Three-Stage Method of Text Region Extraction from Diagram Raster Images

Jerzy Sas and Andrzej Zolnierek

Abstract. In the paper the combined approach to the problem of text region recognition problem is presented. We focused our attention on the chosen case of text extraction problem from specific type of images where text is imposed over graphical layer of vector images (charts, diagrams, etc.). For such images we proposed three-stage method using OCR tools as some kind of feed-back in process of text region searching. Some experimental results and examples of practical applications of recognition method are also briefly described.

1 Introduction

Although *optical character recognition* (OCR) is a mature and widely used technique as far as clear text image is being processed, its performance is still low in case of text printed over textured background or the text is an element of the image consisting of many graphical elements not being text characters. In such cases, before the image is passed to OCR engine, the text must be extracted from background or the graphical elements not constituting the text must be removed. In this work we consider the specific case of text extraction form specific type of images where text is imposed over graphical layer of vector images. Typical examples of category of images are: plots, UML diagrams, organization charts, flow charts, electric schematic diagrams etc. Many works have been described in literature aimed on

Jerzy Sas
Wroclaw University of Technology, Institute of Applied Informatics,
Wyb. Wyspianskiego 27, 50-370 Wroclaw, Poland
e-mail: jerzy.sas@pwr.wroc.pl

Andrzej Zolnierek
Wroclaw University of Technology, Faculty of Electronics, Department of Systems and Computer Networks, Wyb. Wyspianskiego 27, 50-370 Wroclaw, Poland
e-mail: andrzej.zolnierek@pwr.wroc.pl

R. Burduk et al. (Eds.): *CORES 2013*, AISC 226, pp. 527–538.
DOI: 10.1007/978-3-319-00969-8_52

text extraction from real world pictures ([3]), where text appears over the complex background. In comparison with such kind of problems, text extraction from diagram-like images seems to be relatively easy. The experiments made with diagram images showed however that methods elaborated for text extraction from real word images, exhibit tendency to detect big number of "false positive" areas, i.e. the image fragments not being actually text areas are detected. The method presented in this article is an extension of approaches known from the literature, where multi-pass approach is used ([3], [4]). The first pass selects candidate fragments, possibly containing text, using edge-density based methods borrowed from the literature ([8]). Typically, it selects much more fragments that is finally expected. Then two subsequent passes refine this selection by applying more restrictive criteria based on geometric features of the fragment contents, specific for text images. The last selection stage uses results of OCR recognition ([2]). The relation of recognized text to the area aspect ratio as well as the analysis of the recognized string is used in order to further restrict the set of text area candidates. On the other hand in ([7]) connected-component-based character locating method is presented. Then in this method we can find several stages including color layers analysis, an aligning-and-merging-analysis and OCR applications. Another attempt is presented in ([1]) where mathematical morphology is used in text extraction before standard OCR tool is applied. In all above mentioned papers during the process of text region findind, before OCR module is applied some heuristic methods are proposed. Almost the same methodology is proposed here, but the novelty of this paper consists in field of application (charts, diagrams etc.) where appropriate algorithms should be applied. Moreover we propose using OCR module as some kind of feed-back in process of text region extraction.

The work described here is a part of wider project aimed on knowledge acquisition from electronic sources. Text extraction from images is used there a)for support of automatic images annotation (i.e. labeling images with keywords), b)finding documents which contents may help in answering user queries, c)extracting knowledge from images. Separation of text from remaining graphical elements of the image also makes easier the process of recognizing elements appearing on the image and finding determining relations between them. In order to achieve the last aim, it is necessary to precisely separate geometrical elements of the image (in the case of diagrams being considered here - line segments obtained as the result of the image vectorization) for text in the situation where graphics and text touch each other. This topic is also mentioned in this article.

The organization of the paper is as follows: after introduction the problem statement is presented in second section. Next, the description of three-stage method of text region extraction is presented. Every stage of our complex method is presented in subsequently subsections containg as well very well-known algorithms as well new ideas of this work. In the section 3 the results of empirical investigation are presented and in the end we conclude this paper.

2 Extraction Method of Region with Text

As stated in the introduction, the text extraction form the image is carried out for the sake of analysis of documents containing images. The complete electronic document (either scanned and stored as complete image, or structured, e.g. described in HTML) is first subdivided into structural elements (chapters, chapter titles, sections, paragraphs). The document analysis extracts images embedded in the document and stores them as raster images. These images are input to the text extraction procedure described here. Let us consider a digital raster image automatically extracted from the complete documents image containing some geometrical shapes (line segments, polygons, ellipses, vectorized symbols specific to diagram type etc.) and possibly containing areas where text is located. Text areas can be included inside the geometrical shapes, can be located out of them or can intersect edges of geometrical shapes. Our aim is to split the image into two layers: graphical and textual. The graphical layer contains all geometrical shapes not being elements of text. The textual one contains only elements belonging to texts included in the image. For the sake of OCR processing and in order to bind text to graphical shapes it is necessary to gather characters in the textual layer into *text regions*. The text region is the rectangular area tightly enclosing characters constituting the text string.

In the previous work the method of text regions extraction using texture features (local stochastic properties of visual features in pixels of the image) based on the algorithm described in ([8]) was applied. However, this method was excessively finding the region, which did not contain any text. In this work the methods based on texture properties are also used but they are augmented with additional features calculated for candidate regions which are rectangle covering the connected component. These features include the texture properties (in the form of a map like in ([4])) and the shape characteristics.

For the sake of further application in document analysis, the text extraction procedure creates the following artifacts:

- two binary images containing separated graphical and textual layer,
- the set of images corresponding to individual extracted text regions in two versions: as a binary image with white background and black foreground and as a simple fragment of the gray scale input image (including background),
- XML file describing the extracted text regions (position in the image, size, recognized text,link to the text region image).

The method proposed in this paper consists of three stages:

- first the set of *candidate regions* (which usually apart from the correct regions contains also the regions without the text) is determined and for such regions the feature vectors for pattern recognition task are calculated,
- in the second stage the candidate regions are classified as text regions or regions without the texts using the classifier (in which the above mentioned feature vector is used) based on simple decision tree, the regions recognized as not containing text are rejected from further considerations,

- on the last stage the elements classified as text region are recognized using standard OCR tool and the final decision about the region of interest is made using the likelihood of OCR decisions.

2.1 Determining of Candidate Regions

In our method it is assumed that texts appearing on the image are horizontally oriented. In order to extract vertically oriented texts proposed method is first applied to the original image. Then identified text regions are replaced by background color and the modified image is rotated by 90 degrees. Next the text extraction procedure is applied again to the rotated image. The ultimate text region set contains regions detected in the first or second pass.

The stage of candidate region elicitation consists of the following operations:

- conversion of the image into gray scale,
- filtering with the median filter to remove some noise being usually the result of scanning or lossy compression applied to the image,
- increasing the font sharpness by applying sharpening filter with Laplace mask,
- binarization of the image using Otsu method,
- removal of long line segments,
- finding connected components,
- removal of the connected components which are of one pixel size (in this way we can eliminate the reminder of the noise),
- the division of connected components into sets of big and small components, where the threshold value is experimentally chosen,
- removal of big connected components,
- merging the connected components into candidate regions basing on their geometric neighborhood.

The removal of big connected components is motivated by the assumption that the size of text in the image is limited and big graphical elements probably are not text characters. The size threshold for separation of big and small components is determined taking the image resolution x_{res}, y_{res} and the maximal assumed text size s_t (assumed to be 70 pixels).

$$f_S = \max(0.3 * \min(x_{res}, y_{res}), s_t) \tag{1}$$

Application of connected components as elements of candidate regions fails if the text is intersected by elements of other geometrical shapes. In order to avoid such cases as much as possible, we applied the additional step consisting in removal of long line segments. The lines removal procedure must not destroy the overlapping text image, because it would decrease the text detection accuracy as well as the accuracy of the OCR procedure. In order to achieve it, the binary raster image is skeletonized and vectorized, i.e., its representation as the set of line segments is created. Using vector representation of the image, basic 2D shapes like triangles,

ellipses and rectangles are recognized. The shape recognition procedure applies the set of geometric rules specific to each shape. All line segments belonging to recognized shapes are marked as excluded and assigned to the set E. Additionally, line segments not assigned to recognized shapes but which length exceeds the threshold defined by equation 1 are also assigned to E. Next, width of the lines in the original image corresponding to segments in E is approximated. Let $\pi(e)$ denotes the set of pixels of the skeleton line approximated by the line segment $e \in E$. The line width is evaluated as the doubled average distance between the skeleton line pixels $\pi(e)$ corresponding to the approximating line segment e to the closest background pixel:

$$w_e = 2\frac{\sum_{p\in\pi(e)} \min_{b\in B} d(p,b)}{|\ \pi(e)\ |}, \tag{2}$$

where $d(p,b)$ denotes the Euclidean distance between pixels p and b and B is the set of background pixels. The idea of the line width approximation is shown in Fig. 1. Having line widths approximated, the lines in the set E are redrawn with the background color using their approximated widths. In this way, the long line segments disappear from the image. Unfortunately, also segments of text regions may be overwritten with the background color. To repair it, the foreground pixels located on the opposite sides of the line consisting of pixels $\pi(e)$ are connected by 1-pixel wide line segments drawn in the foreground color if their distance in less than $\sqrt{2}w_e$.

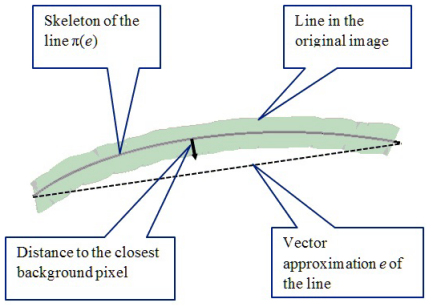

Fig. 1 Line width approximation by finding the closest background pixel

2.2 *Classification of Candidate Regions*

The stage of selection of candidate region consists of two steps:

- computing features for candidate regions,
- classification of candidate regions in order to eliminate the regions, for which the probability that they include the text is very small.

The regions elicited during the first stage of the whole process are separated only on the base of size and neighborhood of connected components in the binary image. In most cases, these regions in fact include the text fields but also very often the pseudo text regions which include other graphic elements of the image are created. That is why we need the selection process in which we can reject the region with small probability of text inclusion basing on another more complicated criteria. We applied typical approach using two-class pattern recognition, where the classes are: "the area includes text" and "the area does not include text". For candidate regions we determined the following features:

- foreground pixels coverage factor $f_{FB} = n_{FG}^{(a)}/(x_{res}^{(a)} * y_{res}^{(a)})$, where $n_{FG}^{(a)}$ is the number of foreground pixels in the region a and $x_{res}^{(a)}$, $y_{res}^{(a)}$ define the horizontal and vertical resolution of the region,
- the ratio of average line thickness to the height of the corresponding region (it allows for rejection of such regions in which this ratio is small what happens typically for graphical elements),
- the difference between the average background brightness and the average foreground pixels brightness (it allows for rejection of regions which are rather textures areas),
- density of connected components, i.e. the ratio of the connected components number to the area width / height aspect ratio

$$f_{CD} = \frac{n_C^{(a)}}{x_{res}^{(a)}/y_{res}^{(a)}} \tag{3}$$

  where $n_C^{(a)}$ is the number of connected components in the area a,
- the average width of consistency component calculated with respect to the height of the field (we assumed that consistency components correspond to the individual graphic characters, so their width should be less or equal to the height of the field),
- the average distance of foreground pixels to the regression line $y = ax + b$ (obtained by the Least Square Method) for the set of foreground pixels with respect to the hight h of the field (it allows for rejecting the regions containing the simple shapes like a line segments):

$$f_{AD} = \frac{\sum_{i=1}^{n} | y_i - ax_i - b |}{nh} \tag{4}$$

- maximal distance of foreground pixels to the regression line determined for the set foreground pixels, divided by the hight of the region,
- the average value from gradient map obtained as a sum of directional detectors of edges obtained by the method described in ([4]).

The classification rules for text regions or for the regions without the text are of the form of comparison the feature values with their thresholds. Next these predicates are used in a classifier based on simply decision tree. In such manner, a part of

candidate regions is rejected and the rest of them is subsequently processed using OCR tool.

2.3 Verification of Text Recognition with OCR Tool

The regions selected during the second stage of classification process now are analyzed by OCR tool. In our work, taking into account its accuracy proved by other researchers, the Tesseract OCR tool was chosen. Unfortunately, Teseract as well as many other robust OCR systems does not provide the evaluation of confidence of the recognition results. Because this knowledge is important in our system, then we try to estimate the confidence taking into account the contents of the recognized text string. One of the obvious approaches may use Levenshtein distance ([6]) of the recognized string to the closest sequence of words from the fixed dictionary. This approach hardly can be applied in the problem being considered here because it is almost impossible to build the dictionary of symbols appearing in a diagram. We however observed that in the case of the falsely selected regions, OCR applied to it produces the sequence of characters mainly containing symbols not being letters nor digits (*dirty symbols*). The set of dirty symbols consists of semi-graphic characters, punctuation marks " .,:;~! " and operators " #, -, +, /, *, &, ^" (if they appear side by side). If the sequence recognized by OCR contains plenty of such symbols (especially out of the set of typical punctuation marks like ".,:;~" then the result is not reliable, and most probably the region passed to OCR is not a true text region. As a measure of the likelihood for the recognized inscription, the following formula is adopted. The leading and trailing white characters: spaces and tabs, are not taken into account:

$$\eta = max(0, \frac{l - n_d - n_p - 0.2n_L}{l}) \tag{5}$$

where: l is the length of the recognized string, n_d is the number of dirty characters contained in the string not being typical punctuation marks nor operators, n_p is the number of punctuation marks and operators not being neighbors of other dirty characters and n_L is the sum of minimal Levenstein distances computed for subsequences of the recognized string consisting exclusively of letters or digits. By minimal Levenstein distance with respect to the dictionary, we mean the distance to the closest word in the dictionary. In order to calculate n_L the complete dictionary is necessary for language which is used in the inscriptions on the image. In some types of diagram (for example in UML diagrams) inscriptions containing the identifiers which are not the words from the vocabulary are very likely to appear. That is why this factor has small significance in the formula (5). The small value of likelihood coefficient denotes either that the quality of image is insufficient for correct recognition or the region transfered to the OCR module does not contain pure text. In both cases it seems to be a reasonable cause for rejecting the field as real text region.

Additionally for evaluation of text likelihood one can use the comparison between expected and real width of text field. We noticed that in the case of false classification of the field as text region, there are significant differences between the

real width of region under consideration and its expected width calculated as a sum of average widths of its component characters. These width are computed with respect to the high of the text field. By taking into account above mentioned relation, the recognition accuracy evaluation can be calculated as:

$$\lambda = \sqrt{2^{-|log_2(w/hw_{est})|}} \tag{6}$$

where w_{est} is the estimated width of text recognized by OCR module, obtained by summing average ratio coefficients (width/hight) for all characters appearing in recognized inscription. Finally the likelihood of recognition is determined as product $\eta\lambda$. During experiments we found that the region should be rejected if calculated likelihood is less than 0.3 .

3 Experimental Results

In tests we used seven typical class of diagrams: organization charts, UML sequence diagrams, pie charts, line charts, UML use case diagrams, flow charts, column charts. As a pattern the text field was chosen. We defined the text field as a geometrical closed region containing literary marks creating at least either one word of length not less than three or a number consisting of at least one digit. We did not expect recognition of isolated very short words of length less than three letters because they do not contain any useful information in document serching process and moreover it is very difficult to discriminate them from another small graphic elements of the image. As a measure as usual accuracy and recall were chosen. In the description of empirical studies the following notation was introduced:

- n_{tp} - the number of text fields, which were correctly found on the image,
- n_{tn} - the number of text fields, which were on the image but we did not find them correctly,
- n_{fp} - the number of subimage, which were recognized as a text fields but in fact they are not (we did not count the region including only one letter or pair of letters, beacuse in reality we deal with text data).

In consequence the total number of words on the images is equal to $n_{tp} + n_{tn}$ while the total number of regions indicated by our method is equal to $n_{tp} + n_{fp}$. The accuracy and the recall were determined as follows:

$$f_K = \frac{n_{tp}}{n_{tp} + n_{tn}}, \qquad f_P = \frac{n_{tp}}{n_{tp} + n_{fp}} \tag{7}$$

The result of empirical investigation concerning the accuracy and the recall of text extraction are presented in the table 1.

Let us notice that our investigation was focussed on accuracy of text region finding but not on the accuracy of OCR.

Table 1 Accuracy and recall of text extraction for different kind of diagrams

Category	n_{tp}	n_{tn}	n_{fp}	f_K	f_P
Organization charts	1595	175	77	0.9011	0.9539
UML sequence diagrams	291	37	36	0.8872	0.8899
Pie charts	143	14	3	0.9108	0.9795
Line charts	1177	185	67	0.8642	0.9461
UML use case diagrams	112	3	8	0.9739	0.9333
Flow charts	442	40	3	0.9170	0.9933
Column charts	703	72	26	0.9071	0.9643
Total	**4463**	**526**	**220**	**0.8946**	**0.9530**

The accuracy of our method is the greatest in the case of flow diagrams and block schemes and the least for schemes of UML sequences. The accuracy is decreasing in these case when:

- there exist parts of text which intersect with the graphic elements of the image,
- there exist small grapic elements of shapes similar to literary marks,
- there exist text elements written with the small font (the accuracy rapidly is decreasing when the font is of hight less than seven pixels).

We can also notice that the long inscriptions, which are the most important in text recognition on the image, were almost always correctly found. In the case of inscription written with the small letters the OCR module plays the crucial role as well in text region finding as in final text recognition. In this work external OCR module

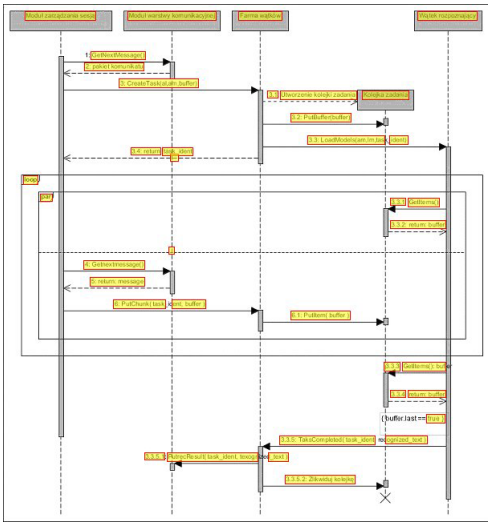

Fig. 2 The results of text extraction on the UML sequence diagram

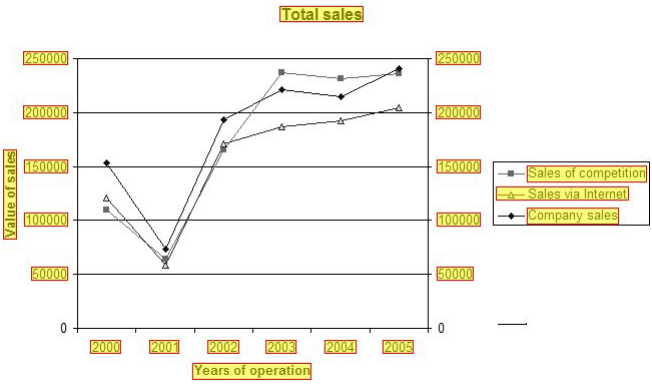

Fig. 3 The results of text extraction on the line chart

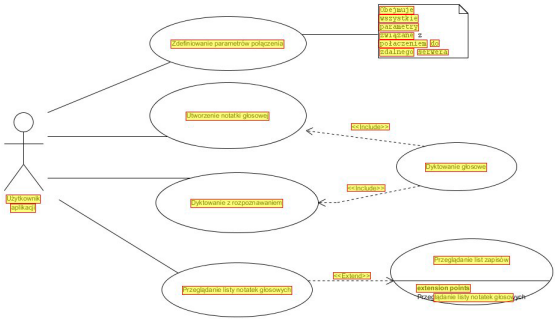

Fig. 4 The results of text extraction on the UML use case diagram

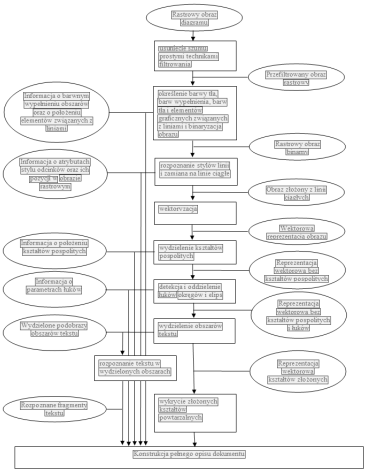

Fig. 5 The results of text extraction on the flow chart

was applied and its accuracy had the geatest influence for accuracy of whole method in the case mentioned above.

The results of text region extraction on the images of chosen investigated category were presented on the pictures 2 to 6. Additionally on the picture 7 the result of application of proposed method for drawings which is not a chart or a diagram is presented. The fields identified as the text regions are enclosed by rectangular boxes filled with gray shade.

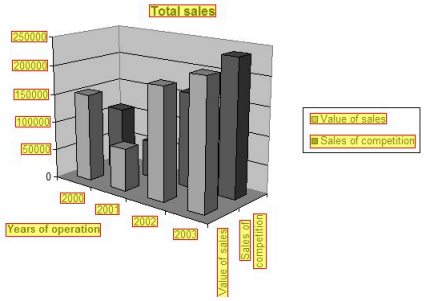

Fig. 6 The results of text extraction on the column chart

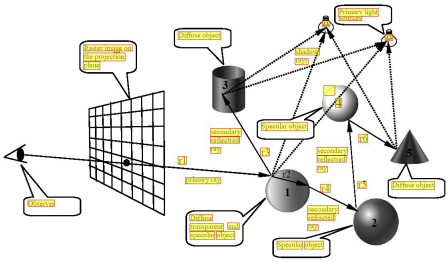

Fig. 7 The results of text extraction on the image which is not a diagram

4 Conclusions

In this paper we have focused our attention on the three-stage algorithm of text extraction from the images in case of text printed over textured background or the text is an element of the image consisting of many graphical elements not being text characters. In this work we consider the specific case of text extraction form specific type of images where text is imposed over graphical layer of vector images. Typical examples of category of images are: plots, UML diagrams, organization charts, flow charts, electric schematic diagrams etc. For such images we adopted several heuristic algorithms allowing finding the text region. We also applied some

kind of feed-back in this process using OCR tool taking into account its estimated accuracy.

Presented experimental results concerning proposed method imply the following conclusions:

- this algorithm seems to be an interesting alternative to another algorithms for text extraction,
- the accuracy and the recall of our method is aceptable for practical application,
- it can be effectively used also in the case when do not deal with diagrams,

although more practical investigations should be made. In particular we plan to apply our method to ICDAR 2003 benchmark base although our method is diagram-oriented. Moreover, it seems that applying own OCR procedure or procedure in which the support vectors for decision making are known could improve the quality of whole system because these factors play important role there.

Acknowledgements. This work has been in part supported by the Innovative Economy Programme project POIG.01.01.02-14-013/09.

References

1. Babu, G., Srimaiyee, P., Srikrishna, A.: Text extraction from hetrogenous images using mathematical morphology. Journal of Theoretical and Applied Information Technology, 39–47 (2010)
2. Epshtein, B., Ofek, E., Wexler, Y.: Detecting text in natural scenes with stroke width transform. In: Proc of IEEE Conference on Computer Vision and Pattern Recognition, CVPR, pp. 2963–2970 (2010)
3. Gllavata, J., Ewerth, R., Freisleben, B.: Proc. of 3rd International Symposium on Image and Signal Processing and Analysis, ISPA 2003, Rome, pp. 611–616 (2003)
4. Liu, X., Samarabandu, J.: Multiscale edge-based text extraction from complex images. In: Proc. of IEEE International Conference on Multimedia and Expo., pp. 1721–1724. IEEE Computer Society (2006)
5. Marti, U.V., Bunke, H.: Using a Statistical Language Model to Improve the Performance of an HMM-Based Cursive Handwritting Recognition System. Int. Journ. of Pattern Recognition and Artificial Intelligence 15, 65–90 (2001)
6. Paleo, B.: Levenshtein distance: Two Application in Data Base Record Linkage and Natural Language processing. LAP Lambert Academic Publishing (2010)
7. Wang, K., Kangasb, J.: Character Location in Scene Images from Digital Camera. Pattern Recognition 36, 2287–2299 (2003)
8. Wu, V., Manamatha, R., Riseman, E.: Finding text in images. In: Proc. of the Second ACM International Conference on Digital Libraries, pp. 3–12 (1997)