Time-Efficient Stroke Extraction Method for Handwritten Signatures

BENCE KOVARI
Department of Automation and Applied Informatics
Budapest University of Technology and Economics
1111. Budapest, Goldman György tér 3.
HUNGARY
http://www.aut.bme.hu/signature

Abstract: - Off-line signature verification methods have to tackle with a number of high resolution images, therefore it is vital to use efficient image processing algorithms. This paper proposes a heuristic stroke extraction algorithm to create a compact representation of the signature whilst minimizing the loss of semantic information and the computational needs. Some issues of the algorithm are highlighted and – as far as possible – also eliminated. Finally an experimental validation of the algorithm is presented and evaluated. This algorithm could be used to refine existing signature verification methods, and also to create new ones.

Key-Words: - signature verification; off-line; stroke extraction;

1 Introduction

Signature recognition is probably the oldest biometrical identification method, with a high legal acceptance. From the technical perspective, it can be divided into two classes, the off-line (aka. static) and the on-line signature recognition problem. In the prior case the only input consist of scanned images of signatures, whereas in the latter case the whole process of the signing is recorded with a capturing device (digital tablet, camera, etc.). Examining the literature of the past decades it can be shown that on-line signature analyzers outperform off-line analyzers by a whole magnitude. This can be explained by the fact that they can take advantage of the dynamic features like acceleration, velocity and the difference between up and down strokes. However in the most common real-world scenarios, this information is not available, because it requires the observation and recording off the signing process. This is the main reason, why static signature analysis is still in focus of many researchers. Off-line methods do not require special acquisition hardware, just a pen and a paper, they are therefore less invasive and more user friendly. In the past decade a bunch of solutions has been introduced, to overcome the limitations of off-line signature verification and to compensate for the loss of accuracy.

2 Related work

One of those limitations is the absence of temporal information, which can be used to give an almost unambiguous matching between selected features of both signatures. This allows on-line methods to concentrate on the comparison of the given features [1][2]. To give off-line signature verifiers the same opportunities, the whole process of signing should be reconstructed, which can only be based on stroke extraction. Several stroke extraction methods have been introduced in the past. Some robust stroke extraction solutions have been developed for the purpose of recognizing handwritten text [3][4] and there seems to be an extensively wide study of extracting strokes from Chinese characters [5][6][7][8]. However, these methods tend to work at a high level of abstraction (they focus on recognizing letters and words) and are thereby not suitable for detecting the fine features used in signature verification.

An other class of methods is based on simple line tracing. Either because the resolution of the signature is already low [9], or because (as in the most of the cases) they apply some line thinning algorithms [10][11][12]. In both cases the loss of semantically important information (Based on the list of 21 discriminating elements of handwriting used by forensic document examiners [13]) is high. Although these methods (as one of our previous works [14]) can deliver comparable results to other solutions [15] they are hard to improve above a given level.

Jose L. Camino et al. [9] guess the pen movements during the signing by starting at the left and bottom most line-end and then following it in the original image. There are also other approaches trying to reconstruct the signing process. In [16] stroke, and sub-stroke properties are extracted and used as a basis for the comparison. A three-stage stroke extraction method, involving an interesting stroke following method has been proposed in [17]. However, this only targeted characters and graphemes. Based on own experience, these latter approaches seem to be the most promising, because their results can be explained (and therefore improved).

3 Proposed method

In the following section a robust algorithm is introduced with the purpose to identify the way how the signer wrote his signature. The main goal was to create an algorithm that performs well on noisy, unprocessed images (An average image has a resolution of 800*400 pixels, with a color depth of 32bits).

In general, this method traces a signature using the image of it, extracts control points from it, determines their order, and finally assigns them to strokes. This gives a graph representation of the signature, which can be used for further processing.

3.1 Basic method

Inputs were raw, scanned images on which no noise filtering or morphological operators (for the thinning process) were used.

First, some new definitions must be introduced.

3.1.1 Definitions

Signature point: a pixel in the image, which belongs to the signature.

Stroke point: strokes are in fact polylines represented by the endpoints of their segments. These endpoints are called here stroke points

p: pen width (in pixels)
r: the radius of the scanning circle. Usually $r = p * 2$
d: constant value, usually one unit smaller than r

Weighted middle point: given a series of points $P_1, P_2 ... P_n$, with corresponding intensity values of $i_1, i_2 ... i_n$, the weighted middle point (P_m) is the first point in the series where

$$\sum_{k=1}^{m} i_k > \frac{1}{2} \sum_{k=1}^{n} i_k$$

Free point: any pixel which has a minimum distance of *d* from any previously detected stroke point.

Connected points are pixels which can be connected with a straight path consisting only of signature points

3.1.2 Algorithm 1

Step 1.

Going from the bottom to the top and from the left to the right, locate the first free signature point (*P*). This point is going to be the starting point of our next stroke. $S := P$

If there is no free point, then exit.

Step 2.

Examine all points of a circle with a center *S* and radius *r*. More than 3 connected signature points on the circle define arcs. Select the weighted middle points ($A_1, A_2 ... A_n$) of these arks as possible stroke points.

Step 3.

Deselect all A_i where A_i is not free, or not connected with *S*

Step 4.

If count(selected A) is

- 0: Finish the current stroke and go to *Step 1*
- 1: Take the ark point as the next stroke point ($S := A$) and go to *Step 2*
- >1: finish the stroke and begin two (or more) new strokes with starting point S, and take $A_1, A_2 ...$ as the second point. Now, follow *Step 2* for each of them.

A sample run of the algorithm is illustrated in Fig 1.

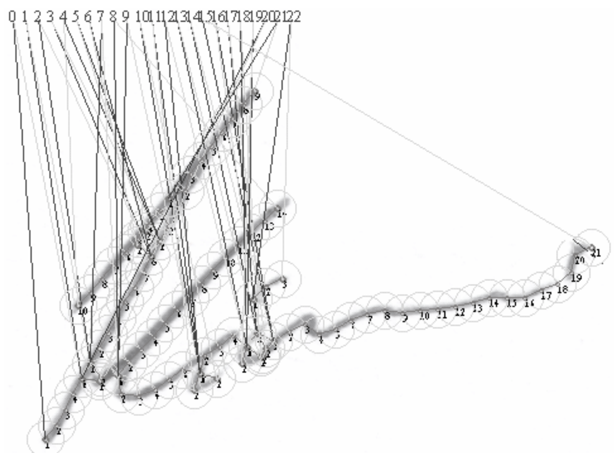


Fig. 1: a sample run of Algorithm 1. Strokes are numbered and marked by the dark lines. Scanning circles are illustrated in light gray color.

3.2 Improved method

It can bee seen in Fig. 1 that Algorithm 1 successfully processed the signature. However, the first tests also revealed some problems, which have to be addressed.

3.2.1 Connection of strokes

When reaching junction points (*Step 4*) several new strokes are defined. This is a necessary side effect of the algorithm, which can be eliminated in a post processing step.

A junction point is the meeting point of at least 3 but possibly more strokes. Stroke pairs can be chosen and the strokes in a pair can be linked to form a new single stroke and thereby reduce the number of detected strokes in the signature. All possible combinations are examined and a goodness value is assigned to each of them to make the best choice about which strokes should be connected.

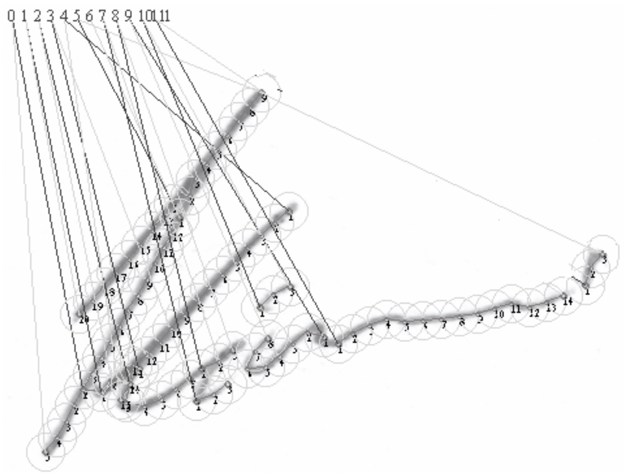


Fig. 2: the number of strokes could be reduced from 22 to 11.

3.2.2 **Refinement of signature point**

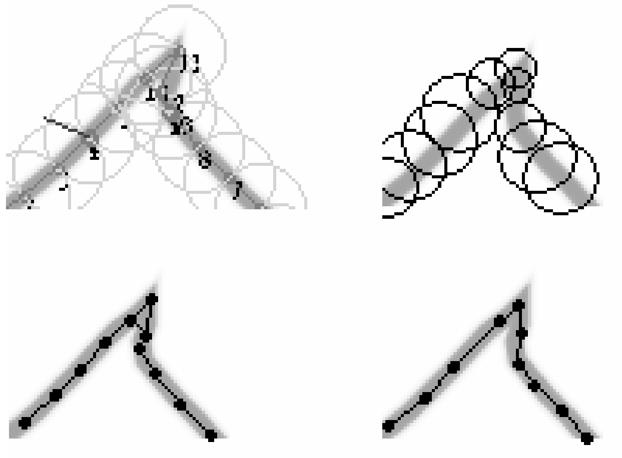
In real world scenarios a significant part of the pixels in an image can not be unambiguously defined as “signature” point or “paper” point. A third class of points “undefined” is introduced to overcome this problem. The distinction is made by examining the intensity values (especially the blue component). This also implies some modifications in our previous definitions:

Connected points are pixels which can be connected with a straight path consisting only of “signature” and “undefined” points where the number of connected “undefined” points can not exceed 2, and the total number of “undefined” points must be less than the total number of “signature” points

In *Step 2* an arc is also allowed to contain “undefined” points as long as their total number does not exceed the total number of signature points.

3.2.3 **Adaptive circle radius**

Although the original algorithm performs well when following straight lines, curves and even when detecting junctions but the high value the scanning radius (r) makes for it impossible to correctly detect sharp curves in the original strokes. This can result in the detection of phantom strokes (see Fig 3a).



a) b)

Fig. 3: stroke detection with constant (a) and with adaptive (b) scanning radius.

To tackle this problem *Step 2* is modified to allow the changing of *r*. Whenever the angle between two stroke segments falls below α or the number of points in an ark goes significantly beyond the pen width (*p*) a new scanning circle is tested with a reduced radius (*r*). The radius is decreased more times if necessary but of course it can not get under *p/2*. In the next iteration *Step 2* will start with the reduced radius, but contrary to the previous case it will try to increase the radius if it can, whilst maintaining a maximum value of 2*p*,

A sample run of the improved algorithm is demonstrated in Fig 4. The algorithm still has some minor flaws, but we have shown a way to extract stroke point from noisy signatures. The order of the points should be handled with greater care, but this tends to be an easy task based on [18] and [19].

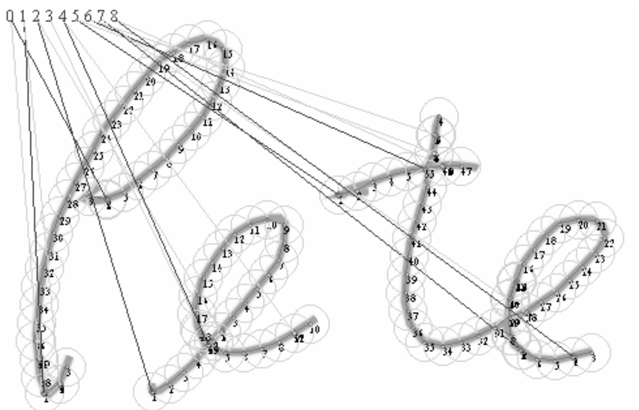


Fig. 4: Improved stroke extraction on a reconstructed signature

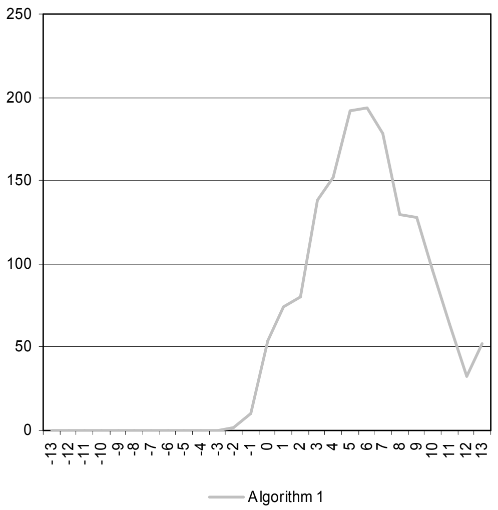


Fig 5: The distribution of the deviance in the number of strokes of our stroke extraction algorithm compared to the original strokes. (Y-axis: number of signatures, X-axis: deviance)

4 Experimental results

Because of the use of several heuristic methods in the algorithm, a continuous monitoring of the accuracy was essential. Although we were able to validate single cases with human interaction on our database [20], a statistical validation is hard to obtain, because of the missing on-line information. To compensate for this, the database of the Signature Verification Competition 2004 [21] was used. This is an on-line signature database therefore it already contains the original stroke information, but no images are provided. The stroke information was used to synthesize signatures similar to the original ones. Stroke points were connected with straight lines, fading out on the line borders. Bicubic interpolation and anti-aliasing were used to make the final image smoother. An example of reproduced signature can bee seen on Fig. 4. Although these signatures are still far from good forgeries, they are adequate for testing our stroke extraction algorithm. 1600 signatures from 40 signers (20 originals and 20 forgeries from each) ensure a sample large enough for our purposes.

Without going deeper into the semantics, a rough comparison can be given by comparing the numbers of detected strokes in a signature. Fig. 5 shows the number of signatures with given deviations in stroke counts. It can bee seen, that our algorithm almost always overestimates the count of strokes, which can be tracked back to the fact, that nearby strokes have not always a common point (as assumed in 3.2.1). Therefore the connection of nearby strokes is not always possible (Fig 6).

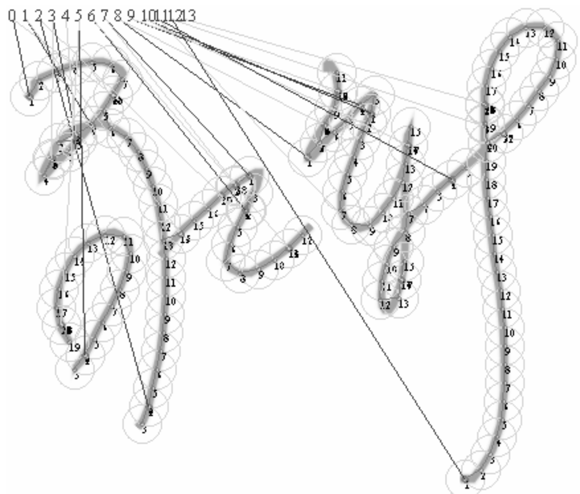


Fig. 6: 13 strokes are detected instead of the original 4.

However, these results are still promising, because these "extra" strokes are detected in the same way in all signatures. Fig. 7 shows the average deviation among the detected stroke counts of the same signature. The average value of 1.9 is relatively low compared to the average stroke count of 13.6. This indicates that the output is (mainly) consistent and could thereby be used in comparison methods.

The average processing time for a signature (on Pentium 4, 3GHz computer) was 0.921sec.

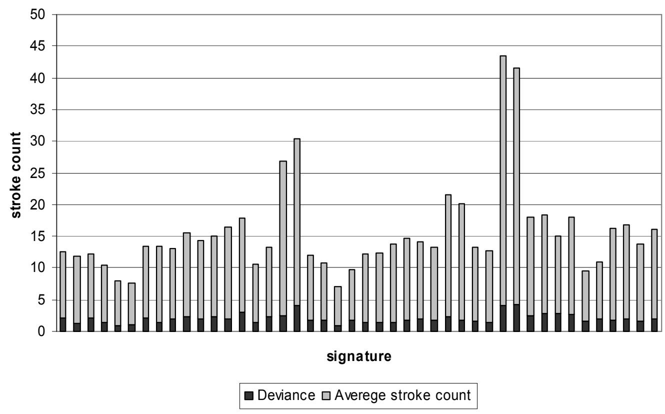


Fig. 7: Deviances in the detected stroke counts within signatures

5 Conclusion

A method has been proposed for detecting and efficiently representing strokes in scanned images of handwritten signatures. Several related problems were introduced and solved and it has been demonstrated that applying these, a higher accuracy can be reached. This makes the algorithm a suitable base for further use in a signature verification process, which is subject to our ongoing researches and will be targeted in our future works.

References:

[1] S. Hangai, S. Yamanaka, T. Hamamoto, "Writer Verification using Altitude and Direction of Pen Movement," IEEE, *Proceedings of the International Conference on Pattern Recognition* (ICPR'00)

[2] M. E. Munich, P. Perona, "Visual Identification by Signature Tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25*, no. 2, February 2003

[3] B. Talko, "Extracting stroke information off-line for cursive handwriting recognition", Honours Report, University of Melbourne, 1995

[4] A. Pérez-Hernández, A. Sánchez and J.F. Vélez, "Simplified Stroke-based Approach for Off-line Signature Recognition", *Second COST 275 Workshop*, 2004

[5] Y. M. Su, J. F. Wang, "A learning process to the identification of feature points on Chinese characters*," IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans, Vol. 33*. No. 3, May 2003

[6] R. Cao, C. L. Tan, "A Model of Stroke Extraction from Chinese Character Images," *15th International Conference on Pattern Recognition* (ICPR'00) - Volume 4 p. 4368

[7] I.-J. Kim, C.-L. Liu1, J.-H. Kim, "Stroke-guided Pixel Matching for Handwritten Chinese Character Recognition", *Proceedings of the Fifth International Conference on Document Analysis and Recognition*, 1999

[8] Ke Liu, Yea S. Huang, and Ching Y. Suen, "Identification of Fork Points on the Skeletons of Handwritten Chinese Characters," *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 21*, No. 10, October 1999

[9] J. L. Camino, C. M. Travieso, C. R. Morales, M. A. Ferrer: Singature Classification by Hidden Markov Model. *IEEE 33rd Annual 1999 International Carnahan Conference on Security Technology* 481-484

[10] Yu Qiao, M. Nishiara, M. Yasuhara, "A Framework Toward Restoration of Writing Order from Single-Stroked Handwriting Image," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 28, No. 11, November 2006

[11] K. Ueda, "Investigation of Off-Line Japanese Signature Verification Using a Pattern Matching," *Proceedings of the Seventh International Conference on Document Analysis and Recognition* (ICDAR'03)

[12] J. Mahmud, C. Mofizur Rahman, "On the Power of Feature Analyzer for Signature Verification," *Digital Image Computing: Technqiues and Applications, Proceedings*, 2005. DICTA ' 05.

[13] R. A. Huber and A. M. Headrick. *Handwriting Identification: Facts and Fundamentals*. CRC Press, LCC, 1999

[14] B. Kővári, Zs. Kertész, A. Major, 2007, Off-Line Signature Verification Based on Feature Matching *INES, 11th International Conference on Intelligent Engineering Systems*, 2007

[15] B. Kővári, "The Development of Off-Line Signature Verification Methods, Comparative Study," *Proceedings of microCAD, International Scientific Conference*, 2007

[16] J. K. Guo, D. Doermann, A. Rosenfeld, "Off-Line Skilled Forgery Detection Using Stroke and Sub-stroke Properties," *Proceedings of the International Conference on Pattern Recognition* (ICPR'00)

[17] V. Pervouchine, G. Leedham, and K. Melikhov, "Three-stage Handwriting Stroke Extraction Method with Hidden Loop Recovery," *Proceedings of the 2005 Eight International Conference on Document Analysis and Recognition* (ICDAR'05)

[18] S. Lee and Jack C. Pan (Presentted by Michael Janssen), 2005, Offline tracing and representation of signatures JC PAN *IEEE transactions on systems, man, and cybernetics* 22:44, pp. 755-771, 2005

[19] K. K. Lau, P. C. Yuen, Y. Y. Tang, "Stroke Extraction and Stroke Sequence Estimation On Signatures," Pattern Recognition, 2002. *Proceedings. 16th International Conference* on pp. 119-122

[20] Off-line signature database, Budapest University of Technology, IT2, [Online]. Available: http://www.aut.bme.hu/signature

[21] First International Signature Verification Competition, 2004 [Online]. Available: http://www.cs.ust.hk/svc2004/