On-line Handwriting Signature Verification Based on Using Extreme Points Extraction

Dr. Rushdi Saleem Abu Zneit
BAU, FET, Computer Engineering
Amman, Jordan
zneit_rush@hotmail.com

Abstract—**This paper presents a method for on-line Handwriting Signature Verification (HSV) using Extreme Points Matching (EPM). EPM does not use direct computation of curve's curvature thus it does not expect smoothness in the analyzing process of the trajectory. In the proposed method the curve's form is described by a small set of extreme points and the method thus seems more precise. Furthermore, it provides an effective preprocessing of the curve and can be utilized in one-to-many pattern matching in a restricted access systems.**

Keywords- line signature verification; HSV; trajectory; extreme points; hand writing; curvature

I. Introduction

Signatures are used legally and financially for identity verification in case of official certificates, contracts, credit cards and checks. Signature verification remains an open challenge as it cannot be considered just as a combination of characters. Automatic signature verification has received extensive research interests in pattern recognition. Signature verification methods are classified [5-6, 8, 12-13] to offline (static) and online (dynamic). In offline methods, signatures are recorded as images, scanned into computers and processed using offline verification stages. Offline signature verification uses static features like shape, style variation, distortion, rotation variation, etc. Online methods, uses dynamic features such as pressure, velocity, stroke length, pen up/down time, etc. and the shape of the signature [12].

One of the key requirements of a signature verification system can be achieved with greater precision in the online signature verification due to the availability of dynamic information system. The intensive developing in computer technology provides the capability to analyze the dynamic drawing motion of handwriting signature as they are written. A special graphical pad provides the ability to trace the hand angle and the motion and pressure of the pen. A hand pad together with an instructed pen [5-6, 8, 13] or a video camera [12] may be used to obtain the online information of pen tip (position, speed, and pressure), therefore, the input may be considered as a sequence of features. Online signature verification has been shown to achieve much higher verification rate than offline verification [5-8, 12-15]. State-of-the-art of online methods achieve error rates (EERs) ranging from 2% to 5% [5-6, 8, 12-13], while the EERs of offline verification are as high as 10%-30% [7, 12]. This difference is largely due to the availability of dynamic information in online methods [5, 14-15].

Hence, the matching and annotation problems for 2-dimensional images are more difficult in signature verification and more time consuming than those for 1-dimensional sequences. Although online verification methods outperform offline ones, their use of special devices for recording the pen-tip trajectory increases their system cost and brings constraints on their applications. However, in some applications such as check transaction and document verification, offline signature is obligatory. In [16], an identification and verification of signatures method is proposed, it is a scale and rotation invariant technique that involves the extraction of rotation invariant sub uniform as a feature vector from the 12 blocks within an image, and it is a combination of chi square distance and linear support vector machine. This paper focuses on online signature verification and on the preprocessing step of curves of online HSV by extracting the extreme points of the curves in signature. HSV has been used earlier in the field of criminal investigations and the start of active research in the field of HSV is established in [1].

II. Problem formulation

In online HVS a set of parametrical sequence given curve $G\left\{x(t), y(t), t \in \left|0, T\right|\right\}$ is considered. Informations included in this timing sequence reflect the dynamic motion of hand muscles and can be used as personal biometrical characteristics. This paper proposes a method that gives the ability to input handwriting signatures to the signature verification system and then the system finds the matching signature from its database and performs the identification. This method of biometrical identification has great functionality and a wide area of implementations. To solve the problem of finding matches from a database, the description of the dynamic curve by a vector of extreme points is proposed. The dynamic curves are parameterized by timing and trajectory as shown in:

$$\left|\frac{\partial G}{\partial s}\right| = 1 \qquad (1)$$

Equation 1 describes motion with constant speed through trajectory which is also applicable for offline parameterization given the following condition:

$$\left|\frac{\partial G}{\partial s} \times \frac{\partial^2 G}{\partial s^2}\right| = 1 \qquad (2)$$

Equation 2 describes the constancy of the parallelogram area constructed on the vectors $\frac{\partial G}{\partial s}$ and boundary $\frac{\partial^2 G}{\partial s^2}$, where s is calculated by [3]:

$$s = \int_0^t \left|\frac{\partial G}{\partial \tau} \times \frac{\partial^2 G}{\partial s^2}\right|^{\frac{1}{3}} \qquad (3)$$

Dynamic curves include information such as pen dynamic motion and timing sequence of points that form the trajectory. To solve the problem of identification (searching etalon), this paper focuses only on the shape of the curve, therefore, the parameterization through the curve is described as:

$$G : \{x(s), y(s), s \in [0, S_G]\} \qquad (4)$$

where the parameter s is calculated by:

$$s(t) = \int_0^t \left\|\frac{\partial G}{\partial \tau}\right\| d\tau \qquad (5)$$

Given that the rounding pen speed is lower than the speed of the straight movements, it is noticed that dynamic hand motion (the speed of writing curves) is tightly close to trajectory's shape. There are different signature elements that may complicate the authentication. In [3], some signature verification methods only considered the shape of the trajectory and ignored time parameterizations and some other parameters such as the pressure of pens and the slope hand. In this paper, only the time parameterizations are ignored as in [2].

III. Comparisons of the Trajectories Based on Extreme Points

Dynamic curves recognition is very close to the continuous human speech recognition. The data derived from the graphic digitizer is similar to a sound signal. As with phoneme speech recognition, objects of handwriting recognition (letters and symbols) are clearly defined. The shape of handwritten signs depends on their position in the word, which is conceptually close to the variation of phonemes in the pronunciation of words depending on their entourage. All these allowed the adaptation of developed methods from the field of speech recognition to the field of handwriting text recognition [4, 5]. However, direct application of these techniques to cursive text recognition is associated with certain difficulties of finding diacritical signs [6]. At the same time, the words in the written language can quite often easily be separated from one another, unlike the sounds in continuous speech.

The main methods of handwritten signature recognizing is a statistical approach based on hidden Markov models, an approach using neural networks, and the comparison of dynamic curves by dynamical transformation of these curves in timescale (dynamic time warping (DTW)). Depending on the principles of the trajectories mapping, there are various methods of DTW: Symmetric Dynamic Time Warping [7], Continuous Dynamic Time Warping [8], Piecewise Aggregate Approximation Dynamic Time Warping [9] and Iterative Deepening Dynamic Time Warping [10].

In most studies [11, 12], trajectory comparisons are exposed to all points of the trajectory. However, it was noted that the DTW method has two major drawbacks [13]: high computational cost and bringing spurious signatures and conform them with the reference signature. To eliminate these drawbacks, a method of comparing the signature based on finding a match between extreme points (Extreme Points Warping (EWP)) was introduced. Many researchers [13, 14] represented the input signal as a series of "peaks" and "valleys" since such a conformity approach [14] needs to find the matching between the points of the same type which significantly accelerate the comparison process. On the basis of the samples of 25 users' signatures, in [13] empirical assumptions were made that there are three types of errors of comparisons: Missing segment at the beginning of a signature, missing segment at the end of a random pair of peek and valley (ripples) with a little difference in height.

To exclude from consideration these points, a heuristic rule is introduced: a point is marked as extreme, if the two following conditions are performed at the same time: $r \geq h_o, d \geq h_o$ where r is the height difference between successive "valley" and "Peak»; *d* is the drop in height between a "peak" and the subsequent "valley»; h_o is a threshold (one pixel). The resulting extreme points are compared by dynamic programming [13, 14]. A direct move diagram when a path from *(i, j)* point can be extended to one of three points *(i+1, j+1), (i+1, j+3), (i+3, j+1)* which gives the minimum penalty of inconsistency. Straight pass allows to find the difference between two curves, the sequence of points *(i, j)* defines the path in which each point is a locally-optimal match between the i^{th} point and its reference (reference points are numbered on the x-axis) and the j^{th} point of the sample (points of sample are numbered on the y-axis). The rules of possible continuation of the path followed by from the fact of interleaving extremes, wherein, an extension is selected, which gives minimum penalty for discrepancy. At the end of the count the table's upper right corner contains the measure of the difference between the two curves. If necessary, the backward pass by using an existing path allows finding the best match of two curves.

This paper proposes an improved version of finding the extreme points [13, 14]. The proposed method takes in consideration some of the characteristics of a handwriting signature as well as the accumulation of errors during the comparison process of successive chords formed by the corresponding extreme points.

IV. Methodology and Results

A. Method

Many handwritten signature verifiers are based on the procedure of finding characteristics or extreme points. In earlier handwritten signature verification methods, points are manually selected, then the coordinates of these points are entered to computers for processing. However, the usage of touch panels and graphical tablets automates this process. However, the concern is to find the most informative points on the curve and those with a value greater than the threshold curvature. The calculation of the curvature of differential geometry requires the existence of second derivation with respect to the time, which puts a limit on the smoothness of the investigated trajectory. Therefore, the curvature is determined by the velocity of the pen using the first derivation as the correlated value [8] at the same time, 15-18% of the average velocity of writing is selected using the threshold value and replacing the limit $c = \lim_{\Delta s=0}\left(\frac{\Delta\alpha}{\Delta s}\right)$, where *Δα* is a change in the angle of the slope of the tangent for the part of trajectory from *s* to *Δs*. The curvature of the trajectory in the i-th point is the set of points $P_{i-2}, P_{i-1}, P_i, P_{i+1}, P_{i+2}$ determined by the angle between two vectors $\overline{P_{i-2}, P_i}$ and $\overline{P_i, P_{i+2}}$. As a variant, it is possible to replace the function of the curvature by its discrete analog time-series of angle change [14].

In this paper, extreme point's definitions that are slightly different from the terminology in the previous section are introduced:

1- Vertical extremum is a point $\{x(t^*), y(t^*)\}$, for which: $\frac{\partial y(t)}{\partial t}|_{t=t^*} = 0$

2- Horizontal extremum is a point $\{x(t^*), y(t^*)\}$, for which: $\frac{\partial x(t)}{\partial t}|_{t=t^*} = 0$

Each vertical exremum defines a local maximum or minimum in the trajectory. The extreme points that respect to curvature, have a local curvature of the trajectory which is larger than the threshold value.

It may be argued, that the above described methods of finding the extreme points are not always optimal, and thus ineffective. However, those vertical and horizontal extremes are closer to the curvature's extreme points. At the same time, incorrect threshold setting for the specific given signature may miss the important points that are not having a large curvature. Figure 1 shows an example of two English letters "L" with different shapes of trajectory. When skipping the point which characterizes the convexity of the pull-down section, the first letter may be erroneously classified as the letter "e". The developed method for finding the extreme points allows tracking changes of the trajectories. This is particularly important in recognition of handwritten letters and words, for example, when the user enters a graphical text on handheld computers.

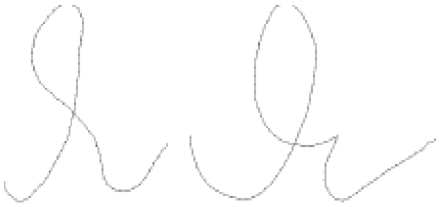

Fig. 1. English letter "l".

In Figure 2, a "skeleton" representation of the same letter is obtained by connecting extreme points, it is clear that all the necessary information about the structure of the letters have been preserved and assigned to a smaller number of points. Figure 3 shows the steps that must be performed to obtain the set of extreme points. Firstly, the original trajectory is smoothed and interpolated (re-sampled) to obtain an equidistant curve. It is known that noise due to digitizing and trembling of the pen points is distributed along the trajectory, as a result, it is desirable to implement a smoothing process on the curve. In this paper, a simple and an effective smoothing method is implemented to the coordinates of adjacent points. Compared to Gaussian smoothing, the implemented smoothing method has a lower computational cost. The implemented smoothing method is based on the following equations that are separately implemented for each of the coordinates:

$$X_{smoothed}(i) = 0.25X(i-1) + 0.5X(i) + 0.25X(i+1)$$
$$Y_{smoothed}(i) = 0.25Y(i-1) + 0.5Y(i) + 0.25Y(i+1) \quad (6)$$

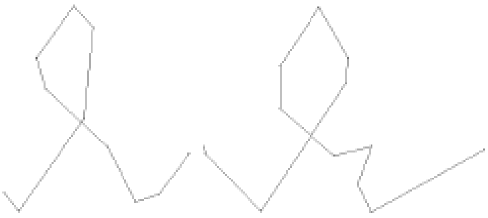

Fig. 2. Example of finding extreme points.

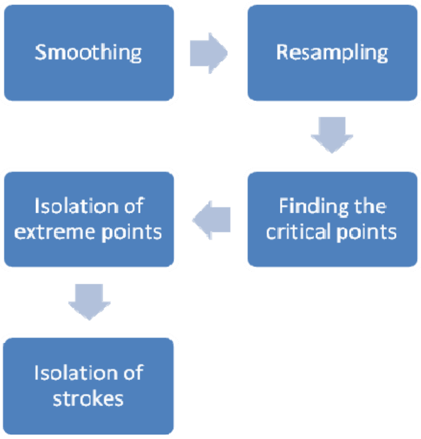

Fig. 3. Steps of searching extreme points.

The resulting representation of the curve is used to find the critical points. It should be noted that the term "critical point" refers to the pre-found extreme points, as then some of these points will be filtered. Unlike [5], this paper does not exclude endpoints with a small drop in height, however, such points form a so-called "plateau" (see Figure 4). After smoothing and resampling, finding the critical points step takes place. The critical points that are important in this step are: the endpoints of the curve segments, vertical and horizontal extremes, the boundary points of the vertical plateau, and the point of inflection. For finding the vertical extremes, a simple heuristic is applied: point P_i is the extreme point if the following condition is true: $\left(P_{i-1}^{y}-P_{i}^{y}\right)\left(P_{i}^{y}-P_{i+1}^{y}\right)\prec 0$ if for some i^{th} point the above equation drawn to zero, and the coordinates of three consecutive points do not match, a point was added to the set as a boundary point of the plateau. Similarly, there were also horizontal extremes (critical point). Since special attention was paid to the speed and efficiency of the algorithm. For infliction points we analyze the sign of vector product of $\operatorname{sgn}\left\lfloor \overline{P_{i-2}P_{i}\times P_{i+2}P_{i}} \right\rfloor$ the last expression in this case can be replaced by a simplified coordinate entry to the flat case:

$$\operatorname{sgn}\left[\left(P_{i-2}^{x}-P_{i}^{x}\right)\left(P_{i+2}^{y}-P_{i}^{y}\right)-\left(P_{i-2}^{y}-P_{i}^{y}\right)\left(P_{i+2}^{x}-P_{i}^{x}\right)\right] \quad (7)$$

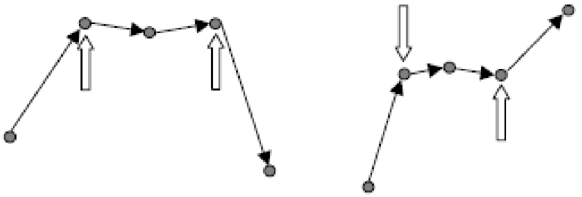

Fig. 4. The horizontal plateau

So, after we finish this step:

- We obtain a set of critical points, which, strictly speaking, are redundant. Depending on the parameters of the program the number of points obtained in the 2-2.5 times higher than the finally found amount of the extreme points.
- The next step is to remove the extra points, and the addition of new ones, that more accurately characterize the shape of the trajectory. Particularly, all the inflection points must be deleted, because usually they are located in the middle of the "straight" section, which reduces their informative value. Instead of that, the set of extreme points, conventionally called the curvature extremes are added, although the curvature value at these points cannot be very large. The basis of the filtering procedure supposed the following mere:

In horizontal plateau, Critical points are the features of the plateau. Here we consider that quantities $dy_{prev}=P_{i}^{y}-P_{i-1}^{y}\ and\ dy_{next}=P_{i+1}^{y}-P_{i}^{y}$, where i^{th} point belong to the plateau if and only if $dy_{prev}=0\ or\ dy_{next}=0$. Suppose that for the sequential viewing points on the curve for i^{th} point $dy_{next}=0$, then for the point *i+1* $dy_{pre}=0$, which means that i^{th} and *(i+1)*th points belong to the same plateau. The above allows to formulate selection rules for the plateau points P_{max} and P_{min} where P_{min} means that the point may be a local minimum and P_{max} means that the point may be a local maximum. In this notation, Figure 4 presents variant plateau $\{P_{\max},P_{\max}\}\ and\ \{P_{\min},P_{\min}\}$. If we denote by the letter V the vertical extremum, and the letter C the extremum of the curvature, the selection rules are written in the following form:

$$rule1:\{P_{\max},P_{\max}\}\Rightarrow\{V_{\max}\}\ or\ \{P_{\min},P_{mi}\}\Rightarrow\{V_{\min}\}$$

if the size of plateau less than threshold value;

$$rule2:\{P_{\max},P_{\max}\}\Rightarrow\{C,V_{\max},C\}\ or\ \{P_{\min},P_{\min}\}\Rightarrow\{C,V_{\min},C\}$$

if the size of plateau greater than threshold value;

$$rule3:\{P_{\min},P_{\max}\}\Rightarrow\{\theta\}\ or\ \{P_{\max},P_{\min}\}\Rightarrow\{\theta\}$$

if the size of plateau less than threshold value;

$$rule4:\{P_{\min},P_{\max}\}\Rightarrow\{C,C\}\ or\ \{P_{\max},P_{\min}\}\Rightarrow\{C,C\}$$

if the size of plateau greater than threshold value;

Similar rules can be formulated for the case of horizontal extremums. The algorithm is based on the comparison of the curves on the idea of finding matching between the vertical extreme points. The algorithm is based on local features of the curve, which can minimize the loss of important information. In Figure 5 an example of a signature for which the generated view is not accurate because boundary point of the horizontal T-stroke does not indicate the presence of extreme vertical or plateau is shown (although this is not the fault of the program). It happened due to trembling in the exercise of that stroke, and due consideration that specified the local neighborhoods of the critical points. But this can lead to an error, since the matching algorithm used is focused on locating (finding) pairs of vertical extremes of the opposite type. In order to avoid unexpected errors, the last stage of the process of marking up of the signature by using extreme points is the secretion of horizontal portions. At this stage the final horizontal strokes are marked if they are presented in the signature.

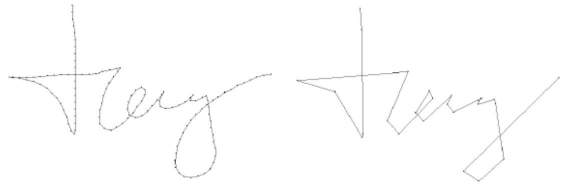

Fig. 5. Example signature with horizontal stroke.

B. Experiment Results

The proposed on-line signature verification system was applied to 200 genuine signatures from 50 users, 150 skilled forgeries from 75 forgers and 150 random signatures. Table I shows the on-line signature verification results for the proposed method. We used two parameters to evaluate the performance of signature verification systems, namely: False Acceptance Rate (FAR) and False Reject Rate (FRR). As it can be seen from Table I the performance for the genuine signatures was FAR=0.5%. For skilled forgeries and random forgeries: FRR=4% and 0% respectively. Comparing these results with the other methods of verification in the literature, our proposed method seems more accurate and it highly succeeded in verifying the online signatures.

TABLE I. RESULTS

Type of Signature	Tested	Accepted	Rejected
Genuine signatures	200	199	1
Skilled forgeries	150	6	144
Random forgeries	150	0	150

V. CONCLUSION

In this paper, a method for efficient on-line signature verification is proposed, so as to increase security in access control and financial transactions. In this proposed method, finding the extreme points is used to speed up the comparison of the dynamic curves. In contrast to the existing alternative approaches, the proposed algorithm takes into account the particular shape of the trajectory. Moreover, the acceptance rate for genuine signatures and the rejection rate for forgeries are found high enough. Compared to other signature verification algorithms, the proposed method seems more accurate and efficient.

REFERENCES

[1] R. Plamondon, G. Lorette, “Automatic signature verification and writer identification – the state of the art”, Pattern Recognition, Vol. 22, No. 2, pp. 107–131, 1989

[2] M. E. Munich, P. Perona, “Visual signature verification using affine arc-length” Conference on Computer Vision and Pattern Recognition CVPR, pp. 2180-2186, 1999

[3] V. S. Nalwa, “Automatic on-line signature verification”, Proceedings of the IEEE, Vol. 85, No. 2, pp. 215-239, 1997

[4] T. Starner, J. Makhoul, R. Schwartz, G. Chou, “On-line cursive handwriting recognition using speech recognition methods”, IEEE Conference on Acoustics, Speech, and Signal Processing, Vol. 5, pp. 125-128, 1994

[5] L. Yang, B. K. Widjaja, R. Prasad, “Application of hidden Markov models for signature verification”, Pattern Recognition, Vol. 28, No.2, pp. 161-170, 1995

[6] G. Seni, J. Seybold, “Diacritical processing for unconstrained on-line handwriting recognition using forward search”, International Journal on Document Analysis and Recognition, Vol. 2, No. 1, pp. 24-29, 1999

[7] T. Hastie, E. Kishon, “A model for signature verification”, IEEE International Conference on Decision Aiding for Complex Systems, pp. 191-196, 1991

[8] M. E. Munich, P. Perona, “Continuous dynamic time warping for translation-invariant curve alignment with applications to signature verification”, 7th IEEE International Conference on Computer Vision, Vol. 1, pp. 108-115, 1999

[9] E. Keogh, M. Pazzani, “Scaling up dynamic time warping for datamining applications”, 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 285-289, 2000

[10] S. Chu, E. Keogh, D. Hart, M. Pazzani, “Iterative deepening dynamic time warping for time series” IEEE International Conference on Data Mining, Maebashi City, Japan, 2002

[11] A. K. Jain, F. D. Griess, S. D. Connell, “On-line signature verification”, Pattern Recognition, Vol. 35, No. 12, pp. 2963–2972, 2002

[12] S. D. Connell, A. K. Jain, “Template-based online character recognition”, Pattern Recognition, Vol. 34, No. 1, pp.1-14, 2001

[13] F. Hao, C. W. Chan, “Online signature verification using a new extreme points warping technique”, Pattern Recognition Letters, Vol. 24, No. 16, pp. 2943-2951, 2003

[14] X. Li, M. Parizeau, R. Plamondon, “Detection of extreme points of on-line handwritten scripts”, Progress in Handwriting Recognition, World Scientific, 1997

[15] R. C. Sonawane, M. E. Patil, “An effective stroke feature selection method for online signature verification”, 3rd International Conference on Computing Communication & Networking Technologies (ICCCNT), pp. 1-6, Coimbatore, India, July 26-28, 2012

[16] S. Singh, A. Kaur, “Off-Line signature verification using sub uniform local binary patterns and support vector machine”, International Conference on Chemical Engineering & Advanced Computational Technologies (ICCEACT’2014), Pretoria, South Africa, Nov. 24-25, 2014