Open access • Proceedings Article • DOI:10.1109/ICDAR.2001.953888

Line detection and segmentation in historical church registers — Source link

M. Feldbach, Klaus D. Tönnies

Institutions: Otto-von-Guericke University Magdeburg

Published on: 10 Sep 2001 - International Conference on Document Analysis and Recognition

Topics: Line (text file), Optical character recognition and Image segmentation

Line Detection and Segmentation in Historical Church Registers

Markus Feldbach and Klaus D. Tönnies
Computer Vision Group
Department of Simulation and Graphics
Otto-von-Guericke University of Magdeburg
P.O. Box 4120, D-39016 Magdeburg, Germany
{feldbach, klaus}@isg.cs.uni-magdeburg.de

Abstract

For being able to automatically acquire the information recorded in church registers and other historical scriptures, the writing on these documents has to be recognized. This paper describes algorithms for transforming the paper documents into a representation of text apt to be used as input for an automatic text recognizer. The automatic recognition of old handwritten scriptures is difficult for two main reasons. Lines of text in general are not straight and ascenders and descenders of adjacent lines interfere. The algorithms described in this paper provide ways to reconstruct the path of the lines of text using an approach of gradually constructing line segments until an unique line of text is formed. In addition, the single lines are segmented and an output in form of a raster image is provided. The method was applied to church registers. They were written between the 17th and 19th century. Line segmentation was found to be successful in 97% of all samples.

1. Introduction

Many historical documents existing in libraries and various archives may be exploited electronically. Automatic reading of the documents would provide historians or sociologists with efficient means for extracting information. Thus, we developed a new method for segmenting lines of text in church registers in order to provide necessary preprocessing tools for automatic reading of names and dates from these documents. The registers differ in several ways from contemporary documents. Text line segmentation needs to be adapted to their specifics.

Most line segmentation methods assume that the image of the text does not change much and that lines are well separated [6, 1]. Church registers, however, were written with lines of text being close to each other, with type, size and shape of the handwriting changing between different registers and with text of a given line possibly reaching into adjoining lines. Even methods that deal with line segmentation of data, where the writer does not know that his writings will be read automatically (so-called unconstrained data [9]), often require separation of the text with sufficient space between lines (e. g., [4], [8]) or straightness of the lines [7]. The method of [5], that allows every word to have its own baseline, assumes only a single line to be computed.

Our algorithm segments lines of text in cases where lines are close to each other, text from adjacent lines may touch each other and lines of text may not be straight (shown in Figure 1). The method uses a chaincode representation that is derived from the text in a preprocessing stage [2]. It proceeds by first finding baseline segments that are potential candidates for baselines. Baseline segments are joined together to form baselines. Information on the baseline location that is supplemented with information from the data is used to find the centre line. Ascender and descender lines were not segmented because lines of text were too close to each other with text crossing several lines. The algorithm was tested on texts from church registers of the county of Wegenstedt that span a range of 300 years.

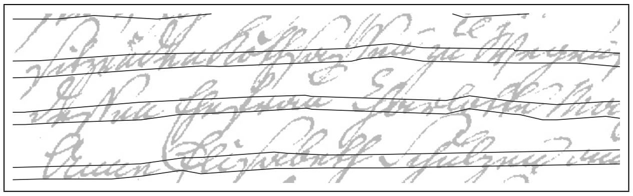


Figure 1. Part of an entry in a church register of the county of Wegenstedt in 1812, the reconstructed base and centre lines.

2. Preprocessing

Church registers were scanned on a flat bed scanner at a resolution of 300 dpi resulting in 24 bit RGB images. The registers were aligned such that the lines of text were approximately horizontal. Pen strokes are separated from the background using a threshold filter on the one color channel that exhibits the highest contrast after correcting for local brightness variations (see [2]).

The strokes are skeletonised using modules from the VECTORY software (Graphikon [3]) that creates a chaincode representation of the skeleton together with distance information on the width of each stroke. The representation consists of a set chaincode elements representing a part of a stroke between intersections. The ends of a chaincode element may be connected to zero, two or infrequently three chaincode elements. The set of connected chaincode elements forms a foreground object which is called a *continuum*. The text is represented by a set of continua.

3. Line Detection

Prior to assigning continua to lines of text, their defining thin "ledger" lines have to be found. Four different "ledger" lines, the ascender line, the descender line, the baseline and the centre line bound the characters in a line of text. The baseline is the most pronounced line in the church registers, and it will be our major input for segmenting lines of text. From the baseline, the location of the centre line can be deduced and corrected based on evidence from the pen strokes. The ascender line and the descender line are not well pronounced because ascenders and descenders too infrequently occur and their height has a large variance. We will not search for these lines but we will have to be aware of the problem it poses for later recognition of text. Characters from adjacent lines may reach into the current line so that continua reaching over several lines of text will have to be split up.

The baseline is found based on the local minima of all continua in y-direction. Even though lines of text curve, they are assumed to be locally straight. Local minima indicate, for the most part, points on the baseline and on the descender line with the majority stemming from baseline minima. Thus, the only line stretching over the whole width of the page and being made up of local minima from continua, that are close enough together and locally straight, should be the baseline. To a lesser extent, the same argument holds for finding the centre line based on the local maxima of the chaincode of the continua. Finding baseline and centre line is carried out in four steps: *(1)* Potential baseline segments (pBLSs) are found that are segments of straight lines through local minima of the chaincode. *(2)* Baseline segments (BLSs) are selected or constructed from the pBLSs. *(3)* Baselines are created by joining BLSs which represent the same baseline. *(4)* Centre lines are created based on the local maxima of the chaincode and on the assumption that they run approximately in parallel to adjacent baselines.

Parameters that were set for the processes described below assume a letter height (capital letters) of 6 mm and a letter width of approx. 2.5–3.5 mm.

3.1. Detection of Potential Baseline Segments (pBLSs)

The pBLSs are created from local minima of all continua on the page. Local minimum vertices $v^i_{\min}$ are marked. A pBLS consists of a direction α and an ordered list of these vertices with at least four elements. Adjacent vertices in this list must not be further apart than the double width of the letters (7 mm). None of the vertices may vary by more than 0.5 mm from the straight line connecting these vertices and defined by the direction α.

pBLSs are created independently for each $v^i_{\min}$ and for each direction at increments of 1° within $\pm 20^\circ$ (We found this range to be sufficient). It is attempted to add new vertices $v^j_{\min}$ that lie in direction α constrained by the abovementioned distance and deviation tolerances. The search for a pBLS terminates when no new vertices can be added.

After finding all possible pBLSs, an average line orientation is estimated from the histogram of directions of all pBLSs. Now, those pBLSs are excluded that deviate by more than $\pm 7^\circ$ from this main direction because our experiments indicated a variation of direction of BLSs by less than 12°.

3.2. Selecting Baseline Segments (BLSs)

The set of pBLSs contains many segments that are either not part of the baseline or that are contained in other pBLSs. The next step creates a subset of baseline segments (BLSs) from the set of pBLSs by removing segments if they are completely contained in another pBLS or if they are identical with another pBLS in an adjacent direction that has attributed a smaller maximum deviation from the average pBLS orientation.

From the remaining segments those are treated specially that represent conflicting baseline interpretations. This is the case if the vertex lists p_1 and p_2 of two crossing pBLSs exist with $p_1 = \left\{ v^a_{\min}, \ldots, v^i_{\min}, \ldots, v^j_{\min}, \ldots, v^m_{\min} \right\}$ and $p_2 = \left\{ v^b_{\min}, \ldots, v^i_{\min}, \ldots, v^j_{\min}, \ldots, v^n_{\min} \right\}$ (shown in Figure 2(a)).

Such line segments are separated into three subsets. The middle part consists of the identical set of vertices $v^i_{\min}, \ldots, v^j_{\min}$. One subset on the left side and one subset on the right side is chosen. The subset is chosen of

$v^a_{\min}, \ldots, v^i_{\min}$ or $v^b_{\min}, \ldots, v^i_{\min}$ and of $v^j_{\min}, \ldots, v^m_{\min}$ and $v^j_{\min}, \ldots, v^n_{\min}$ that contains the larger number of vertices.

In order to come to a decision between the subsets $v^a_{\min}, \ldots, v^i_{\min}$ and $v^b_{\min}, \ldots, v^i_{\min}$ respectively between the subsets $v^j_{\min}, \ldots, v^m_{\min}$ and $v^j_{\min}, \ldots, v^n_{\min}$, one is chosen that contains the larger number of vertices.

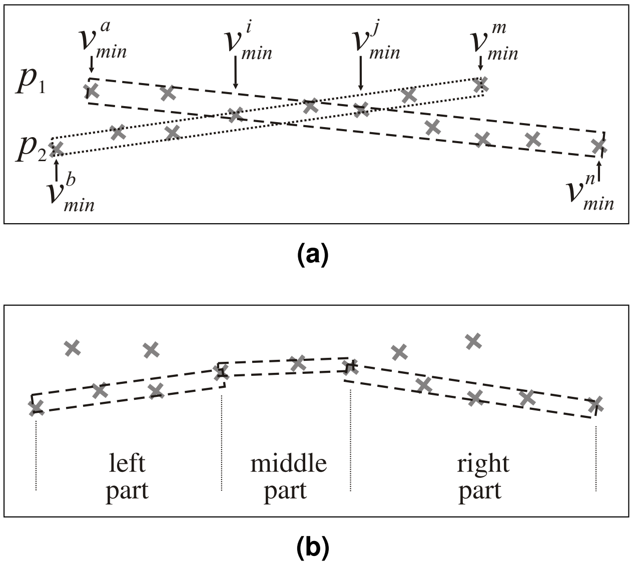


Figure 2. (a) Problem: crossing BLSs with vertex list p_1 and p_2. (b) Three resulting, new BLSs.

3.3. Creating Baselines

Elements of the set of BLSs are joined in order to form baselines. The process starts with the leftmost and uppermost BLS that is not part of a baseline and attempts to create a baseline by joining this segment with another BLS. The joint is possible if the next BLS is not further away with its leftmost vertex than 25 mm (approx. sevenfold character width) from the right most vertex of the current BLS and if the vertical distance difference is less than 3 mm (half the size of a capital letter). The process proceeds until no more BLS can be added. It is repeated for new baselines until no BLS exists that is not part of a baseline.

It may happen that false baselines are created from combining local minima, lying off the baseline, with artefacts from ascenders and descenders and even true local minima at the baseline (the latter because a tolerance of ±7° is still large). If two potential baselines intersect, that baseline will be deleted which contains the lower number of vertices.

Remaining wrong baselines are located between lines of text and they are generally shorter than true baselines. They may be detected because of these facts. First, the average vertical distance d_v between adjacent base lines is computed from all baselines that are longer than 80% of the average length of all baselines. The latter excludes wrong baselines from the computation. Then, baselines are discarded that are less than $0.6 \cdot d_v$ apart from adjacent baselines.

3.4. Computing the Centre Line

The centre line could be computed in a similar fashion from local maxima of the skeleton of the continua than the baseline, but the local maxima give a much weaker support for the line segments. Moreover, the path of the baseline is known. Given the assumption that the centre line is approximately parallel to the baseline and given the assumption that the distance between centre line and baseline is less than $0.6 \cdot d_v$ of the distance between two adjacent baselines, the centre line can be reconstructed.

Based on the position and direction of a BLS, a centre line segment (CLS) is searched to connect the maximum number of local maximum vertices. Such a vertex is counted if it does not deviate by more than 0.5 mm from the CLS. The horizontal distance constraint is not used because the existence of this CLS is known and only its position is searched.

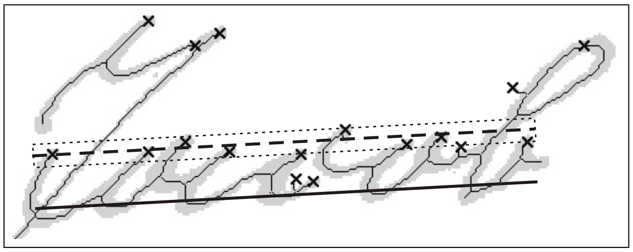


Figure 3. Chain-code of writing, BLS (solid line), CLS (dashed line) with tolerance space (dashed rectangle) for the maxima (crosses)

4. Line Segmentation

Before segmenting lines of text, baselines are shifted downward and centre lines are shifted upwards by the average stroke width. This is done in order to compensate the fact that the lines were computed based on the skeleton.

The knowledge about the path of the line and the position and size of the continua is used for carrying out the final segmentation. Each continuum has a fixed and known relation to any base and centre line. It may be situated either above or below this line or it intersects the line. This information is used to allocate the continua to lines of text.

4.1. Allocation of Continua to Lines

The *area of interest* of a line is the one part of the image which is important for its recognition. The upper bound of this area is defined by the baseline of the line above. The lower bound is defined by the centre line of the line below.

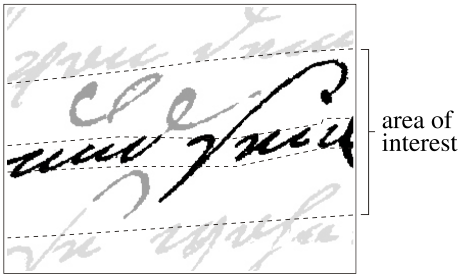


Figure 4. Allocation of continua.

For every line the continua are assigned to one of three classes (see Figure 4): The continuum is *definitely* (black), *possibly* (dark grey), or *not* (light grey) part of the line. Continua lying off the area of interest are not part of the line. Continua, which intersect the baseline or lay between base and centre line, are definitely part of the line. If a continuum is situated within the area of interest but above the centre line or underneath the baseline, there is no way for an absolute decision. Therefore, this continuum is labeled "possibly part of the line".

4.2. Splitting of Connected Lines

Interfering ascenders and descenders between the lines, being a significant feature of old script, cannot be allocated by the method described above. In such cases continua are extending over several lines. Different parts of such a continuum have to be allocated to different lines.

The continuum is divided into *subcontinua* by cutting the chaincode elements which connect the lines at their intersection with the baseline. The subcontinua are then also assigned to one of the three classes described above.

5. Results

The algorithm was applied to images from 61 paragraphs in 7 pages of the years 1649 (2 pages), 1727, 1741, 1812, 1817, and 1838. All 61 paragraphs together consist of 300 lines.

Currently, the parameters that are necessary for the processing, are determined manually. Some of them are dependent on the script size. For instance, the horizontal size influences the distances between the local extrema as well as the distances between the BLSs. The quality of the path of the line determines the range of angles in which the BLSs are searched. Furthermore, an uneven path of lines requires an increment of the vertical tolerances for the construction of BLSs and complete lines. In order to get an optimal result, parameters were to be adjusted to the kind of handwriting. However, the algorithm produces good results with constant values on different kinds of handwriting. With such a constant setting of parameters, 222 lines (90%) of 246 lines in 49 paragraphs containing six different handwritings were reconstructed correctly. In this test, records from the 18th century with another seventh handwriting were not used, because it was approx. 60–70% bigger than the other handwritings (example shown in Figure 5(c)).

If the parameters are adjusted according to the kind of handwriting, the quota of the correct recognition raises to 97%. We tested 300 lines containing seven handwritings. In 291 lines the baseline and the centre line were correctly reconstructed (e. g. Figure 5(a)). Five small errors occurred, where the textlines were found, but with the path differing slightly from the real one (see Figures 5(b) and 5(d)). An error was called serious if the difference between the reconstructed base and centre lines and the real lines was higher than the script size, if a textline was not found (see Figure 5(c)), or if an extra line was found at a wrong place. In this case a correct segmentation of lines was not possible. Serious errors happened four times. A 100% error-free reconstruction without preceding recognition of text itself will be impossible. This is the case because local straightness, horizontal continuity of lines and vertical continuity of distances between lines must allow for some variation. Cases can be constructed where this variation will lead to wrong selection of line segments (e. g. Figure 5(b)).

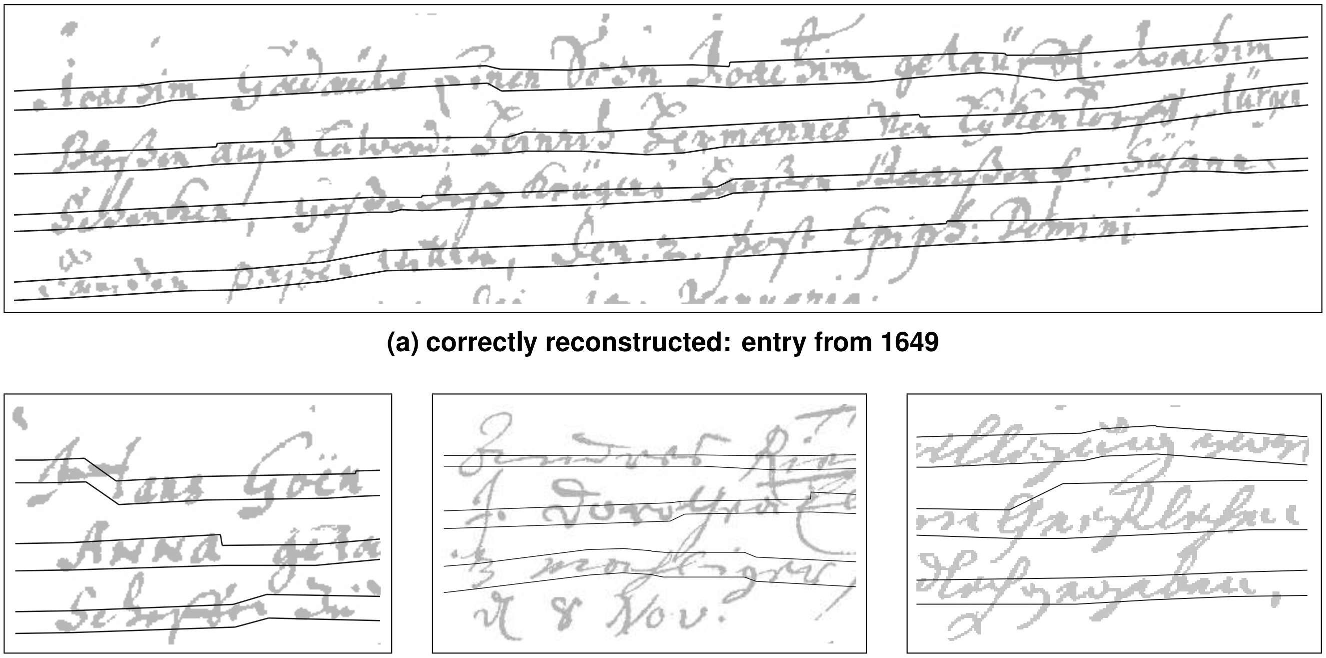
6. Conclusions

This paper provides an approach for segmentation of lines of text in old church registers. This segmentation is the required preprocessing for a later word recognition process.

After the computation of the skeleton, the path of base and centre line is reconstructed. These textlines have the best quality, and they are the most important for the recognition. Contrary to other handwritten documents, the textlines in this old script consist of several segments. These segments result from the layout of local extrema and are combined to complete textlines.

Connections between adjacent lines are another feature of old scripts. Therefore, in the segmentation step these connections are located and divided.

The results of this processing could be used by a word recognizer developed for this kind of script. In addition to



(a) correctly reconstructed: entry from 1649

(b) slightly differing baseline, cause: uncommonly wide 'H'

(c) lost line in an entry from 1727, cause: no centre line found

(d) problem in part of an entry from 1838: wrong centre line, cause: higher vertical variance of local maxima

Figure 5. Examples for correct and incorrect line reconstruction in entries of the county of Wegenstedt.

the objects of a line, such a recognizer could use the information about the allocation of continua to lines of text and the path of base and centre line.

In further research the determination of parameters will be automated. In order to reach this aim, geometrical features like script size will have to be computed.

References

[1] A. El-Yacoubi, M. Gilloux, R. Sabourin, and C. Y. Suen. An HMM-Based Approach for Off-Line Unconstrained Handwritten Word Modeling and Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8):752–760, Aug. 1999.

[2] M. Feldbach. Generierung einer semantischen Repräsentation aus Abbildungen handschriftlicher Kirchenbuchaufzeichnungen. Diploma thesis, Otto-von-Guericke University of Magdeburg, 2000.

[3] Graphikon Gesellschaft für Bildverarbeitung und Computergraphik mbH. Mandelstraße 16, D-10409 Berlin. URL: www.graphikon.com. Software: VECTORY, Version 4.0.

[4] G. Kim, V. Govindaraju, and S. N. Srihari. An Architecture for Handwritten Text Recognition Systems. *International Journal on Document Analysis and Recognition*, 2(1):37–44, Feb. 1999.

[5] S. Madhvanath and V. Govindaraju. Local Reference Lines for Handwritten Phrase Recognition. *Pattern Recognition*, 32:2021–2028, 1999.

[6] S. Madhvanath, E. Kleinberg, and V. Govindaraju. Holistic Verification of Handwritten Phrases. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(12):1344–1356, Dec. 1999.

[7] Y. Pu and Z. Shi. A Natural Learning Algorithm Based on Hough Transform for Text Lines Extraction in Handwritten Documents. In *Proceedings of the Sixth International Workshop on Frontiers of Handwriting Recognition (IWFHR VI),Taejon, Korea*, pages 637–646, 1998.

[8] M. Shridar and F. Kimura. Segmentation-Based Cursive Handwriting Recognition. In H. Bunke and P. S. P. Wang, editors, *Handbook of Character Recognition and Document Image Analysis*, pages 123–156. World Scientific, Feb. 1997.

[9] P. Steiner. Zwei ausgewählte Probleme zur Offline-Erkennung von Handschrift. Diploma thesis, University of Bern, Aug. 1995.