FI MU

Faculty of Informatics
Masaryk University

Off-line Recognition of Cursive Handwritten Czech Text

by

Pavel Smrž
Štěpán Hrbáček
Michal Martinásek

FI MU Report Series FIMU-RS-98-02

 February 1998

Off-line Recognition of Cursive Handwritten Czech Text

Pavel Smrž, Štěpán Hrbáček and Michal Martinásek

February 27, 1998

Abstract

In this paper a part of the system for recognising off-line cursive Czech text is presented. Recently, various systems for recognition of cursive English text has been developed, however, to our knowledge no method has been presented yet for Czech, a language rich in diacritic marks. This paper deals with preprocessing which is different for Czech and English handwritten texts. For finding the letter boundaries a method based on minimising a heuristic cost function has been used.

1 Introduction

Handwritten form of a language is used in notebooks, personal letters, on envelopes, cheques, etc. Taking into account the possible importance of these documents the benefits of automatic recognition of handwritten texts are obvious.

The problem of handwritten character recognition can be subdivided into two categories: off-line recognition [1, 2, 3] and on-line recognition [4, 5]. On-line recognition deals with real-time data processing and has the ability to integrate pen-movement and pressure information. Off-line recognition, however, is based on a static input of the data and relies only on pixel information for the recognition of each word [6].

Off-line cursive script recognition has progressed in the past thirty years from a novelty to a technology that can be implemented into commercial

(a) (b)

Figure 1: Histograms of handwriting: a) in English, b) in Czech

applications. Processing the characters with diacritic marks that are common in Czech, however, still represents a problem and has not been satisfactory solved yet. This paper deals with the preprocessing part of cursive script recognition in which specific features of a language rich in diacritic marks play the key role.

2 Finding Text Line Boundaries

Before starting let us define a useful term we will use in the following text: Smoothed pixel density histogram s is a histogram defined as follows:

$$s(i) = \sum_{j=-2..2} h\left(i+j\right), \qquad (1)$$

where $h(i)$ is a pixel density histogram at the point i.

Splitting the page to rows is the first step which needs specific handling in Czech. The typical profile of a horizontal smoothed pixel density histogram for an English text is shown in the Fig. 1a. In the case of a Czech text this typical form is disturbed due to the acute accents and inverted circumflexes that influence especially the characteristic form in the ascender part (see Fig. 1b). Algorithms looking for the boundaries of the text rows could not be therefore based on searching characteristic patterns in the horizontal histogram because the style of the upper part of the histogram differs in Czech texts not only according to the style of writing (the slant of acute accents and position of writing of the inverted circumflexes) but also to the contents of the text (the number of diacritic marks above the characters on the row).

The second problem we deal with in this part is the position of acute accents and inverted circumflexes above the text. Acute accents and inverted circumflexes may be too high above the text and a simple algorithm for finding the text rows could consider them an independent row. We have solved this problem in this way: We suppose that the rows have approximately the same height on the whole page. After an initial estimation of text lines the algorithm adjoins too low rows to the following row. Based on our experiments it is reasonable to assume that true rows are only those which are higher than a half of an average row.

We have tried to find the borders of a line as precisely as possible. The rows of a written text in Czech usually have not straight line boundaries. The boundaries are often overlapped due to acute accents and inverted circumflexes and due to characters descending beneath the lower boundary. For the first approximation of boundaries we should calculate the horizontal smoothed histogram and estimate the baselines for all rows. Then the contour following algorithm may be used for the parts of characters which overlap the low borderline in order to find precise row boundaries. Possible problems with some thin and slanted strokes can be avoided by using the 3x3 Gaussian mask for the scanned picture of handwriting before the algorithm is applied.

3 Splitting Rows To Words

The first phase of row processing is based on finding the reference lines. The reference lines of a text line are the four horizontal lines that mark the top of the ascenders, the top of the main bodies of letters, the baseline and the bottom of the descenders [7]. This phase should not be biased by Czech diacritic marks. The only exception may be the situation when the acute accents and inverted circumflexes are positioned too low above the text. In this case it may be a problem to find the top reference line.

Let us suppose that the slant of a writing on a particular row is known and we are trying to use it for splitting the row to words which is a logical continuation of the whole process. The problems which arise are similar to that with splitting a page to rows so that we can use an analogous way to solve them. First we calculate the vertical histogram at the same angle as that of the slant of written text and smooth it again. Then the spaces between the words may be estimated using the values of calculated histogram (text density). This approach works well if a good method for the estimation is employed. Corrections of erroneously split rows are possible after the words are recognised by comparing them with a dictionary.

Serious problems could arise by an improper positioning of acute accents and inverted circumflexes at the end of a word. If they are written too far behind the text the algorithm could separate them as an individual word. To avoid this situation a horizontal histogram for every short word is computed. The process reveals whether the word found in this way is located on the baseline of the text. A diacritic mark found in this way is joined to the precedent word.

4 Extracting Style Parameters of Czech Text

In the previous section we supposed that the slant of writing is known. The way to obtain its value along with other important characteristics of writing is described in following text.

Histograms are not sufficient for the phase of splitting the words. The inner structure of particular words is more complex than the structure of the whole page or a row. Therefore, it is necessary to calculate the following parameters that characterise the word to be split:

- dominant slant of writing
- thickness of the pen
- average width of characters
- average height of characters

It is obvious that the result of splitting the words largely depends on the accuracy of determination of particular parameters. To minimise the inaccuracy it is necessary to work not only with average values of these parameters but also with their deviations and to take them into consideration when the word is split.

The basic parameter used in handling with words is the slant of writing. All the methods mentioned in the following text are based on it. Let h be a histogram. Define $d(h)$ as a sum of differences of all pairs of adjacent values in histogram h. To determine the slant of writing the vertical histograms at angles of -20, -10, 0, 10, 20, 30 degrees are created and the value of

$d(h)$ is calculated for all of them. The angle with a minimum value is to be considered the dominant slant.

The vertical histogram for the dominant slant of writing can be used to find the thickness of the pen. As the average thickness of the pen the average of the set containing the smallest non-zero elements of the histogram in a given range is employed.

To determine the average width of characters we calculate the vertical histogram at the dominant slant angle. Then we search for places where the histogram value approximately equals to the determined width of the pen and the histogram values are rising on the right side. We calculate the average of the distances between these points as well as their deviations. The determination of the average width of characters already in this phase is rather difficult and the result is not fully reliable. Moreover, the characters like **n, u, m, w**, etc. can be misread as pairs or triples of letters. This problem can be eliminated in the phase of recognition and postprocessing only.

In the determination of the average height of characters we can considerably simplify the task by using the height of the highest character as the searched value.

Vertical histograms are influenced by the diacritic marks much more than the horizontal ones. The determination of the slant of writing is influenced by diacritic marks because the slant of the acute accents and the style of writing of inverted circumflexes considerably affect the histogram computed at different angles. The style of writing diacritic marks may differ from writing style of true letters and there is a risk that the slant of writing the marks could override the true slant of writing in case of short rows with a relative large number of letters with diacritic marks. The determination of the average width of characters could be also influenced by an improper position of acute accents. The influence of diacritic marks can be eliminated by a proper heuristics. Therefore, we use the values based on the histogram of the whole rows in which the deviation are eliminated by averaging.

5 Finding Letter Boundaries

The process of splitting words can begin after all style parameters are determined. The influence of diacritic marks is critical in this part of processing.

In this stage we work only with particular words. Therefore, the deviations in vertical histograms are not eliminated by averaging so that they can lead to misinterpretation of data. The slant of acute accents manifests itself by strong jumps in the histogram which prevent the algorithm from finding the position of letters. Simply, the algorithm designed for splitting letters in English texts does not work for Czech texts.

The solution of this problem can be temporal elimination of diacritic marks. It is performed by using the algorithm for finding single graphical objects in the upper part of the row. Then the original algorithm for splitting words into particular characters can be applied. However, problems may arise where diacritic marks are adjoined back to the letters because the acute accents and inverted circumflexes are not written at the constant place with respect to the position of the letter they belong to. Therefore, we decided not to attach the diacritics back to the text in this phase but to process it separately. A Czech text in which diacritic marks are absent is processed in the same way as an English text (including the recognition of particular letters). Diacritic marks are then adjoined to the final text. Re-attachement of diacritic marks based on a language model only to the letters where it is possible considerably improves usefulness of the algorithm.

The algorithm for splitting words is described in the following paragraphs [7]:

In the first phase the procedure is similar to the determination of the width of characters. Based on the prevalent slant of writing we choose the set of four angles in its neighbourhood and apply the algorithm for the determination of the width of characters to the histograms at corresponding slants with the following changes: When a proper place for division is found we add its position to the set M which was initialised to be empty. This process is repeated for all chosen angles. The set M can be, therefore, considered the set of candidates for division of a given word and our task is to use the most appropriate one. It is useful to order the set M after all candidates are added.

The first point in the ordered set M is chosen as the initial new point of division. Then we create the set N of points which are from the initial point distant of less than a half of the average width of character. This points we remove from the set M. In the next step we remove the point with the smallest value of the cost function (defined later) from the set N and insert it to the set Q of the final division points. This process is repeated until the set M is empty. Finally we unify the points in the set Q the inter-distance of which is less than a quarter of the width of characters. It is probable that such points correspond to the same boundary of a character.

The detailed algorithm can be described by the following steps:

1. Find the division point p
2. Let the threshold point t be at a distance of a half of the width of character from p.
3. Let the set N be the set of division points between p and t
4. If there are more division points with the same angle in the set N, delete all the points after the second one.
5. Chose the division point q with the smallest value of the cost function from the set N and insert it to the set Q
6. Unify the points in the distance less than one quarter of a character in the set Q

Finally we need to define the above mentioned cost function:

$$cost\,(\alpha, p) = w_1 \left(\frac{p - pp}{ew}\right)^2 - w_2 \left(\frac{p - pp}{ew}\right) + w_3\,(tc) + w_4\,(hc) \qquad (2)$$

The function is defined for the pair (α, p) where α is the angle of the division line and p is the position of the division point on the baseline. pp is the position of the previous division point on the baseline, ew is the average width of characters, tc is the number of pixels intersected by the division line normalised by the average width of the pen, and hc is the height of the highest point intersected by the division line normalised by the height of the row. Experimentally found constants w_1, w_2, w_3, w_4 control the correct division of letter pairs and triplets.

References

[1] C. Faure and E. Lecolinet. OCR: Handwriting. In R. A. Cole et al, editor, *Survey of the State of the Art in Human Language Technology*, pages 86–89. Center for Spoken Language Understanding, Oregon Graduate Institute, 1995. http://www.cse.ogi.edu/CSLU/HLTsurvey/ch2node6.html.

[2] P. Smrž. Handwritten characters recognition, 1995. (in Czech).

[3] A. W. Senior. Off-line handwriting recognition: A review and experiments. Technical Report CUED/F-INFENG/TR 105, Cambridge University Engineering Department, December 1992.

[4] C. Higgins and P. Bramall. An on-line cursive script recognition system. In M. L. Simner, C. G. Leedham, and A. J. W. M. Thomassen, editors, *Handwriting and Drawing Research — Basic and Applied Issues*, pages 285–298. IOS Press, 1996.

[5] R. K. Powalka, N. Sherkat, L. J. Evett, and R. J.Whitrow. Dynamic cursive script recognition: A hybrid approach. In *Advances in Handwriting and Drawing: A multidisciplinary approach*, 1994.

[6] S. Wesolkowski. Cursive script recognition: A survey. In M. L. Simner, C. G. Leedham, and A. J. W. M. Thomassen, editors, *Handwriting and Drawing Research — Basic and Applied Issues*, pages 267–284. IOS Press, 1996.

[7] B. A. Yanikoglu and P. A. Sandon. Off-line cursive handwriting recognition using style parameters. Technical Report PCS-TR93-192, Department of Mathematics and Computer Science, Dartmouth College, Hanover, NH, June 1993.

Publications in the FI MU Report Series are in general accessible
via WWW and anonymous FTP:

`http://www.fi.muni.cz/informatics/reports/`
`ftp ftp.fi.muni.cz (cd pub/reports)`

Copies may be also obtained by contacting:

Faculty of Informatics
Masaryk University
Botanická 68a
602 00 Brno
Czech Republic