Detecting Arbitrarily Oriented Text Labels in Early Maps

Winfried Höhn

Department of Computer Science, University of Würzburg, Germany
winfried.hoehn@informatik.uni-wuerzburg.de

Abstract. In this work, we propose a novel method for robust, scale and rotation independent text/graphics separation for early maps. We apply a connected component analysis with density, minimum and maximum diameter as main features. In addition, we use a combined threshold region for the density and the ratio of maximum and minimum diameter, extended by an analysis of neighboring components to recognize text with large variations in style, size and orientations. Our method reaches an F1-score of 0.73 which is 0.19 higher than the 0.54 achieved by a state-of-the-art approach from the literature on the same test data set.

Keywords: multi-oriented text detection, early maps, graphical document analysis, connected component analysis.

1 Introduction

Early maps offer a unique insight into the past and are therefore of enormous scientific value. Researchers, library staff and other potential user groups need support from technology that goes beyond simple digitizing and makes finding relevant maps or particular points of interest on these maps easier. Support from technology can be, for example, computer-aided annotation or fully automated extraction of knowledge from maps.

Maps contain different layers with different types of information, for example, transportation, boundaries, hydrography and geographic names. For further automatic analysis, it is necessary to decompose a map into separate layers, which need to be processed with different information extraction methods. For textual information extraction (e.g. geographic names, place names), text recognition methods can be applied.

In this paper we address the problem of finding text in early maps with large variations in style, size and orientation. In contrast to text segmentation in modern maps, the text layer extraction in early maps is of a higher complexity through aging effects like, for instance, yellowed paper and bleached regions. In addition, the text does not follow a line structure, it may have arbitrary orientation or be curved. Moreover, early maps contain symbols for trees and place markers which fall in the size range of the text. Further in this work we refer to such symbols as non-textual symbols. Fig. 3(a) shows a typical fragment of an early map containing these elements.

This paper is organized as follows: in Section 2, we provide a brief overview of existing algorithms for text detection and text/graphics separation in different fields of application and their shortcomings. Further, in Section 3 we explain the proposed method in detail. Subsequently, we present the results and compare them to the current state-of-the-art in Section 4. Finally, in Section 5 we draw conclusions and outline future work.

2 Related Work

A wide range of text detection and text/graphics separation methods have been proposed for various types of images and applications. Most of them are based on texture features [4,10,12,16] or connected component features [8,9,11,15,17]. Texture based methods calculate features, for example local intensity changes, discrete cosine transform (DCT) and wavelet coefficients, for each point and different resolutions in an image. Special characteristics of text can be observed in these features, and therefore enable a separation of text. Methods based on connected components filter non-textual components according to feature-based heuristics and statistics over all connected components.

Methods for text detection in modern maps [3,5,6,14] use features like color to separate text from other elements, which is not applicable to early maps since they only use color in the background. If other elements of modern maps use the same color as text, they fall back to generic text/graphics separation methods.

Text/graphics separation has been often applied to line drawings or architectural floor plans [2,8,11,15]. These methods assume that the graphics are bigger than the text components and use a size threshold to separate them. This cannot be applied to early maps, since they contain non-textual symbols with a similar size as the text. As a result of the degradation over time, they also contain many broken fragments and drawing segments with widely varying sizes.

Similar to early maps, smaller text elements can be found in applications like, for example, natural scenes and photographs [9,17]. However, most of the text extraction methods usually applied for these types of data rely on a horizontal or nearly horizontal text orientation. The ICDAR text locating competition [13] with their well-known test set for this type of text location algorithms also contains almost entirely horizontal text labels.

Our proposed method is aimed to overcome the limitations of the methods described earlier in this section in the context of early maps.

3 Overview of Our Text Extraction Method

Most of the Latin fonts have several characteristics in common. To be easily readable they need to have a certain balance between ink and blank space on a page. This leads to similarities in the density of single or multiple touching characters for different fonts. The alignment of the text to the typographic lines implies that all characters usually fill the space between the baseline and the mean line, and do not exceed the descender and ascender line, see for example Fig. 2. Thus, all characters have a minimum height of the x-height and a maximum height of the distance between the ascender and descender line.

The feature selection for the approach presented in this work was motivated by these observations for Latin fonts. We use a connected component analysis combined with density, minimum diameter and maximum diameter as our main features for the text/graphics separation. We define density $\mathrm{dens}(c)$ for a connected component c as the ratio between the area of the convex hull and the number of pixels in c. If $\mathrm{dens}(c) < 1$ holds, we set it to 1. We combine the minimum diameter $\varnothing_{\min}(c)$ and maximum diameter $\varnothing_{\max}(c)$ to the diameter ratio $\mathrm{dr}(c) = \varnothing_{\max}(c)/\varnothing_{\min}(c)$. Density and diameter ratio are scale and rotation independent and therefore not sensitive to text direction or image resolution. Additionally we use the shortest distance between the pixels in two connected components, written as $\mathrm{dist}(c_1, c_2)$. Due to vastly varying sizes of text and non-textual symbols, we cannot specify separate thresholds for single features, as previously used in the text/graphics separation context, to distinguish between textual and non-textual symbols of similar size. Thus, we use a threshold region that takes into account both, density and diameter ratio, combined with an analysis of neighboring components to filter out non-textual components.

The following subsections explain our method step-by-step, with a set of all 8-connected components (CC_{all}) from the binarized image as our starting point.

3.1 Remove Dashed and Dotted Lines

For the x-height estimation we use information from neighboring text component candidates. Dotted and dashed lines would interfere with this step and the line thickness could be mistaken for the x-height, therefore we remove them in a preprocessing phase. This step is only in case that a map contains more line segments than text segments. As long as more line than text segments are removed, the method improves the conditions for the x-height estimation and thus is not very sensitive to the chosen parameters.

Dashes and dots are both convex and their minimum diameter should be equal to the line thickness. However, this is influenced by printing, aging, and digitizing. Therefore, a tolerance of 20% for the area difference of the segment and its convex hull as well as for the line thickness was chosen by visual inspection and confirmed in the test phase. All tested values in the range between 5% and 70% led in our experiments to exactly the same results for the estimated x-height.

To remove the dashed and dotted lines, the connected components in CC_{all} are filtered by their density. All components c with $\mathrm{dens}(c) < 1.2$ are added to the set of potential line components (PLC). With this filter we only get near-to-convex objects from which we create a set of similar sized and neighboring pairs. We consider pairs as similar sized, if $1/1.2 \leq \varnothing_{\min}(c_1)/\varnothing_{\min}(c_2) \leq 1.2$ holds, and as neighboring, if they satisfy $\mathrm{dist}(c_1, c_2) \leq 2 \cdot \max(\varnothing_{\max}(c_1), \varnothing_{\max}(c_2))$. If we identify three similar sized and neighboring PLCs, we finally label these components as dashed/dotted lines CC_{dash} and exclude them from the x-height estimation process, but not from the following steps. Thus, mislabeling of characters in this step does not mean that they cannot be detected as text subsequently.

3.2 Estimate the x-Height

We use the density and diameter ratio to find the text component candidates for the x-height estimation. For each connected component we determine if the point calculated from these two values lies within the thresholding area (see Fig. 1). To get robust values for these thresholds we analyzed 1214 fonts. We collected the fonts from "Google web fonts"[1] and "Fraktur mon Amour"[2], since many of our maps use Fraktur or Blackletter hand. This collection also contains artistic fonts, script fonts and fonts for different writing systems than Latin. We only considered fonts which satisfied the allowed difference for the number of connected components compared to the number of characters in the original strings and contained all characters appearing in our training. The training strings consisted of thousand randomly selected German place names and were rendered at different font sizes. The allowed difference was set to a maximum of 20%. For each component the size ratio and density were calculated. From these values we built a two-dimensional histogram. In this histogram we selected the area which is above 0.1% of the maximum histogram value and use it to discriminate between text and non-text components.

The threshold of 0.1% was selected for the following reasons. The thresholding area was constructed in the way that for each font at least 70% of the connected components in our training strings the diameter ratio and density is covered by the resulting thresholding area. This threshold was selected empirically by controlling the number of fonts covered to a reasonable extent. With the threshold of 0.1% of the maximum histogram value, we only found 10 fonts from the training set of 1214 fonts which had a lower coverage than 70% of connected components by the thresholding area.

After the thresholding area filter was applied to all components from CC_{all} without CC_{dash}, we have a set CC_{ta} of connected components which consist for the most part of single text characters or a few connected text characters.

We search for neighboring connected components within CC_{ta}. For this purpose, we calculate an interval from the smallest to the biggest x-height value for which the components at least fill the space between baseline and mean line and stay between the descender and ascender line, and also have a distance smaller than the x-height. The maximum values for the ascender ar_{max} and descender dr_{max} are given relative to the x-height to stay independent of the absolute size of the characters. For our experiments we set $ar_{\text{max}} = dr_{\text{max}} = 1.2$, which were the maximum values observed in our font training set. This results in a new set of neighboring connected components.

We build a histogram from the identified neighboring connected components for all their x-height intervals and determine the x-height with the maximum voting's. This is our estimated x-height h_{est}.

[1] `http://www.google.com/webfonts`

[2] `http://www.fraktur-mon-amour.de/de/`

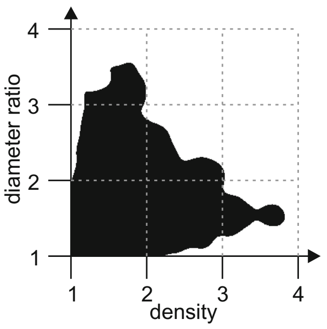

Fig. 1. Thresholding area

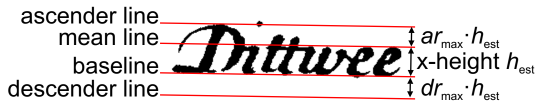

Fig. 2. Typographic lines

3.3 Filter Text Components

This step filters CC_{all} to obtain a set of components compatible with the calculated x-height and the maximum line height $lh_{\max} = (1 + ar_{\max} + dr_{\max}) \cdot h_{\text{est}}$. A component c is considered compatible if $\varnothing_{\min}(c) \leq lh_{\max}$ holds and one of the following conditions is satisfied: either $\varnothing_{\min}(c) \geq h_{\text{est}}$ or $h_{\text{est}} \leq \varnothing_{\max}(c) \leq lh_{\max} \cdot \alpha$ must hold. Italic and touching characters may have a maximum diameter that exceeds $lh_{\max}$, but still fit into the line. To compensate for this, we use the factor α which is set to $\sqrt{2}$ in our calculation, because the slant for italic fonts is usually lower than 45°. The result set of this step is referred to as FTC.

3.4 Cluster Text Components

In this step we group the connected components to text lines. For all ordered pairs of components from FTC we test if the distance between them is lower than $h_{\text{est}}/2$ and can form a line. Two or more components can form a line with an x-height of h_{est} if they are big enough to fill the space between the baseline and mean line and are small enough to fit between the descender and ascender line. We build initial clusters for all pairs of components, which are ordered tuples that hold these constraints.

From all tuples which have a component c at the first position we keep the tuples that contain the two neighbors of c with the lowest distance and their inverse tuples. We further refer to this set of neighboring text components as NTC. Nevertheless, a connected component can have more than two neighbors after this step.

We iteratively form lines from the components with the tuples from NTC as starting point. In each iteration, we extend the ends of the tuples from the previous iteration with their neighboring components. If they can still form a line according to the constraints described for a pair, they are added to the set of result tuples for the iteration step. This way, in each step we obtain lines which have one additional component. The results of all iteration steps are combined to a set of possible lines in the image. These lines can overlap or even be completely contained in another line. Since we use the component count of lines in our final character selection, we further combine lines which have an overlap of at least three components to new lines without checking our line constraints. This allows for the detection of slightly bent lines.

3.5 Final Component Selection

Finally we sort the detected lines by their component count. Starting from the longest line, all components of the selected line are added to the set of final character components, referred to as CC_{final}. All subsequent lines containing one of these components are removed. This step is repeated until there are no more lines left. All components in the resulting set CC_{final} are labeled as text components by our method.

4 Evaluation

Evaluation of text segmentation results can be done on different levels of abstraction, for instance, pixels, connected components, regions and recognized text. Each of these evaluation levels has its own drawbacks. Pixel-based evaluation cannot be done independently from the used binarization method. Furthermore, the classification of particular pixels as text or non-text is problematic for touching and overlapping textual and non-textual elements. OCR-based evaluation of the performance of a text segmentation method is strongly related to the performance of the OCR method, which is influenced among other factors by the language and used font. For an evaluation based on connected components or regions, a matching between the ground-truth and the automatically found text is challenging due to broken, merged and overlapping components.

A comprehensive study regarding the evaluation of text extraction on complex color images has been done in [7], where handling of merged or broken text components is addressed, but text touching non-textual elements is not considered. The authors introduce the notion of atoms as an identifiable part of the text. For instance, the symbol “i” consists of two identifiable parts, but several touching characters build one identifiable part and thus one atom. The evaluation is based on a matching of the extracted text with the atoms.

We use precision/recall as a measure for the recognition accuracy, which is usually taken as a quality measure in the literature. However, the description of the method for character classification is often omitted, see for instance [15] and may be unnecessary for data, where text does not touch the graphics and all characters are clearly separated and not broken. In early maps, we have to deal with broken, merged, touching and overlapping elements.

In order to make the evaluation objective and to ensure comparability of the method, we classified the elements as follows: all connected components, which contain also non-text parts or only consist of non-text parts, are annotated as non-text, even if the non-text part of the connected component is considerably smaller than the text part. The component does not have to be recognizable as

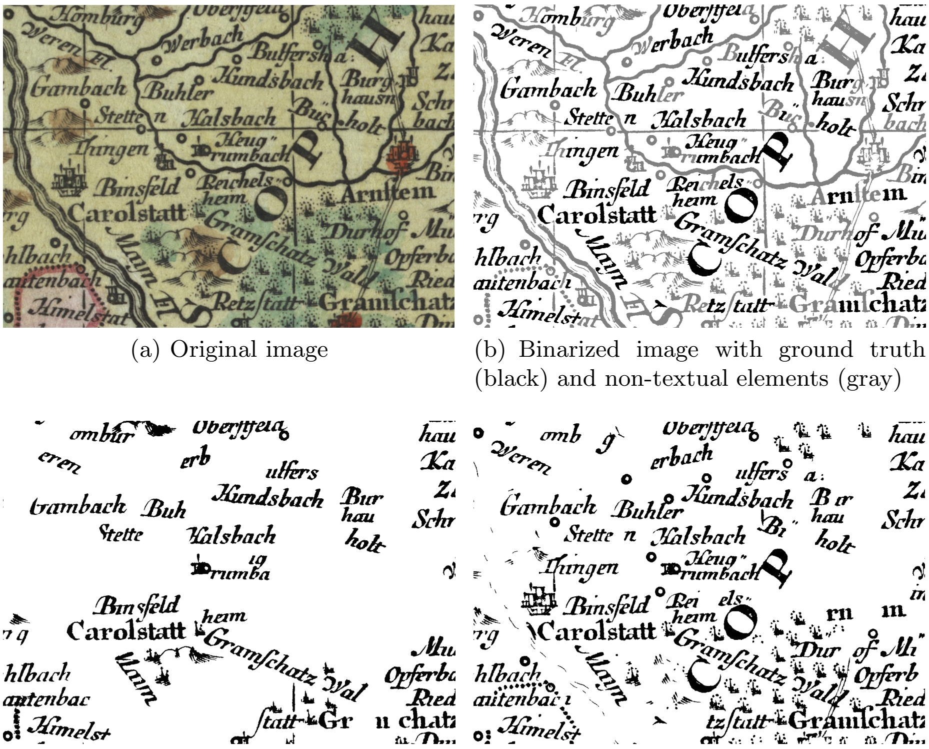
(a) Original image
(b) Binarized image with ground truth (black) and non-textual elements (gray)
(c) Result with proposed algorithm
(d) Result with Tombre's algorithm

Fig. 3. An extract of an early map, Circulus Franconicus from Matthias Seutter (1731)[4]

text itself, thus the point of an "i" or broken parts of characters are annotated as characters. In this way we omit ambiguous classification of touching and overlapping text and non-text described earlier in this section. In addition, we use a single binarized picture for the ground-truth annotation and for the tests. We compare the performance of our algorithm to a state-of-the-art algorithm introduced by Tombre et al. [15].

Fig. 3 illustrates the performance of the method proposed in this article compared to the method introduced by Tombre et al. [15] on an early map fragment. Fig. 3(a) depicts a raw unprocessed data fragment. Fig. 3(b) shows the ground truth annotation of the fragment. Fig. 3(c) presents the result produced by the

[4] Circulus Franconicus : in quo continentur Episcopat(us) Würtzburgens(is), Bambergensis, et Aichstadiensis. Status Equitum Teutonicorum, Ducat(us) Coburgensis, Marchionat(us) Culmbac. Baruth. Et Onoldinus, Principatus Schwarzenberg, Comitat(us) Henneberg, Wertheim, Holach, Reineck, Pappenheim, Erpach, Hanau, Castell, Baronatus Sensheim, Territor(ium) Noribergense /accurate delineatus per Matthaeum Seutter. - Augsburg, (1731) - coloured etching, 55,3 x 47,4 cm - scale approximately 1:450 000.

Table 1. Results

		Correct	Unexpected	Missing	Precision	Recall	F1
Initial	CC_{all}	6915	13692	0	0.34	1.00	0.50
Our method	CC_{ta}	5918	3362	997	0.64	0.86	0.73
	FTC	5846	3586	1069	0.62	0.85	0.72
	CC_{final}	4880	1541	2035	0.76	0.71	0.73
Tombre et al.		6460	10618	455	0.38	0.93	0.54

proposed method. In Fig. 3(d) the result of Tombre's method applied to the same data set is shown. As the reader may see, Fig. 3(d) contains considerably more non-textual components than Fig. 3(c), only large connected components like river systems were removed.

We summarize the results of the evaluation in Table 1. Our method outperforms the one proposed by Tombre et al. on the given test data set with an improvement of 0.19 in F1-score. Due to many small non-textual components and only few big connected components, Tombre's approach reaches results similar to the unprocessed connected components CC_{all} of the binarized image. Our preprocessing steps, CC_{ta} and FTC, which take place before comparing neighboring connected components, deliver higher F1 and precision scores than Tombre's method. However, in contrast to Tombre's approach, our method does not cover detection of isolated text symbols.

5 Conclusions and Future Work

A novel method for scale and rotation independent detection and extraction of text components from early maps has been presented in this article. The proposed method combines connected component analysis with consistency checks of neighboring connected components. The evaluation results show that the presented method improves the percentage of correctly detected text by 19%. However, the detection for isolated characters requires further elaboration. In addition, methods for recognition and separation of text touching graphics will allow to cover and evaluate the recognition more precisely. To achieve better results in these areas we plan to integrate key point matching as proposed in [1]. Different thresholding areas and their automatic, dynamic selection for different fonts can be applied to improve recognition of, for example, bold and regular fonts which differ in their blackness. Multiple thresholding areas would reflect these differences. Moreover, the detected text components can be used to identify common place name prefixes and suffixes (for example, "-hausen" and "-dorf" for Germany) and their spatial relations and match this with a geographic database to find the area depicted in a map.

References

1. Ahmed, S., Eichenberger-Liwicki, M., Dengel, A.: Extraction of text touching graphics using SURF. In: 10th IAPR International Workshop on Document Analysis Systems (DAS), pp. 349–353. IEEE (2012)

2. Ahmed, S., Weber, M., Eichenberger-Liwicki, M., Dengel, A.: Text/graphics segmentation in architectural floor plans. In: International Conference on Document Analysis and Recognition (2011)
3. Cao, R., Tan, C.-L.: Text/Graphics separation in maps. In: Blostein, D., Kwon, Y.-B. (eds.) GREC 2001. LNCS, vol. 2390, pp. 167–177. Springer, Heidelberg (2002)
4. Chen, X., Yuille, A.: Detecting and reading text in natural scenes. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2004)
5. Chiang, Y.Y., Knoblock, C.A.: An approach for recognizing text labels in raster maps. In: Proceedings of the 20th International Conference on Pattern Recognition, pp. 3199–3202 (2010)
6. Chiang, Y.Y., Knoblock, C.A.: Recognition of multi-oriented, multi-sized, and curved text. In: Proceedings of the Tenth International Conference on Document Analysis and Recognition (2011)
7. Clavelli, A., Karatzas, D., Lladós, J.: A framework for the assessment of text extraction algorithms on complex colour images. In: Proceedings of the 9th IAPR International Workshop on Document Analysis Systems, DAS 2010, pp. 19–26 (2010)
8. Fletcher, L., Kasturi, R.: A robust algorithm for text string separation from mixed text/graphics images. IEEE Transactions on Pattern Analysis and Machine Intelligence 10, 910–918 (1988)
9. Gatos, B., Pratikakis, I., Perantonis, S.J.: Text detection in indoor/outdoor scene images. In: First International Workshop on Camera-based Document Analysis and Recognition (2005)
10. Gllavata, J., Ewerth, R., Freisleben, B.: Text detection in images based on unsupervised classification of high-frequency wavelet coefficients. In: Proceedings of the 17th International Conference on Pattern Recognition, vol. 1, pp. 425–428. IEEE (2004)
11. Kasturi, R., Bow, S.T., Member, S., Member, S., El-Masri, W., Shah, J., Gattiker, J.R., Umesh, Mokate, B.: A system for interpretation of line drawings. IEEE Transactions on Pattern Analysis and Machine Intelligence 12, 978–992 (1990)
12. Kim, K., Jung, K., Kim, J.: Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(12), 1631–1639 (2003)
13. Lucas, S.M.: ICDAR 2005 text locating competition results. In: ICDAR, pp. 80–85 (2005)
14. Roy, P.P., Vazquez, E., Lladós, J., Baldrich, R., Pal, U.: A system to segment text and symbols from color maps. In: Liu, W., Lladós, J., Ogier, J.-M. (eds.) GREC 2007. LNCS, vol. 5046, pp. 245–256. Springer, Heidelberg (2008)
15. Tombre, K., Tabbone, S., Pélissier, L., Dosch, P.: Text/Graphics separation revisited. In: Lopresti, D.P., Hu, J., Kashi, R.S. (eds.) DAS 2002. LNCS, vol. 2423, pp. 200–211. Springer, Heidelberg (2002)
16. Wahl, F.M., Wong, K.Y., Casey, R.G.: Block segmentation and text extraction in mixed text/image documents. Computer Graphics and Image Processing 20(4), 375–390 (1982)
17. Yao, C., Bai, X., Liu, W., Ma, Y., Tu, Z.: Detecting texts of arbitrary orientations in natural images. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1083–1090 (2012)