International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:5, 2007

Stroke Extraction and Approximation with Interpolating Lagrange Curves

Bence Kővári, and ZSolt Kertész

***Abstract*—**This paper proposes a stroke extraction method for use in off-line signature verification. After giving a brief overview of the current ongoing researches an algorithm is introduced for detecting and following strokes in static images of signatures. Problems like the handling of junctions and variations in line width and line intensity are discussed in detail. Results are validated by both using an existing on-line signature database and by employing image registration methods.

***Keywords*—**Stroke extraction, spline fitting, off-line signature verification, image registration.

I. Introduction

SIGNATURE recognition is probably the oldest biometrical identification method, with a high legal acceptance. Even if handwritten signature verification has been extensively studied in the past decades, and even with the best methodologies functioning at high accuracy rates, there are a lot of open questions. The most accurate systems almost always take advantage of dynamic features like acceleration, velocity and the difference between up and down strokes. This class of solutions is called on-line signature verification. 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.

II. 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 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 in. Based on own experience, these latter approaches seem to be the most promising, because their results can be explained (and therefore improved).

After reconstructing the strokes, the order of strokes has still to be determined. This problem can be targeted for example with the universal writing model [18].

III. Description of the 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; this is why the term robust is used here. In general, this method traces a

Manuscript received Juny 15, 2007. This project was supported in part by the Innovation and Knowledge Centre of Information Technology BME(IT^2), the National Office for Research and Technology (NKTH) and the Agency for Research Fund Management and Research Exploitation (KPI).

Bence Kővári is with the Budapest University of Technology and Economics, Department for Automation and Applied Informatics, Budapest, Hungary (phone: +36-30-4465647; fax: +36-1-4632871; e-mail: beny@ aut.bme.hu).

Zsolt Kertész is with the Budapest University of Technology and Economics, Department of Control Engineering and Information Technology, Budapest, Hungary (e-mail: kertesz@ iit.bme.hu).

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 spline fitting.

This method is a topological feature extraction method. A topological method was introduced in [19], where a general human-like signature tracing method is described in-depth, using a thinned signature and heuristic rules for the purpose, and defining several solutions for removing noise caused by the thinning process. In [20] a signature thinned to one pixel width is the input for the stroke extraction and then several cost functions are defined for determining the overall stroke sequence.

The main goal was to improve the robustness of these algorithms, thus the inputs were raw, scanned images on which no noise filtering or morphological operators (for the thinning process) were used. (Currently morphological operators are only used for obtaining the starting points of the signature components, but this does not affect the original image.)

The algorithm is based on the use of simple virtual bows or with other word, a compass. Beginning with a start point the pin of the bows is stuck in it and a circle is drawn. Where this circle sections the line of the signature, it gives an arc. The middle point of this arc is selected as a possible following point, and if it meets the necessary conditions, it is taken as the new middle point. Iteratively repeating this step the whole signature can be traversed, but there are several difficulties to face.

First of all the radius the bows uses has to be determined. For this a circle is drawn with a constant radius. If an adequately large arc is obtained, it is stored. We start the circle with the first white point found in order to avoid the loss of an arc, because if we would start in the middle of the signature, we could half an arc that is just big enough and we would throw away its two half. After the first section is obtained, the distance of the two edge points of the arc is calculated, and heuristically 1.5-3 times of its size is used as a radius. Too large values produce too rough representation and information is lost, too small values are simply not big enough to make a section. To decrease the possibility of a wrongly chosen radius size, it is further normalized in the next few steps.

Sometimes it is not an obvious task to differentiate between the points of the signature and the noise. It is assumed that only blue ink is used during the signing. With this information the blue domination can be determined, calculated as the difference of the blue colour component and the average of the other two (red and green) colour components. Splitting this parameter range in three parts three classes of signature points can be defined: paper, ink and undefined. In the paper and ink classes the unambiguous points are categorized with a heuristic threshold, the rest is put in the undefined class.

Convexity of the points was first declared as: two points are convexly connected, if the straight path between them contains points only over a given threshold. Later this did not qualify because of the noisy input, so some undefined and even some paper point had to be accepted.

To further improve this method, “level difference” is calculated between the points: the size of connected points from the same class on the path is calculated, and where at least two continuous points of the same kind are found, the average intensity of the two points is calculated. This way a quantified path is obtained, and the difference of the highest and lowest level is calculated. This difference is a necessary measure when too close points must be separated, because going off the line and coming back again can be detected this way.

Another way of path improving comes useful at junction points. If one of the possible following points can be reached from another one on a better path (the maximum and total size of the undefined and paper points is used for this parameter) than from the junction point, then the connection of it to the junction point is replaced with a connection to the other point.

Loops also have to be detected and handled with care. A loop is detected if looking ahead from the actual point for a short distance a previously visited point can be seen and convexly connected to the current point. During this search the points are prioritized in the end, junction, common point return order (the first one found is returned).

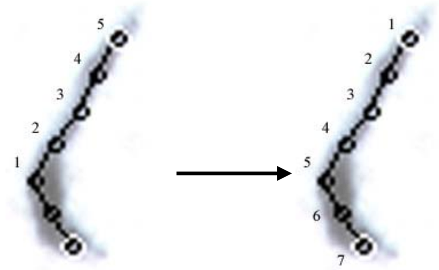

Fig. 1 Point reordering at starting junction point

To trace the signature the algorithm steps on and over the points determined with the algorithm. If a point has more than one possible follow-ups (this is called junction point) then it continues in the direction leading furthest from the previous point and stores the other ones. If there is no acceptable following point then the stored points are looked up, and one of them is chosen. If there are no stored points either, then the algorithm steps on the next component if available. Otherwise, the algorithm is finished. If a component starting point is also a junction point, then the algorithm goes as far as can, then inverts the order of the points of the stroke and continue. This is necessary, because a starting junction point is a fake junction.

A sample run of the algorithm is demonstrated in Fig. 2. 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 [19] and [20].

International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:5, 2007

Fig. 2 strokes of a signature: extracted points (black) and end points of the strokes (white)

IV. Spline Fitting

To make further processing (especially comparison) of signatures computationally feasible, the previous representation of the signature should be simplified. To eliminate stroke points with low significance a spline fitting method was applied.

The algorithm uses the following steps:

1. Pick ***m*** points from the first ***k*** points of a stroke and use them as control points for the spline
2. if the spline deviates from the original curve increase ***m*** (refine spline)
3. if deviation can not be eliminated by increasing ***m***, decrease ***k***. (try with shorter spline)
4. if deviation is within a given threshold, increase ***k*** (try with longer spline)
5. repeat until ***k*** can not be further increased

Deviation is calculated as the sum of distances of the ***k-m*** stroke points, which were not used as control points.

The algorithm is based on the observation that Lagrange curves tend to oscillate when reaching extreme points in the curvature (see Fig. 3). This oscillation is detected by step 3 and will result in the termination of the loop in step 5.

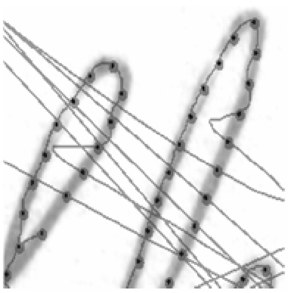

Fig. 3 Oscillation of Lagrange curves

Three different types of splines were tested (Fig. 4). Bezier splines gave the best covering of the original signature, but unfortunately, they only eliminated 10-20% percent of the stroke points. Catmull-Rom performed better (80% elimination) but gave a poor coverage of the original strokes. Lagrange curves seem to have the advantages of both previous spline types without their disadvantages. An 80% reduction of stroke points with an acceptable coverage of the original strokes.

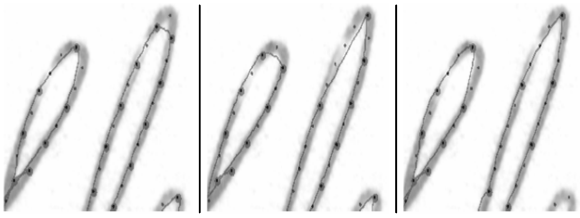

Fig. 4 Bezier spline, Catmull-Rom spline, Lagrange spline

(small dark points represent the stroke points, big dark points represent the control points of the splines)

V. Validation and 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, 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. 5. 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.

Fig. 5 Generated signature

A. Comparison of Stroke Count Deviations

Without going deeper into the semantics, a rough comparison can be given by comparing the numbers of detected strokes on a signature (Fig. 6). There were in average 5.5 strokes in a signature which were detected by our algorithm with a standard deviation of 2,2 This deviation is well balanced, the mean deviation is only 0,02.

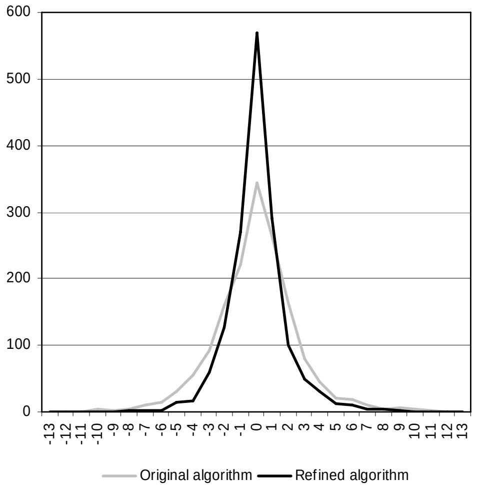

Fig. 6 The distribution of the deviance in the number of strokes of our stroke extraction algorithm compared to the original strokes. (Y-axis: number of signatures, X-axis: deviance)

Differences between the original stroke counts and detected stroke counts can be explained by 3 main factors.

a) errors in stroke detection (not detected strokes and falsely detected strokes)
b) incorrectly directed strokes (original strokes differ from detected strokes in their direction)
c) incorrectly partitioned strokes (original strokes differ from detected strokes in their length)

Error kind a) means major faults in our algorithm, error kind b) means minor faults (correct line detection but wrong detection of direction), error c) is almost insignificant, because it can be compensated with future processing (by connecting some strokes).

B. *Comparison of Detected Strokes and Signatures*

To get an explanation for the differences and refine our stroke extraction algorithm an image registration method was employed. Image registration is the process of determining correspondence between points in two or more images of the same scene. This treatment is used in a large number of different applications, from the medical imaging to the remotely sensed data processing. There are two major categories of the image registration, the area-based and the feature based methods. We used this feature-based, non-rigid image registration for comparing handwritten signatures. The task can be divided into two major components: the extraction of features from images; and the search among the extracted features for the matching pairs that represent the same object in the field of view of the images to be matched. The endpoints of strokes and the connections are identified after a thinning algorithm in a preprocessing step [14]. The endpoints are pixels with one single neighbour connected, which directions are calculated from the first 10 stroke pixels next to the attached endpoint. As a side effect of the thinning algorithm connection points always have three branches, these 3 directions are defined in a similar way as the endpoints (10-10 pixels along branches). For the matching Chui and Rangarajan [23] describe a feature-matching method that is an extension of the iterative closest point (ICP) method. They determined the transformation function and the feature correspondences applying a Robust Point Matching (RPM) algorithm include spline-based deformations.

In our application the main idea was to use an off-line signature database [22], apply the stroke extraction method described in III, fit splines like described in IV, and create artificial images of the signatures with a similar method described at the beginning of this chapter, but this time splines were used instead of straight lines. The images of the generated and the original signatures were compared to detect major differences (mainly of type a. and b.) thereby pointing out the weak points of the stroke detection algorithm.

Measuring the distance between registered signatures is the most critical step of the comparison. There are some different metrics in the literature, in which region based evaluation is applied. The mean squares metric serves our purposes.

Using the results we were able to identify and eliminate some major problems in the stroke detection. The algorithm described in III. is already the refined method, with a much higher accuracy than the original implementation (Fig. 6).

VI. Conclusion and Future work

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] K. K. Lau, Pong C Yuen and Yuan Y Tang, "Recovery of Writing Sequence of Static Images of Handwriting using UWM," Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR’03)
[19] 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
[20] 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
[21] First International Signature Verification Competition, 2004 [Online]. Available: http://www.cs.ust.hk/svc2004/
[22] Off-line signature database, Budapest University of Technology, IT^2, [Online]. Available: http://www.aut.bme.hu/signature
[23] H. Chui and A. Rangarajan, A new point matching algorithm for nonrigid registration, Computer Vision and Image Understanding Volume 89, 2003.