with the output of the previous steps, by applying a bitwise AND on the two images. The result is morphologically modified by a dilate-erode step.

On the resulting image, the contours are once again detected and their bounding rectangles are the final regions that represent the output of the algorithm.

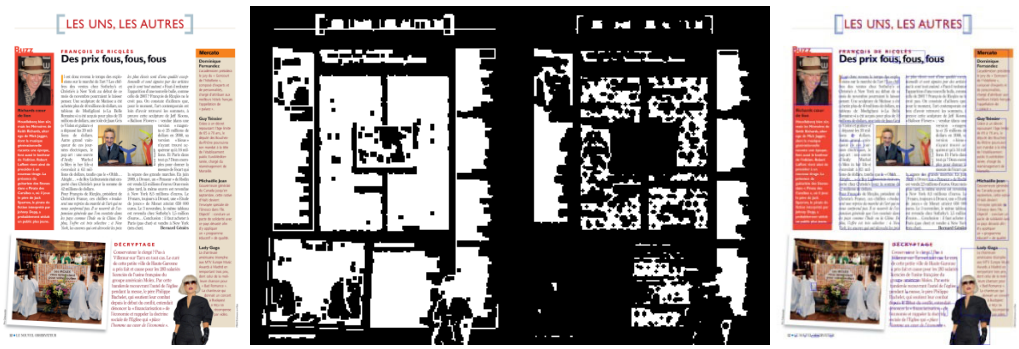


Figure 5. Document through different stages of processing: original, morphed Canny, morphed Canny with images removed and final output regions

3.3. Geometrical algorithm

The following algorithm, first introduced by Breuel in [3], then further discussed in [4], is a bottom-up heuristic approach. It attempts to find individual letters and then merge the bounding rectangles into text paragraphs.

The original document is passed through a Canny edge detection algorithm. The generated result is blurred and then a *findContours* function is applied. This step finds all the individual characters in the document.

Afterwards, we apply an incremental merge of the bounding rectangles. All rectangles with a certain ratio of black and white pixels are considered as text and merged. After multiple iterations of the algorithm we determined that the constant ratio of 0.2 yields the best results.

This process is applied iteratively until a preset constant number of final bounding rectangles is extracted. The generated output forms the final paragraphs which will be considered in the voting process.

3.4. Voronoi algorithm

In this section, we will briefly describe a method based on the approximated area Voronoi diagram for solving the page segmentation task of a layout analysis system [5]. Voronoi edges can be found between all adjacent connected components. This means that each component is represented as a set of adjacent Voronoi areas.

In order to achieve page segmentation, Voronoi edges - effectively, boundaries between various components in the document - are selected as a result of 3 main stages:

1. Through labeling, the connected components in the document are detected.
2. The area Voronoi diagram is generated
3. The unnecessary Voronoi edges are pruned.

Two criteria are used to determine the edges that can be eliminated:

- Minimum distance - edges found inside narrow spaces (such as spaces between characters) can be removed.
- Area ratio – because minimum distance is efficient for preserving edges between columns (which have thick white areas) but not as efficient for cleaning the boundaries between components.

Figure 6 shows the equivalent Delaunay triangulation of the area Voronoi diagram of a document processed as shown above.

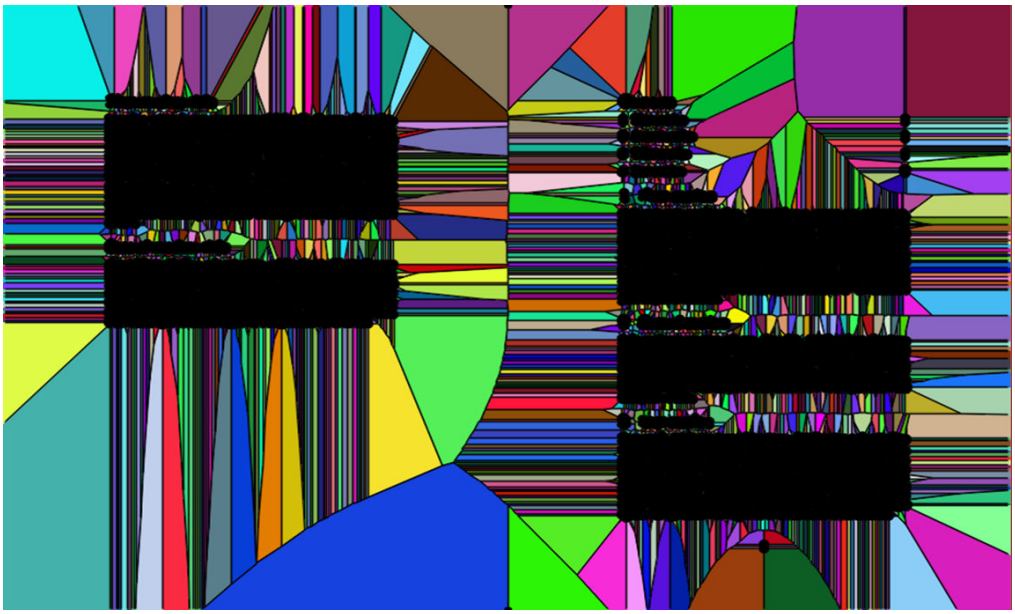


Figure 6. The Delaunay triangulation of a processed document

4. VOTING METHODS

With the use of a voting algorithm, a higher degree of accuracy can be obtained. Instead of running a single generic algorithm, several algorithms with some degree of similarity are used in parallel, and their results are combined. This approach can mask the errors that the methods may produce for specific inputs, when run individually.

Such voting systems can be grouped by very diverse criteria such as being either hardware voters or software, by the nature of the working environment - synchronous or asynchronous, but in this case, the most relevant classification is on their functionality [6].

Generic voters either choose one of the generated results, or combine them to produce a new one. In this category we include the following:

- unanimity voters: all generated results are in agreement
- majority voters: at least $\frac{n+1}{2}$ among n generated results agree
- plurality voters: m-out-of-n voting where m is less than a strict majority
- median voters: they produce a correct result up to a maximum of $\frac{n-1}{2}$ faulty inputs

In the presented system, the outputs of the algorithms presented at **3.2, 3.3** and **3.4** are collected and the voting is done on the identified rectangular surfaces. Votes are accumulated for every surface, and after using a *majority voter* and a *unanimous voter,* the aggregated results are compared with each other and against the individual performance of the algorithms (figure 7).

Figure 7. Comparison between the original document, voting from all the algorithms and unanimity voted regions

5. CONCLUSIONS

To benchmark our system we chose the LRDE dataset [7], which was also used in [8]. The dataset provides documents for input files and ground truths for OCR detection. We measured the processing time for each image in the set for each of the three algorithms, the accuracy of the majority vote (2-out-of-3) and that of the unanimity vote (all algorithms in agreement).

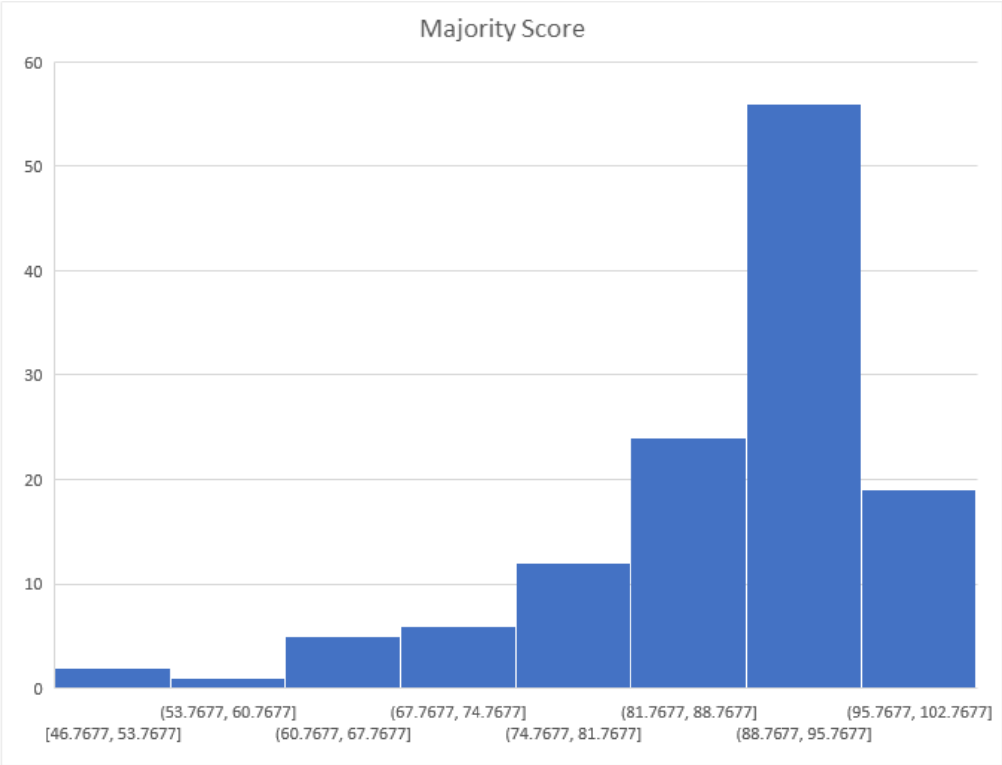
The success rate was calculated as a ratio between the number of pixels in the generated output and the value of the pixels in ground truth images, and the total number of pixels in the image.

The unanimity voting method yielded more consistent results. The average accuracy was 93.66% with a minimum of 54.91% and a maximum of 98.79%.

The majority voting yielded a more uniform histogram of results, with an average accuracy of 87.15%, a minimum of 46.76% and a maximum of 98.08%.

The images in the LRDE dataset are 2516 x 3272 pixels. The average running times over the 126 images dataset was 686*ms* for Heuristic, 5449*ms* for Geometrical and 2813*ms* for Voronoi.

As a future development, the presented research will be integrated into a full document image analysis system, thus combining several previously developed voting-based processing stages [9][10][11][12] in order to further improve the automatic detection accuracy.



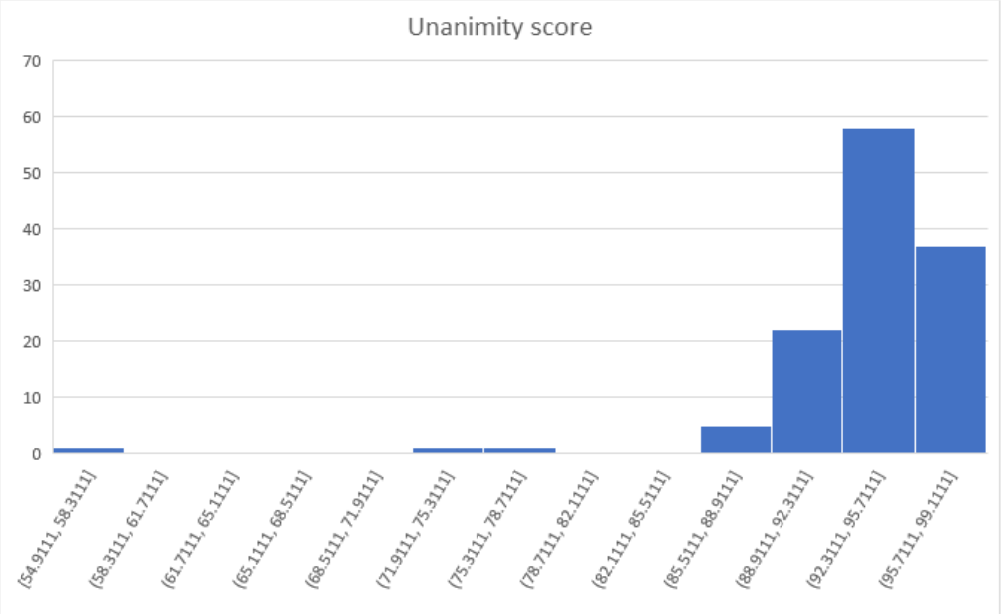


Figure 8. Comparison of the majority and unanimity voting methods

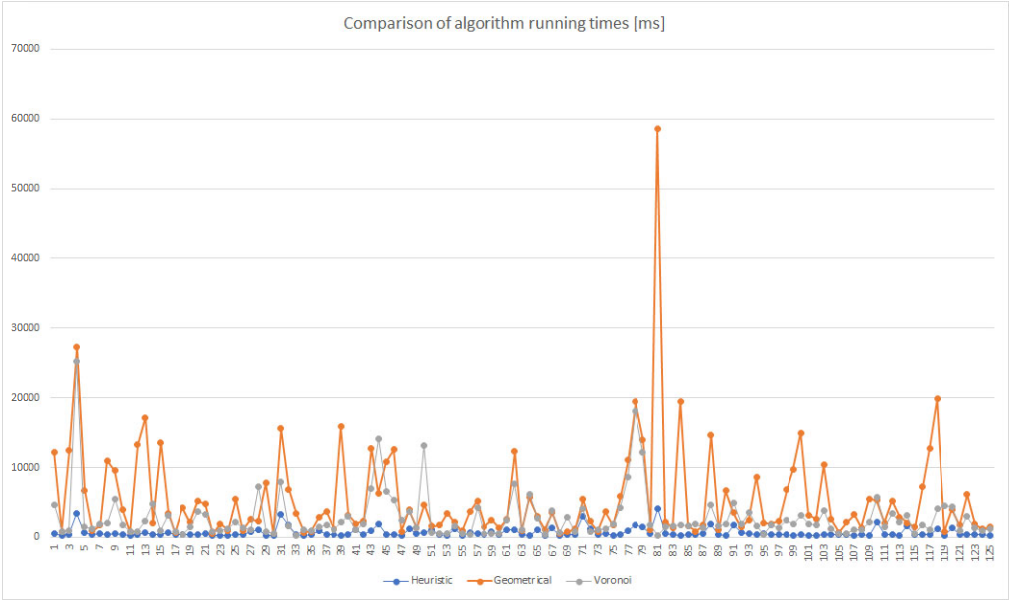


Figure 9. Comparison of running times of the algorithms over the whole dataset

ACKNOWLEDGEMENT

This work was supported by a grant of the Romanian Ministry of Research and Innovation, CCCDI - UEFISCDI, project number PN-III-P1-1.2-PCCDI-2017-0689 / „Lib2Life- Revitalizarea bibliotecilor si a patrimoniului cultural prin tehnologii avansate” / "Revitalizing Libraries and Cultural Heritage through Advanced Technologies", within PNCDI III.

REFERENCES

[1] A. Antonacopoulos, C. Clausner, C. Papadopoulos, S. Pletschacher, "ICDAR2015 Competition on Recognition of Documents with Complex Layouts – RDCL2015", Proceedings of the 13th International Conference on Document Analysis and Recognition, Nancy, France, August 2015, pp. 1151-1155.

[2] T. M. Breuel, “Document Layout Analysis", Image Understanding and Pattern Recognition Group, Technischen Universität Kaiserslautern (TUK).

[3] T. M. Breuel, “Two geometric algorithms for layout analysis”, Proceedings of the Fifth International Workshop on Document Analysis Systems, Princeton, NY, 2002, LNCS 2423, pp. 188-199.

[4] U. Kumar, J. Raheja - “Document Presentation Engine for Indian OCR - A Document Layout Analysis Application”, International Journal of Recent Trends in Engineering, (IJRTE) Vol. 3, No. 3, May 2010, pp 182-186.

[5] K. Kise, A. Sato, M. Iwata - "Segmentation of Page Images Using the Area Voronoi Diagram", Computer Vision and Image Understanding, ISSN: 1077-3142, vol. 70, issue 3, pp. 370-382.

[6] P.Babul Saheb, Mr. k.Subbarao , Dr. S.Phani kumar - "A Survey on Voting Algorithms Used In Safety Critical Systems", International Journal Of Engineering And Computer Science, ISSN: 2319-7242, vol. 2, issue 7, pp. 2272-2275

[7] LRDE Document Binarization Dataset, Available at: https:// www.lrde.epita.fr/ wiki/ Olena/ DatasetDBD, Accessed at: 1 March 2018

[8] M. Soua, A. Benchekroun, R. Kachouri, M. Akil - "Real-time text extraction based on the page layout analysis system", Proc. SPIE 10223, Real-Time Image and Video Processing 2017, 1022305, Anaheim, CA, April 2017.

[9] Costin-Anton Boiangiu, Radu Ioanitescu, Razvan-Costin Dragomir, "Voting-Based OCR System", The Proceedings of Journal ISOM, Vol. 10 No. 2 / December 2016 (Journal of Information Systems, Operations Management), pp 470-486, ISSN 1843-4711

[10] Costin-Anton Boiangiu, Mihai Simion, Vlad Lionte, Zaharescu Mihai – "Voting-Based Image Binarization" - , The Proceedings of Journal ISOM Vol. 8 No. 2 / December 2014 (Journal of Information Systems, Operations Management), pp. 343-351, ISSN 1843-4711

[11] Costin-Anton Boiangiu, Paul Boglis, Georgiana Simion, Radu Ioanitescu, "Voting-Based Layout Analysis", The Proceedings of Journal ISOM Vol. 8 No. 1 / June 2014 (Journal of Information Systems, Operations Management), pp. 39-47, ISSN 1843-4711

[12] Costin-Anton Boiangiu, Radu Ioanitescu, "Voting-Based Image Segmentation", The Proceedings of Journal ISOM Vol. 7 No. 2 / December 2013 (Journal of Information Systems, Operations Management), pp. 211-220, ISSN 1843-4711.