Text Detection in Chart Images[1]

N. Vassilieva[a] and Y. Fomina[b]

[a] *HP Labs, 1 Artillerijskaya str., St. Petersburg, 191104, Russia*
[b] *Studio Mobile, 18A Bolshoy Prospekt, St. Petersburg, 197198, Russia*
e-mail: nvassilieva@hp.com; yu.fomina@gmail.com

Abstract—Common OCR (Optical Character Recognition) systems fail to detect and recognize small text strings of few characters, in particular when a text line is not horizontal. Such text regions are typical for chart images. In this paper we present an algorithm that is able to detect small text regions regardless of string orientation and font size or style. We propose to use this algorithm as a preprocessing step for text recognition with a common OCR engine. According to our experimental results, one can get up to 20 times better text recognition rate, and 15 times higher text recognition precision when the proposed algorithm is used to detect text location, size and orientation, before using an OCR system. Experiments have been performed on a benchmark set of 1000 chart images created with the XML/SWF Chart tool, which contain about 14000 text regions in total.

Keywords: text detection, optical character recognition, chart images, connected components analysis, Hough transform.

DOI: 10.1134/S1054661813010112

INTRODUCTION

Visual data is important in many applications. It often contains highly valuable information and can provide essential knowledge that is not duplicated in other data formats. But currently visual data is not sufficiently exploited for indexing and knowledge extraction purposes.

Graphical illustrations, diagrams, charts and graphs are main types of visual data in financial and scientific domains. For efficient indexing, processing and analysis of documents in these domains, one needs an automated computer-based system which is capable of providing analysis and semantic interpretation of graphics. An important step here is to detect, extract and recognize text present in graphical illustrations within the document. Textual data in charts and graphs includes axes names, tick mark labels, legends, captions, notes. It carries information, which is important for humans to understand the illustration, and for automated data analysis systems to extract knowledge and facts from the documents.

Common optical character recognition engines, such as Tesseract OCR [9], ABBYY FineReader [10], MODI (Microsoft Office Document Imaging) [11], are tuned to detect and recognize the paragraphs of text in scanned document images and fail to detect small text regions composed of one or a few words. Such text regions are typical for chart images. The accuracy of OCR systems on scanned document images can exceed 99% nowadays [6]. In particular, the recognition accuracy of Tesseract OCR is up to 97% when applied to scanned document images [7, 8, 9]. This OCR engine is considered one of the most accurate free software OCR engines currently available. At the same time the recognition accuracy of this engine doesn't exceed 3% for chart images (according to our experimental results).

Existing OCR engines fail to detect text regions in chart images due to their small size and often non-horizontal orientation. But being detected, cropped and deskewed manually or by an additional preprocessing algorithm, even small text regions are well recognized by existing OCR engines.

In this paper we propose a novel algorithm for detecting text in chart images, which can be used at the preprocessing step in order to detect and localize text regions for subsequent recognition.

Text detection is performed in three steps. First, potential characters are detected by connected components labeling and strong non-character components are excluded by heuristics-based filters. Second, the Hough Transform [12] is used to detect text lines and identify their orientations. And third, inter-character distance and character size is analyzed for final localization of text regions.

This algorithm is used as a preprocessing step for further text recognition. The detected text regions are passed to an OCR engine after being cropped and deskewed. We used Tesseract OCR engine [9] in our experiments to evaluate the effectiveness of the proposed solution. The experimental results show a significant increase of the recognition rate when the pro-

[1] The article was translated by the authors.

Received March, 12, 2011

posed algorithm for text detection and localization is used as a preprocessing step.

RELATED WORK

The majority of existing algorithms for text recognition are designed for processing scanned document images. There are also algorithms for text recognition in arbitrary natural scene images. Both the former and the latter usually consist of the similar steps:

(1) image binarization (local, global or adaptive) [2, 7],

(2) text detection and localization (region-based, edge-based or texture-based) [5],

(3) text line detection [2, 7, 9],

(4) character extraction and image enhancement,

(5) character recognition [9].

However, implementation of every step of a particular algorithm exploits image properties of the target class of images for this algorithm.

For document images, the following observations are usually true: all text lines are parallel; text size is uniform; it is easy to separate the background and foreground; large text blocks of several text lines (paragraphs of text) are present [1, 2, 9]. Hence, text detection and localization techniques for such documents are based on the properties of a paragraph of text (for example, horizontal orientation of text lines) rather than separate words. Therefore, text detection algorithms initially designed for scanned document images do not work well on chart images.

For arbitrary natural scene images containing text regions the following observations are usually true: there are few text regions in an image; text regions are small and contain one or few words; background is often complex; text has lower contrast compared to the text in document images [3, 5]. The complexity of text recognition algorithms for natural scenes is higher compared to the complexity of algorithms for document images, while the accuracy is significantly lower. It is not reasonable to apply these algorithms for text recognition in chart images. Chart images have a number of specific features, which are not common for natural scenes and they can be used in order to increase text recognition accuracy.

Chart images belong to none of these classes. They are somewhere in between (text strings are small, but have a good contrast) and thus require an individual approach. Small text regions are typical for chart images as they are for natural scenes. At the same time text regions in chart images usually have high contrast and uniform background as in document images. Thus the problem of text detection in chart images requires an individual approach.

A work by Fletcher and Kasturi is the closest one to our solution [4]. They proposed an algorithm for text string separation for engineering drawings and diagrams. This algorithm might be applied to solve text detection problem in chart images, but it sets a lot of restrictions for an input document. It should be binary and of high resolution. There are constraints on acceptable font size and intercharacter distance. Our solution doesn't impose these limitations. It contains similar steps, but the implementation of every step differs from those proposed in [4].

PROBLEM STATEMENT

Our goal is to develop an algorithm for text detection in chart images. It should automatically detect location, orientation and size of every text region in a chart image.

THE ALGORITHM OVERVIEW

The algorithm for text region detection operates regardless of text orientation and font size or style. The algorithm uses simple heuristics based on the following observations about text regions in chart images.

(1) Geometry: (a) charts can contain textual entries of various font families and font sizes; (b) text lines can have any orientation, the same chart can contain textual entries of different orientations; (c) inter-character distance can vary from one text region to another, but it is usually uniform within the same region.

(2) Size: (a) text regions in charts are often small, contains few characters only in one text line.

(3) Contrast: (a) most chart images have good contrast and strong edges at the boundaries of all chart elements; (b) most chart images are synthetically generated and thus are noise-free.

Our approach to text detection includes the following steps:

(I) character detection using the connected components analysis;

(II) text line detection using the Hough Transform;

(III) grouping characters into words and phrases using the inter-character distance analysis.

We describe these steps in greater detail below.

STEP I: CONNECTED COMPONENTS ANALYSIS

A two-stage bottom-up approach is used to identify the connected components (CCs) in the image. First, pixels which are eight-connected to one another and have similar intensities and colors are grouped together. Second, small CCs with mean color and intensity similar to the color and intensity of a neighboring CC are joined to this neighbor component. The difference of intensities and Euclid distance in RGB space are used to measure the similarity of intensities and colors. Each identified CC is then either rejected or accepted as a character based on its attributes (size, position within the image, the density of pixels belong-

ing to the CC within the enclosing rectangle, ratio of dimensions, area etc.).

Figure 1 shows an example of a chart image. Figure 2 presents a fragment of this image and the results of the first and second stages of CCs labeling.

The result of the first step of the proposed text detection algorithm is a set of CCs (potential characters) and their bounding boxes. The bounding boxes corresponding to the CCs detected in the chart image presented in Fig. 1 are shown in Fig. 3.

STEP II: TEXT LINES DETECTION

To identify text lines and their orientations from a set of potential character components detected in the previous step, we apply the Hough Transform to the points at the centers of bounding boxes. We consider all lines which connect three or more points. As a result, many false text lines are detected (Fig. 4). Rule-based filtering is then applied to exclude the false text lines from further processing. The rules are based on the following properties of a line: the density of points along the line, the number of parallel and orthogonal lines to a given one, the fraction of points being shared with another line.

STEP III: GROUPING OF CHARACTERS

In the third step the distribution of points along the found lines is examined in order to group the characters into words and phrases. The ratio between the intercomponent distance along the line and the size of components in orthogonal direction is used to determine component grouping. The neighbor bounding boxes which satisfy the decision rule based on the above-mentioned ratio are merged together to form a text region.

EXPERIMENTAL EVALUATION

In order to evaluate the performance of the proposed text detection algorithm we have performed a number of experiments comparing the recognition accuracy of Tesseract OCR with and without prior text detection by the proposed algorithm.

For experimental evaluation, we created a test set of 1000 chart images using XML/SWF Charts tool[2]. Every generated chart image has a corresponding XML mark-up file. It contains the ground truth information about every text region in the image, such as location and orientation of the text, visual appearance properties (font family, font size, background and foreground colors, and others) and the text itself. The test set contains in total about 14000 text regions. The collection contains charts of different types: pie diagrams, 3D pie diagrams, column diagrams, area diagrams, line diagrams and

[2] http://www.maani.us/xml_charts.

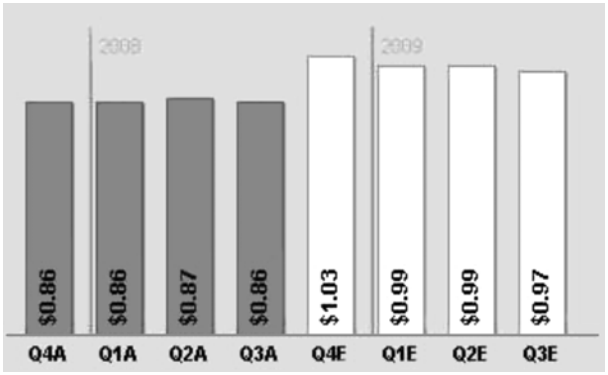

Fig. 1. A chart image

Fig. 2. From left to right: a fragment of the original chart image in Fig. 1, the results of the first and second stages of CCs labeling for the given fragment.

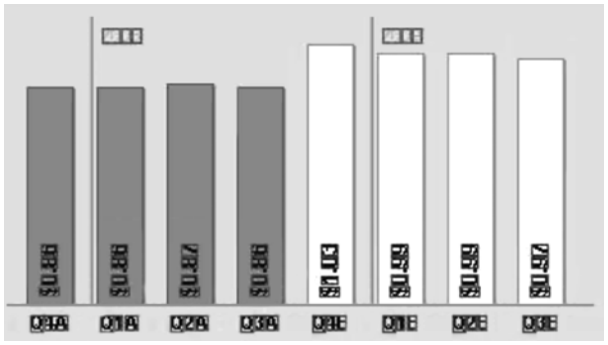

Fig. 3. The bounding boxes of the identified character components for the chart image in Fig. 1.

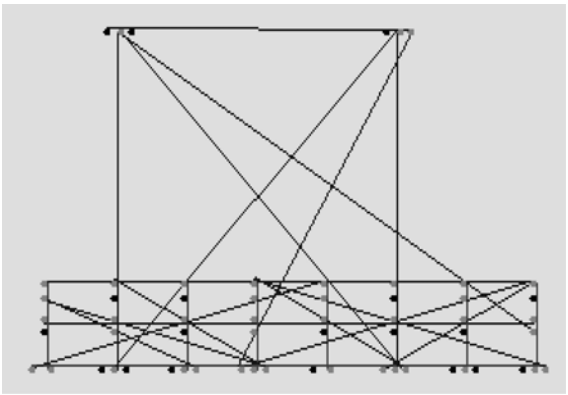

Fig. 4. The detected text lines before filtering: the result of the Hough Transform for the centers of the bounding boxes in Fig. 3

Text detection and recognition results

	LRR	LPR	TRR	TPR
Prep	79.0%	88.7%	45.0%	44.6%
NoPrep	33.6%	84.0%	2.5%	2.9%

mixed diagrams (combinations of the last three types). The following parameters were varied during generation of the chart images: diagram type, text color, transparency and orientation, background color (for some areas), font, number of labels on axes and dictionaries of words and phrases to put in text regions.

We performed several experiments to evaluate the algorithm. The goal of the first experiment was to compare the accuracy of text recognition by Tesseract OCR engine with and without prior text detection by the proposed algorithm. In other experiments we analyzed the dependency of the text recognition accuracy on the text parameters such as text orientation, color, font, size, and others. Two runs were done for each experiment. In the first run, the test collection was preprocessed by the proposed algorithm to localize text regions, and then the detected text regions were passed to Tesseract OCR engine for text recognition. In the second run, the test images were processed by Tesseract OCR directly without prior text detection by the proposed algorithm. We will further refer to these runs as **Prep**—with preprocessing, and **NoPrep**—without preprocessing, respectively.

We used the following metrics to evaluate the performance of the algorithm in the experiments:

(1) LRR, location recognition rate: $LRR = N_{Loc}/NG$;

(2) LPR, location precision rate: $LPR = N_{Loc}/NF$;

(3) TRR, text recognition rate: $TRR = N_{Txt}/NG$;

(4) TPR, text precision rate: $TPR = N_{Txt}/NF$;

where N_{Loc} is the number of correctly localized text regions, N_{Txt} is the number of correctly recognized text regions, NG is the true total number of text regions in the test set, NF is the total number of detected text regions. The metrics values were computed on a ground truth basis by using xml mark-up files. The text regions were considered to be correctly localized if the intersection with one of the ground truth text regions covered more than 90% of its area; correctly recognized—when the Levenshtein distance between the ground truth text and the recognized one was less than 3.

The Table 1 below presents the experimental results.

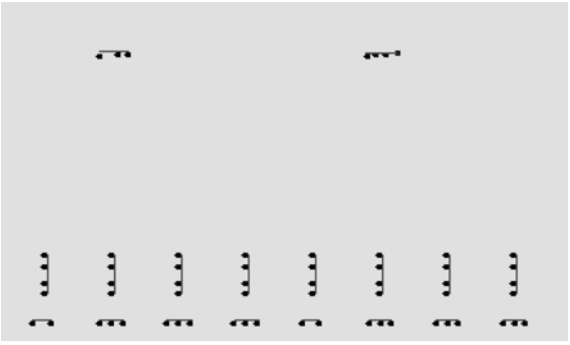

Fig. 5. The detected text lines after excluding false text lines in Fig. 4.

TEXT DETECTION AND RECOGNITION RESULTS

The experimental results shows that when the proposed algorithm is used as a preprocessing step before OCR, one can get up to 20 times better text recognition rate, and 15 times higher text recognition precision.

Besides evaluation of the overall effectiveness of the proposed approach, we have also evaluated the influence of various text parameters on the accuracy of text detection and recognition. With the help of Weka package [13] we have picked out those parameters, which influence the text detection and recognition accuracy at most. The most influential parameters turned to be text line orientation and font size. We have generated another test set of 1000 images for a more detailed analysis of the influence of these two parameters. Text regions in this test set differ by text line orientation and font size only. Other parameters are the same for all text regions in all charts: text color = 000000, font = Arial, transparency = 100; background color = FFFFFF.

We have divided the entire test collection of text regions into several groups by their text line orientation in order to evaluate the influence of this parameter on the accuracy of text detection and recognition. We have considered the following groups:

- horizontal text (text line angle is 0°);
- near-horizontal text (text line angle is within the intervals [−10°, 0°) and (0°, 10°];
- bottom-up diagonal text (text line angle is 45°);
- top-down diagonal text (text line angle is –45°);
- bottom-up vertical text (text line angle is 90°);
- top-down vertical text (text line angle is –90°);
- all angles—the union of all above mentioned groups (the entire test collection).

The text recognition rate was calculated for every group as the ratio of correctly recognized text regions within this group (with a particular text line orientation) to the total number of text regions in a given group (the total number of text regions with this orientation in the entire test collection). The dependence of the text recognition rate on the text line orientation is shown in Fig. 8. One can see the increase of the text recognition rate for all text orientations when the proposed algorithm is used for text detection and localiza-

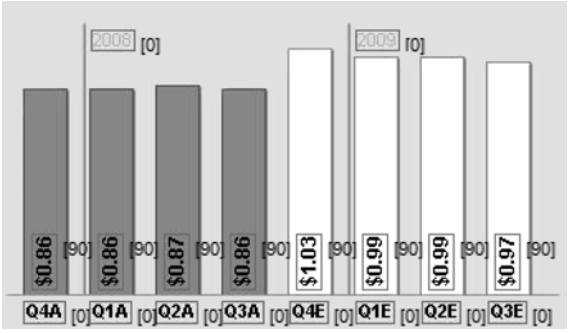

Fig. 6. The detected text regions and their orientations.

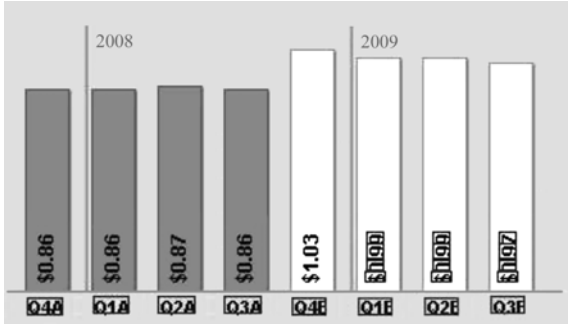

Fig. 7. The text regions detected by Tesseract OCR engine. All text regions are detected as horizontal text.

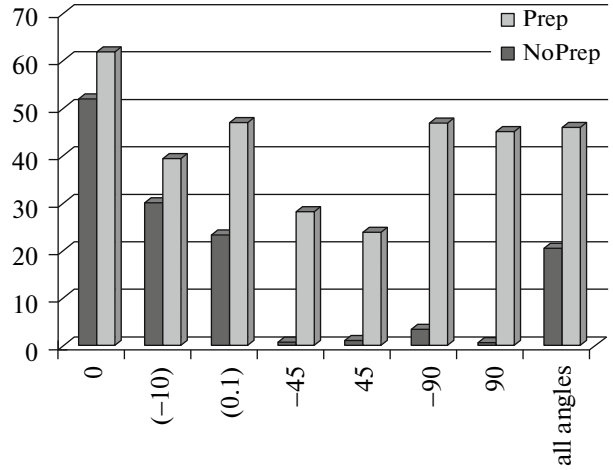

Fig. 8. The dependence of the text recognition rate (in percents) on the text line orientation.

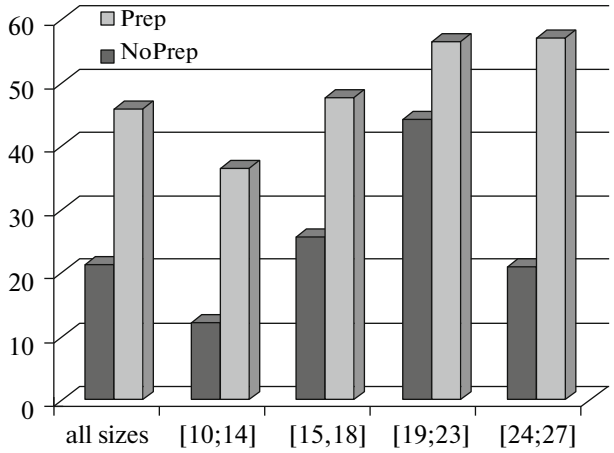

Fig. 9. The dependence of the text recognition rate (in percents) on the font size.

tion. The most significant gain in text recognition rate is obtained for non-horizontal texts.

We have also evaluated the dependence of the recognition rate on the font size in a similar way. We have divided the test collection into the following groups of text regions by their font size:

- font size is within [10pt, 14pt];
- font size is within [15pt, 18pt];
- font size is within [19pt, 23pt];
- font size is within [24pt, 27pt];
- all sizes—the union of all above mentioned groups (the entire test collection).

We have calculated the recognition rate for every group as a ratio of correctly recognized text regions within this group (with a particular font size) to the total number of text regions in a given group (the total number of text regions of this font size in the entire test collection). Figure 9 shows the dependence of the text recognition rate on the font size. The highest text recognition rate for NoPrep runs (without preprocessing for text detection and localization) is observed for text regions with the font size within the interval of [19pt, 23pt]. Adding the preprocessing step for this group increases the text recognition rate by 12%. The gain in recognition rate for other groups is even more significant.

The results of the experiments show that the use of the proposed algorithm as a pre-processing step before OCR not only leads to a considerable increase in text recognition rate but also decreases the influence of such parameters as text line orientation and font size on recognition accuracy.

CONCLUSION

In this paper a new algorithm for text detection in chart images is proposed. It significantly increases text recognition accuracy when the algorithm is used as a preprocessing step for text localization prior to OCR. Text recognition is a necessary step to semantic indexing and knowledge extraction from chart images. The experimental results prove the competitiveness of the proposed solution compared to text detection algorithms in existing OCR systems. Further increase of the recognition rate might be achieved by tuning the CCs detection step and by applying the Hough Transform locally for detection of text lines orientations.

REFERENCES

1. A. Amin and S. Fischer, "A Document Skew Detection Method Using the Hough Transform," Pattern Anal. Appl. **3**, 243–253 (2000).
2. W. Bieniecki, S. Grabowski, and W. Rosenberg, "Image Preprocessing for Improving OCR Accuracy," in *Proc. Int. Conf. Perspective Technologies in MEMS Design* (Lvov, 2007), pp. 75–80.
3. D. Chen, J.-M. Odobez, and H. Bourlard, "Text Detection and Recognition in Images and Video Frames," Pattern Recogn. **37**, 595–608 (2004).
4. L. A. Fletcher and R. Kasturi, "A Robust Algorithm for Text String Separation from Mixed Text/Graphics Images," IEEE Trans. Pattern Anal. Mach. Intelligence **10** (6), 910–918 (1988).
5. K. Jung, K. Kim, and A. Jain, "Text Information Extraction in Images and Video: a Survey," Pattern Recogn. **37** (5) 977–997 (2004).
6. Optical Character Recognition. http://en.wikipedia.org/wiki/Optical_character_recognition
7. M. Pilu, "A Lightweight Text Processing Pipeline for PDAs and Embedded Cameras," HPL Tech. Rep. no. HPL-2002-8 (2002).
8. S. V. Rice, F. R. Jenkins, and T. A. Nartker, "The Fifth Annual Test of OCR Accuracy," Tech. Rep. no. 96-01 (Information Science Research Institute, University of Nevada, Las Vegas, Apr. 1996).
9. R. Smith, "An Overview of the Tesseract OCR Engine," in *Proc. 9th Int. Conference on Document Analysis and Recognition* (Curitiba, 2007), Vol. 2, pp. 629–633.
10. ABBYY Fine Reader. http://finereader.abbyy.com/
11. Microsoft Office Document Imaging. http://en.wikipedia.org/wiki/Microsoft_Office_Document_Imaging
12. R. O. Duda and P. E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures," Commun. ACM **15**, 11–15 (1972).
13. Weka: Data Mining Software in Java. http://www.cs.waikato.ac.nz/ml/weka/

Natalia Vassilieva is a senior research scientist at HP Labs. She graduated with honor from Saint Petersburg State University (SPSU), Mathematics and Mechanics Department, Chair of Software Engineering in 2002; held an intern position in the Multimedia Information Modeling and Retrieval group at the research laboratory CLIPS-IMAG (Grenoble, France) in 2002-2003; obtained PhD in Computer Science from SPSU in 2010; taught at SPSU and worked as a software developer for several IT companies before joining HP Labs in 2007. She is a recipient of a number of awards and scholarships for scientific projects in the area of image analysis, an author of more than 20 papers, an active member of Russian IR community, an organizer of ROMIP image tracks. Her research interests include image analysis, information retrieval, information extraction, machine learning.

Julia Fomina has been a developer of mobile applications in "Studio Mobile" company since 2011. She graduated with honors from the Mathematics and Mechanics Faculty of St. Petersburg State University in 2011. She took part in a project on image comparison in a summer school organized by "Lanit–Tercom" in 2008. In 2009–2010 she participated in the development of remote control mobile services. In 2009–2010 she was an intern at Hewlett-Packard Laboratories and participated in projects on image analysis and information extraction from images. Research interests: image similarity, content-based image retrieval.