Real-time text extraction based on the page layout analysis system

Mahmoud Soua, Alae Benchekroun, Rostom Kachouri, Mohamed Akil

▶ **To cite this version:**

Mahmoud Soua, Alae Benchekroun, Rostom Kachouri, Mohamed Akil. Real-time text extraction based on the page layout analysis system. SPIE Conference on Real-Time Image and Video Processing, Apr 2017, Anaheim, CA, United States. 10.1117/12.2262364 . hal-01525503

HAL Id: hal-01525503
https://hal.science/hal-01525503

Submitted on 21 May 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Real-time text extraction based on the page layout analysis system

M. Soua, A. Benchekroun, R. Kachouri, and M. Akil

ESIEE Paris, IGM, A3SI, 2 Bd Blaise Pascal, BP 99, 93162 Noisy-Le-Grand, France

ABSTRACT

Several approaches were proposed in order to extract text from scanned documents. However, text extraction in heterogeneous documents stills a real challenge. Indeed, text extraction in this context is a difficult task because of the variation of the text due to the differences of sizes, styles and orientations, as well as to the complexity of the document region background. Recently, we have proposed the improved hybrid binarization based on Kmeans method (I-HBK)[5] to extract suitably the text from heterogeneous documents. In this method, the Page Layout Analysis (PLA), part of the Tesseract OCR engine, is used to identify text and image regions. Afterwards our hybrid binarization is applied separately on each kind of regions. In one side, gamma correction is employed before to process image regions. In the other side, binarization is performed directly on text regions. Then, a foreground and background color study is performed to correct inverted region colors. Finally, characters are located from the binarized regions based on the PLA algorithm. In this work, we extend the integration of the PLA algorithm within the I-HBK method. In addition, to speed up the separation of text and image step, we employ an efficient GPU acceleration. Through the performed experiments, we demonstrate the high F-measure accuracy of the PLA algorithm reaching 95% on the LRDE dataset. In addition, we illustrate the sequential and the parallel compared PLA versions. The obtained results give a speedup of 3.7x when comparing the parallel PLA implementation on GPU GTX 660 to the CPU version.

Keywords: Text extraction, Heterogeneous documents, Tesseract, Layout analysis, PLA, I-HBK, GPU

1. INTRODUCTION

Text extraction is employed in OCR systems to separate characters from the background before recognition. In particular, text extraction in heterogeneous documents is difficult because of font variation and possible complex textured background. To overcome this problem, recently, the improved hybrid binarization based on Kmeans method (I-HBK) was proposed to extract text from heterogeneous documents. This method applies an adequate text binarization algorithm on text and image regions. To get these regions, I-HBK employs the Page Layout Analysis algorithm (PLA)[1] from the OCR engine Tesseract[2] as a first step to divide the document into areas of text and non text.

The PLA algorithm[1] combines bottom-up and top-down approaches to devide suitably the document into disjoint regions. As a bottom-up method, PLA classifies pixels of the image, and gather them to get a first meaningfull text/non text regions for tab-stops finding. The key advantage of bottom-up approach is that we can handle arbitrarily shaped regions with ease such as non rectangular regions. However, they struggle to take into account higher-level structures in the image such as columns which often leads to overfragmented regions. As a top-down method,[1,2] the PLA algorithm cuts the image into column or paragraph based on tab-stops. Although this top-down method starts by looking firstly at the largest structures on the page. Actually, the bottom-up method overcomes the drawback of the top-down method namely the unablity to handle the variety of formats that occur in many magazine pages such as non-rectangular regions and cross-column headings.

The PLA algorithm is time consuming due to the computational intensive taks of morphological operations. That is why, this algorithm was accelerated on the graphic processing unit (GPU). In this paper, we show the integration of the PLA algorithm in the I-HBK text extraction method. In addition we present the GPU acceleration of the PLA algorithm. The rest of the paper is organized as follows : Section 2 gives a brief overview of I-HBK. Section 3 describes the PLA algorithm. Section 4 presents the PLA acceleration on GPU which is followed by results and conclusion, respectively, in Section 5 and Section 6.

2. THE I-HBK TEXT EXTRACTION METHOD

In scanned heterogeneous documents, extracting text from complex background is a challenging problem because characters are merged with textured background. Even if the background is homogeneous, the noise of the scan process and the uneven illumination decrease the extraction precision. To overcome these problems, we proposed recently the I-HBK method.[5] It is an adaptive text extraction that selects the appropriate binarization approach according to the region type whether it is text or image.

Figure 1 shows the I-HBK method steps. On image regions, a texture analysis approach is performed to improve images before binarization. Actually, a gamma correction process is employed to enhance the contrast of that region.[6] 100 image samples are generated from the input image, each one with a specific gamma value $\gamma \in [0.1, 10]$. From each image, four co-occurence matrices are computed to generate the energy and contrast features. In addition, for each image the Otsu threshold[3] is computed as a third feature. A selection process is performed from all generated features to get the image sample giving the optimal gamma value γ_{opt}. The input image is transformed based on γ_{opt}, thus, the contrast is enhanced.

Afterward, an efficient binarization method is performed on both improved images and text regions. This binarization consists in a hybrid approach combining a local foreground/background pixel classification based on the Kmeans algorithm[7] and a global approach that reduces the amount of distortion.

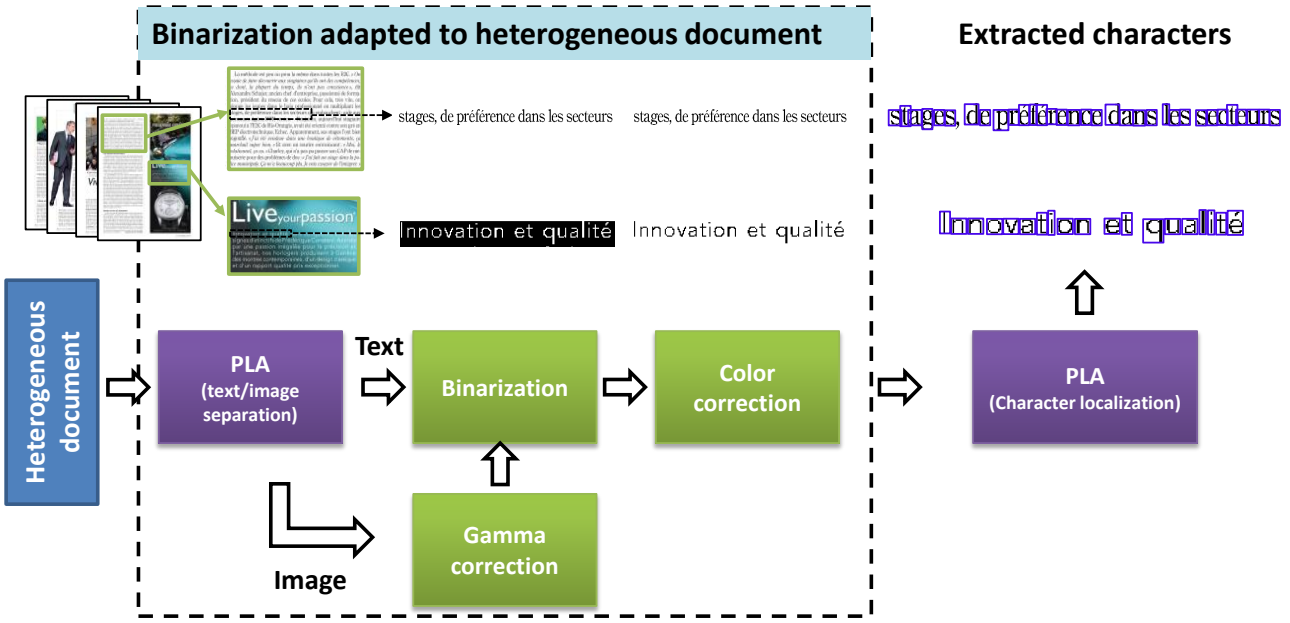

Figure 1. The Improved Hybrid Binarisation method based on Kmeans: I-HBK[5]

Then, a color correction is performed on the binarized image and text regions. The page layout analysis (PLA) algorithm is employed at the biginning of I-HBK processing to feed the binarization stage with the correct region. It includes two levels: First the document is separated into text and image regions. Next, the binary text paragraphs are separated into disjoint characters (character localization). The separation of the heterogeneous document into regions of text and images is highly complex because of the intensive treatment of the morphological operators. Folowing, we show the processing of the PLA algorithm for text extraction in heterogeneous documents.

3. THE PAGE LAYOUT ANALYSIS ALGORITHM (PLA)

Document layout analysis is the process of identifying and categorizing text/image regions in heterogeneous documents. Actually, these regions are bounded by tab-stops defined as column boundaries. The tab-stops bound body text, and also rectangular non-column elements such as images.

The well-known page layout analysis algorithm (PLA)[1] part of Tesseract[2] combines bottom-up and top-down approaches to separate the document into text and image regions. The bottom-up approach is used to look for tab-stops that mark column edges and through further combination with top-down approach, PLA copes easily with non-rectangular regions.

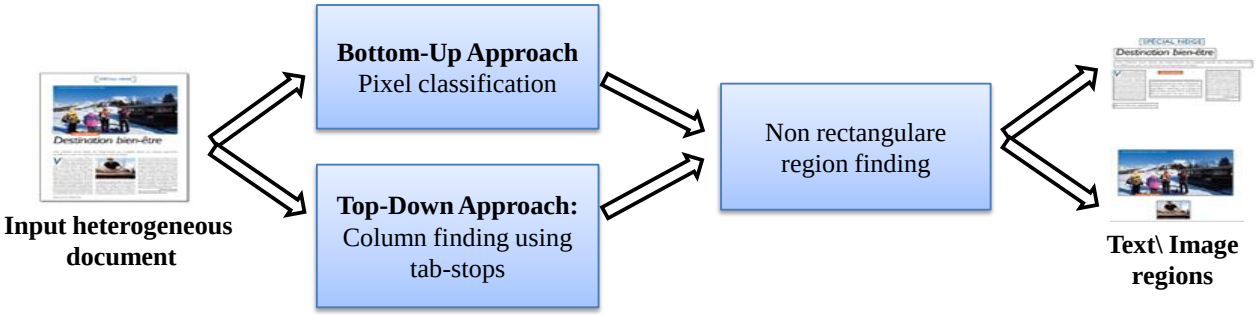

Figure 2. Page Layout Analysis for document separation into text and image regions.

As shown in figure 2, the PLA algorithm includes three main stages: *Pixel classification*, in which bottom-up morphological and connected component analysis form initial hypotheses over data types; *tab-stop detection and column layout finding*; and *region finding*. Following, these stages are explained with more details.

3.1 Pixel classification

The aim of this stage is to identify line separators, image regions, and to separate the remaining connected components into text components and a smaller number of uncertain type. Firstly, lines, image regions and connected components are identified. For this, mathematical and morphological operations from Leptonica[4] are applied on an Otsu[3] binarized document (figure 3.a) in order to detect the horizontal and vertical lines (figure 3.b) and the image mask (figure 3.c). The following mophological operation sequence is performed :

(1). Morphological closure: A dilation followed by an erosion allows to fill the small holes and link the pixels, thus, the text becomes thick;
(2). Morphological opening: An erosion followed by dilation makes a morphological opening with a large box to detect thick regions, so that they are subtracted in step (3);
(3). Substraction (1)(2);
(4). Horizontal morphological opening to find horizontal lines;
(5). Vertical morphological opening to find vertical lines.

Detected lines and image mask are subtracted from the input image before passing the cleaned image to connected component (CCs) analysis. CCs are filtered into text and non text elements.

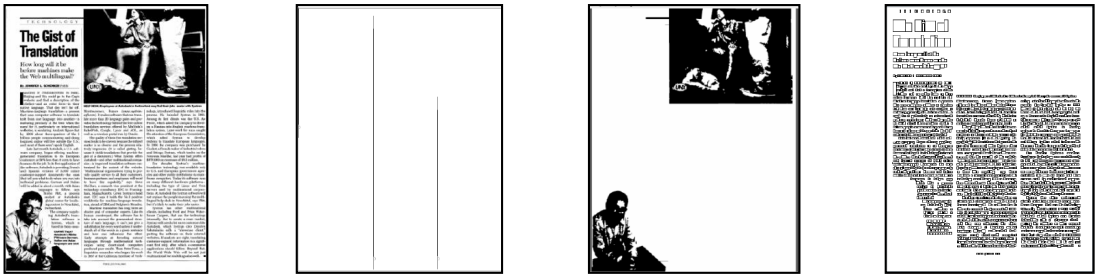

Figure 3. Pixel classification stages of the PLA algorithm.[1] (a): Input document ; (b): Detected vertical lines ; (c): Detected image regions ; (d): Filtered Connected components.

3.2 Tab-stops detection and column layout finding

Detecting tab-stops has several major sub-steps: At first, candidate tab-stop CCs are located using a radial search. Indeed, CCs situated at the edge of a text region are found and marked as right, left or not a tab-stop. Then, candidate tab-stops are grouped into lines. The connections between tab-stop lines are found, enabling removal of false positives. Once valid tab-stops are found, the connected components are gathered into column partitions (CPs) while not exceeding tab-stops. A set of CPs of one horizontal pass is called set of colum partition (CPset). Each CPset is a potential colum division on that location. A list of candidate columns is build based on optimal CP-sets. Duplicated columns are removed based on the A explain B algorithm.[1]

3.3 Region finding

After the columns are found the column partitions (CPs) are given types according to how many columns they span. CPs within a single column are «flowing», partitions that touch more than one column, but do not span to the outer edges are «pullout», and partitions that completely span more than one column are «heading». These types help to get the reading order witch is helpful in later character recognition. Text blocs are generated after merging colum partitions with close line spacing. Finally, isothetic polygons of regions are computed. Following, we present the parallelization of PLA on the graphic processing GPU.

4. PARALLELIZATION OF THE PLA METHOD ON GPU

GPUs are now common place across desktop, mobile and server platfoms. They have 10x compute capabiity compared to CPUs.[11] To exploit the GPU capabilities, the independant OpenCL parallel programming paradigm can be used.[8] Tesseract[2] processing pipeline consists of several parallelizable steps including the PLA algorithm.[1] Indeed, the page layout analysis algorithm[1] was accelerated on GPU using OpenCL.[8] The accelerated part is related to the complex line detection based on morphological closing and opening operations.

The GPU is configured with blocks of 256 threads where each thread handles in parallel 32 pixels. Considering a binary image in which each pixel is represented through a single bit, three bit-words «previous», «current» and «next» give a succession of three groups of 32 pixels in the document (Figure 4). Morphological closing and opening are carried out across horizontal and vertical phases. In the horizontal phase, the GPU is setup with a single dimension (1D) gived by the image size.

To perform horizontal dilation, the maximum values to the left then to the right of each «current» pixel are detected. The two results are summed using a logic «OR» to detect both maximums to the left and right of each «current» pixel in the image. «Current» pixels that have a neighbor equal to «1» are set to «1». The horizontal phase of erosion have the same treatment, except using logic «AND» instead of «OR» to determine the minimum on the right and left of each «current» pixel. In the vertical dilation and erosion phase, the GPU

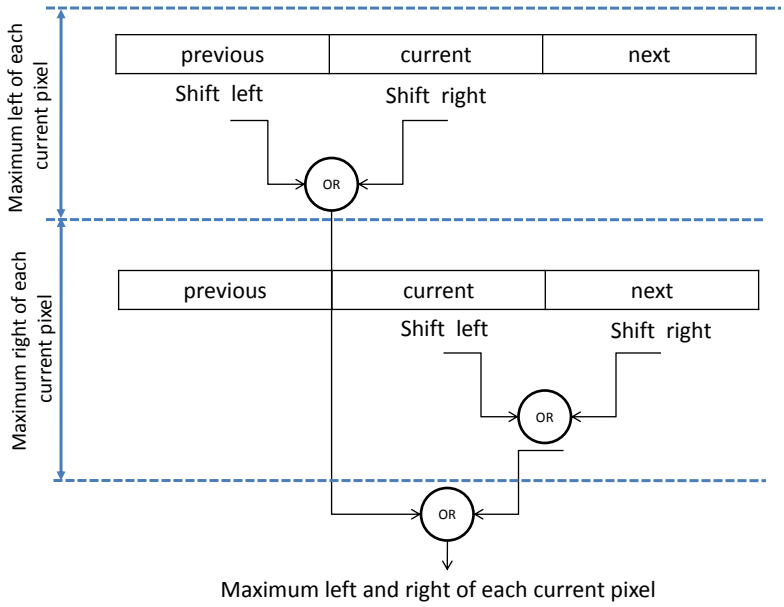

Figure 4. Page Layout Analysis implementation on the GPU.

is setup according to two dimensions (2D). The first and the second dimensions represent respectively image hight (H) and width (W) of the image. We use a logic «OR» to add the 32 current pixels with those on top and bottom to reset to «1» pixels that have a vertical neighbors equal to «1». For vertical erosion we use a logic «AND» which adds the 32 current pixels with those above and below to reset to «0» pixels with vertical neighbor equal to «0».

5. EVALUATION

In this section, we evaluate the accuracy of the PLA algorithm.[1] In addition, we compare the execution time of PLA CPU-based and GPU-based version. The LRDE* dataset[9] is used in our evaluation. It is composed of 125 heterogeneous magazine documents issued from the french magazine «Le nouvel observateur». We use a PC with a CPU processor Intel i3 with 3.07 Ghz and an Nvidia GeForce GTX 660 GPU. The OpenCL version 1.2 is employed as a GPU programming paradigm.

5.1 Page layout analysis accuracy

We study the PLA algorithm accuracy based on Precision, Recall, and F-measure metrics.[13] We present in figure 5 (a) the ground truth of text/image separation of an heterogeneous sample document using the Aletheia tool.[10] Figure 5 (b) illustrates text/image separation using the PLA algorithm.[1] Actually we are giving Aletheia[10] two XML files as inputs including region positions for both ground truth and PLA segmentation.

The PLA segmentation is close to the ground truth one. However PLA can not detect the first paragraph correctly due to the lettrine introducing noise during the separation process. In addition in the page bottom, the PLA algorithm fuses image and text on a red colored background. The F-measure reaches high accuracy of 96% on the shown document (figure 5). It reaches an average of 95% on 125 LRDE heterogeneous documents.

*Copyright(c) 2012. EPITA and Development Laboratory (LRDE) with permission from Le Nouvel Observateur. LRDE-DBD is available online on the web site:
http://www.lrde.epita.fr/cgi-bin/twiki/view/Olena/DatasetDBD

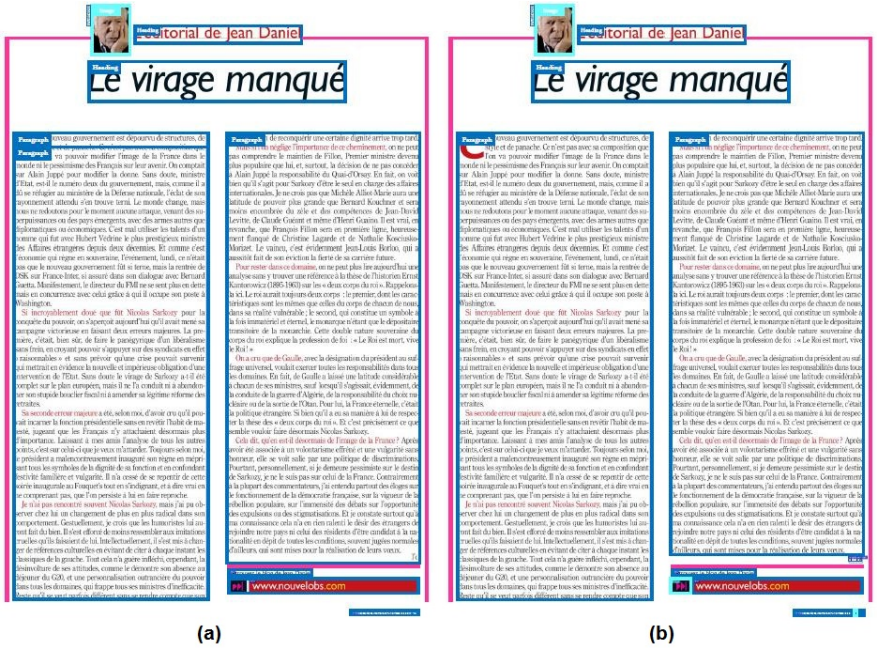

Figure 5. Layout analysis evaluation (Aletheia tool): (a) : Groundtruth ; (b) PLA segmentation

5.2 PLA GPU-based speedup

In the following evaluation, the performances are measured based on time execution computed after memory transfert between CPU and GPU. We show in figure 6 that the GPU and CPU version of the PLA algorithm give the same layout analysis result without any quality loss. Indeed, recall, precision and F-measure are equal to 100%.

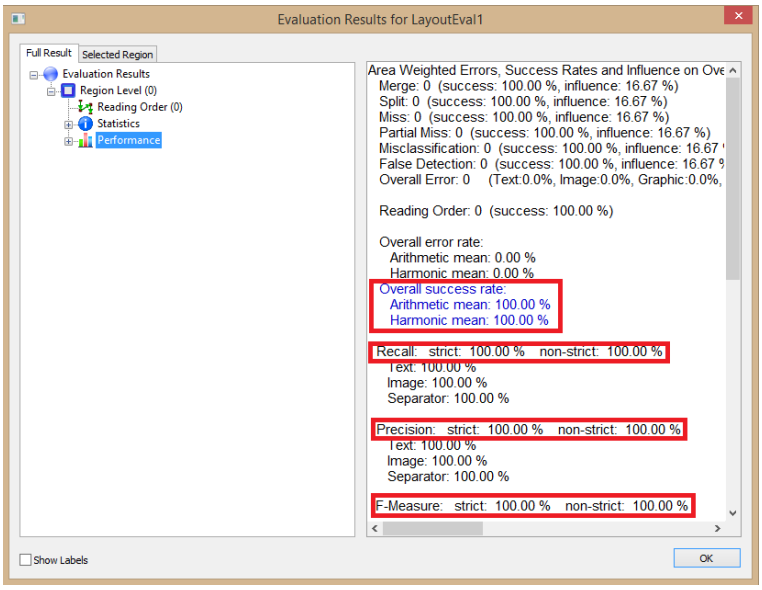

Figure 6. Layout analysis accuracy based of CPU and GPU version.

Following, we measure the execution time of PLA on both CPU and GPU (see figure 7). The average time processing of PLA on CPU is 1.61 seconds. However in GPU it reaches only 0.43 seconds. Indeed PLA is accelerated 3.7x on the GPU compared to the CPU version.

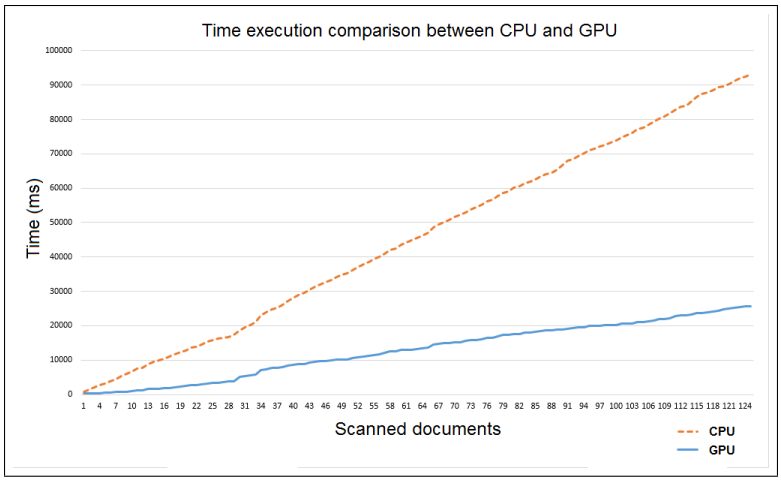

Figure 7. Execution time of PLA on both CPU and GPU on 125 heterogeneous documents from the LRDE dataset.

Figure 8 shows a comparaison between CPU and parallelized processes on GPU in Tesseract[2] including preprocessing : RGB to gray conversion, Histogram computation, Otsu thresholding and morphology operations (part of the PLA algorithm). We see that the acceleration of PLA is one of the highest scores (15x) in addition to the thresholding process that was done in the tesseract engine.[2]

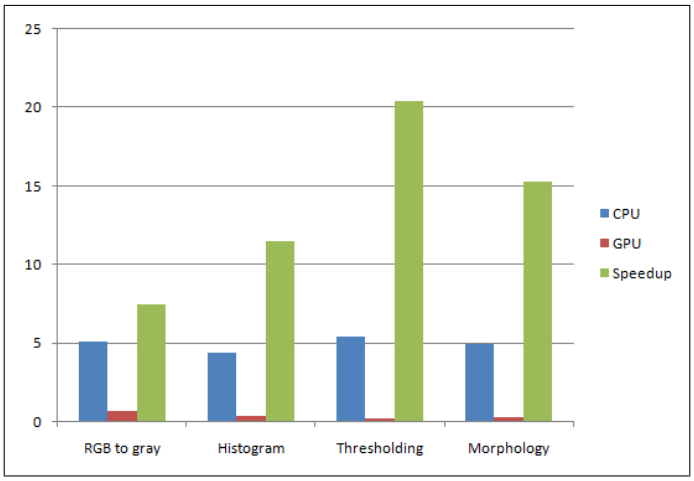

Figure 8. Speedup comparison between the morphology process - part of the PLA algorithm and several preprocessing of Tesseract. Time is shown in seconds on 6 pages color business letter on AMD A10-7850K Kaveri machine with OpenCL 1.2.[11]

Finally, we compare the execution time of the accelerated I-HBK method on the GPU. We note that the CPU-based I-HBK method scores 9 seconds to process 125 hetoregeneous documents from the LRDE dataset on CPU. The GPU version of I-HBK scores only 7.39 secondes when including the accelerated version of PLA, giving a final speedup of 1.21x.

6. CONCLUSION

Text extraction in heterogeneous documents is a difficult task because of the variation of the text due to the differences of sizes, styles and orientations, as well as to the complexity of the background. Recently, we have proposed the improved hybrid binarization based on Kmeans method (I-HBK) to extract suitably the text from heterogeneous documents. In this method, the Page Layout Analysis (PLA) - part of the Tesseract OCR engine

- is used to identify text and image regions. In this work we presented the work of the PLA algorithm inside the I-HBK method. In addition, to speed up the separation of text and image step, we employ an efficient GPU acceleration. Through the performed experiments, we demonstrate the high F-measure accuracy of the PLA algorithm reaching 95% on the LRDE dataset. The obtained results give a speedup of 3.7x when comparing the parallel PLA implementation on GPU GTX 660 to the CPU version. In addition the I-HBK method was accelerated 1.21x when employing the GPU-based PLA algorithm.

REFERENCES

[1] R. Smith, Hybrid Page Layout Analysis via Tab-Stop Detection, ICDAR '09 Proceedings of the 2009 10th International Conference on Document Analysis and Recognition, Pages 241-245, 2009

[2] R. Smith, An Overview of the Tesseract OCR Engine, Proc. Ninth Int. Conference on Document Analysis and Recognition (ICDAR), IEEE Computer Society, 629-633 , 2007.

[3] N. Otsu, A threshold selection method from gray-level histograms, IEEE Transactions on Systems, Man and Cybernetics, 9(1): 62-66, 1979.

[4] http://www.leptonica.com/

[5] Mahmoud Soua, Rostom Kachouri, Mohamed Akil. Improved Hybrid Binarization based on Kmeans for Heterogeneous document processing. 9th International Symposium on Image and Signal Processing and Analysis, ISPA'15, Sep 2015,

[6] C. P. Sumathi and G. Gayathri Devi, Automatic Text Extraction From Complex Colored Images Using Gamma Correction Method, Journal of Computer, Volume 10, Issue 4 Science, Pages 705-715, 2014.

[7] S. P. LLOYD, *"Least square quantization in PCM"*, IEEE Transactions on Information Theory vol.28, no.2, 129-137, 1982.

[8] Khronos. OpenCL: The open standard for parallel programming of heterogeneous systems, http://www.khronos.org/opencl/

[9] The SCRIBO Module of the Olena Platform: a Free Software Framework for Document Image Analysis. In the proceedings of the 11th International Conference on Document Analysis and Recognition (ICDAR), 2011.

[10] E. Saund, Jing Lin, P. Sarkar, "PixLabeler : User Interface for Pixel-Level Labeling of Elements in Document Images", Proc. of the 10thInternational Conference on Document Analysis and Recognition (ICDAR2009), Barcelona, Spain, July 26-29, 2009 pp.446-450.

[11] Ray Smith, Slides from Tesseract tutorial, Google Inc, at DAS, Santorini, 2016.

[12] Nagy, G., Kanai, J., Krishnamoorthy, M., Thomas, M. et Viswanathan, M. (1988). Two complementary techniques for digitized document analysis. In DOCPROCS : Proceedings of the ACM conference on Document processing systems, pages 169-176, New York, NY, USA. ACM.

[13] Van Rijsbergen, C.J. Foundation of evaluation. Journal of Documentation, 30(4):365–373, 1974.