Online Stroke Modeling for Handwriting Recognition

Oleg Golubitsky Stephen M. Watt

Department of Computer Science
University of Western Ontario
London Ontario, Canada N6A 5B7
{oleg,watt}@csd.uwo.ca

Abstract

The process of recognizing individual handwritten characters is one of classifying curves. Typically, handwriting recognition systems—even "online" systems—require entire characters be completed before recognition is attempted. This paper presents another approach for real-time recognition: certain characteristics of a curve can be computed as the curve is being written, and these characteristics are used to classify the character in constant time when the pen is lifted. We adapt an earlier approach of representing curves in a functional basis and reduce real-time stroke modelling to the Hausdorff moment problem.

1 Introduction

A difficulty with most handwriting recognition techniques is that entire curve traces must be obtained before any analysis takes place. This is true even for so-called "online" methods, which recognize one character at a time. This means that recognition on a pen-enabled device, such as a Tablet PC or PDA, typically proceeds in alternating phases with the processor mostly idle while collecting input and with the user waiting while characters are being recognized.

This paper takes a different point of view and asks to what degree it is possible to do useful computation as the digital ink is being written. The initial application is to the problem of mathematical handwriting recognition, where characters are typically written individually and segmentation is not an issue. Our approach is also applicable to other real-time problems such as recognition of other printed handwriting, recognition of entire cursively written words and other classification problems for curves.

Earlier work [2] has shown how coordinate curves $X(t)$, $Y(t)$ for handwritten characters can be modeled succinctly by truncated Chebyshev series and that the series coefficients can be used to classify characters and for recognition. Rather than describing a digital ink trace by a few hundred coordinate values, instead a visually indistinguishable curve can be modeled by twenty series coefficients. If necessary, one can apply Euclidean or affine transformations directly to the polynomial parametrizations of the coordinate curves in time proportional to the degree, faster than to the original sample points. A secondary benefit of this representation is that the geometry of these curves can be analyzed by general analytic techniques rather than *ad hoc* numerical techniques.

The present work starts with a similar approach, but uses a different functional basis allowing useful computation as the curve data is received. The basic idea is to compute moments of the coordinate curves in real time as the stroke is written and then to construct the coefficients for a Legendre series representation of the curves in constant time when the pen is lifted.

We find that the Legendre series representation is just as suitable in practice for representation and analysis of ink traces as the Chebyshev representation, but has the benefit that it can be computed in a small, fixed number of arithmetic operations at the end of a stroke. Additionally, the fixed number of arithmetic operations at each time step (to compute the moments) and on pen up (to compute the series coefficients) makes this technique well suited to inexpensive hardware implementation.

As detailed further in the article, none of the computation during the stroke trace is wasted. This contrasts with other speculative analysis techniques that can give a "best guess so far" as a curve is traced, but whose results can be invalidated as the curve shape evolves. Doing only useful computation can allow for better overall system performance on slow devices and conservation of battery power.

The method we present is "online" in two senses: It is online in the usual sense of character-at-a-time recognition, in contrast to page at a time "offline" document analysis. It is also online in a theoretical sense, computing its result as stroke data is received with no look-ahead. To aid in the evaluation of such methods, we formalize the idea of requiring computation to occur as data is collected and then separately measuring the time required after all data is seen.

The contributions of this paper are

- a complexity model for the on-line recognition problem,
- an analytic representation for digital ink that may be computed online as the stroke is written, requiring only a constant number of operations for each sample point, and
- experimental results that show this representation is as good as that obtained by earlier stroke-at-once methods.

The rest of the paper is organized as follows: Section 2 gives the background context. Section 3 introduces our complexity model that formalizes the notion that computing during pen movement is in some sense free, while time from pen up is precious. Section 4 states in an exact way the problem of whether there is a functional representation that could be computed stroke-online. Section 5 presents a numerical integration method to compute the moments of an input signal. Section 6 shows how the Legendre series coefficients may be computed in constant time from moments that are integrated on-line. Section 7 summarizes our experimental results on the quality of approximation by this method. Section 8 summarizes the main results, outlines future work and concludes the paper.

2 Preliminaries

We present several basic ideas that are used throughout the paper.

Series of orthogonal functions

We say a family of functions $\{h_i\}$ is orthogonal with respect to a functional inner product $\langle \cdot, \cdot \rangle$ with weight $w(t)$ on domain $[a, b]$ if

$$\langle h_i, h_j \rangle \equiv \int_a^b h_i(t) h_j(t) w(t) \mathrm{d}t = 0 \quad \text{when } i \neq j.$$

Under certain conditions, a function f of some general class may be expressed as a linear combination of elements of the family $\{h_i\}$. In this case we call the family a functional basis, and for many classes of functions this basis is countably infinite, so we may write

$$f(t) = \sum_{i=0}^{\infty} \alpha_i h_i(t).$$

For an orthogonal functional basis, the coefficients α_i may be computed by inner products

$$\alpha_i = \langle f, h_i \rangle / \langle h_i, h_i \rangle$$

because

$$\begin{aligned} \langle f, h_i \rangle &= \Big\langle \sum_{j=0}^{\infty} \alpha_j h_j(t),\ h_i(t) \Big\rangle \\ &= \sum_{j=0}^{\infty} \alpha_j \langle h_j, h_i \rangle = \alpha_i \langle h_i, h_i \rangle. \end{aligned}$$

Chebyshev representation

Earlier work [2] showed how the X and Y coordinate functions of handwritten characters could be written as truncated series of Chebyshev polynomials of the first kind, $T_n(t) = \cos(n \arccos t)$. The T_n are orthogonal on $[-1, 1]$ for weight $w(t) = 1/\sqrt{1-t^2}$. It was found that the Chebyshev series coefficients provided a succinct characterization of the coordinate functions. For series truncated at order d, the approximations were

$$X(t) \approx \sum_{i=0}^{d} \alpha_i T_i(t), \quad Y(t) \approx \sum_{i=0}^{d} \beta_i T_i(t).$$

Classification was obtained by measuring the Euclidean distance from the point $(\alpha_0, \ldots, \alpha_d, \beta_0, \ldots, \beta_d)$ to centers of clusters obtained from handwriting samples.

Moments and the Hausdorff problem

The moments of a function f defined on the interval $[a, b]$ are the integrals

$$\mu_k = \int_a^b t^k f(t) \mathrm{d}t.$$

Central moments replace t^k by $(t - \mu_0)^k$ in the integrand and are sometimes more useful.

A key aspect of our approach is to recover a function from its moments. This is the Hausdorff moment problem [4, 5], known to be ill-conditioned. It has been determined that a function can be reliably recovered from the moments only if it is well approximated by the first few orthogonal polynomials (*e.g.* order $\leq$ 15 with double precision). An algorithm for computing the coefficients of the Legendre series from moments has been proposed in [7] together with an analysis of accuracy and stability. As the order increases, accuracy will also increase, but stability will decrease. This algorithm is essentially the same as the step we perform when the pen is raised. Improvements of stability have been proposed in [6] and in [3]. In the first paper the order is decreased using fractional moments, and in the second paper the Hankel matrices that arise in the Hausdorff moment problem are preconditioned.

3 Complexity Model

We wish to capture the notions of useful computation that can be performed as an ink trace is collected and the remaining computation necessary when the pen is lifted.

We consider a general model of online computational complexity where input is received over a span of time during which computation can occur. The computation may use only the input seen so far, with no look-ahead. After receiving the last input, further computation can occur and the output can be given. We do not specify the underlying model of computation (*e.g.* k-tape Turing machine, RAM, *etc.*). Machine-specific models of real-time computation have been studied dating back to the beginnings of complexity theory [8] and the subject continues to be of interest, *e.g.* [1].

We consider the input to be a sequence of N values received over time at a uniform rate. If the real problem does not give its values uniformly over time, then dummy values can be inserted. The size of the problem is N and, because the input values are received uniformly, input values can also be taken to mark time steps. We characterize the time for a problem or of an algorithm by giving the number of computational steps at the n-th input and after the last input. If $T_\Delta(n)$ operations are required at the n-th input step and additionally $T_F(n)$ are required after the last input (when $n = N$), then we write the online time complexity as $\mathrm{OL}_n[T_\Delta(n), T_F(n)]$. For example, linear insertion sort takes online time $\mathrm{OL}_n[O(n), 0]$. Building a heap and printing it requires time $\mathrm{OL}_n[O(\log n), O(n)]$. Similar definitions can be made for average online time, maximum online time, online space complexity, average online energy consumption, *etc.* An algorithm that takes online time $\mathrm{OL}_n[T_\Delta(n), T_F(n)]$ takes total time $\sum_{i=0}^{N} T_\Delta(i) + T_F(N)$.

If an algorithm uses time $\mathrm{OL}_n[T_\Delta(n), T_F(n)]$, where T_Δ and T_F do not depend on n, then we say the algorithm takes *online constant time*. The functions T_Δ and T_F may depend on other parameters of the problem, but not n.

For digital ink, n gives the number of points seen so far in an ink trace. We are interested in what features we can characterize in constant online time and space as a stroke is written.

4 Series Representation for Online Computation

Motivated by the approach of Char and Watt and wishing to find a true online method, we pose the following question:

Problem *Is there a functional basis for digital ink traces for which the series coefficients can be computed in online constant time? If so, show how the coefficients can be computed.*

We answer this question positively.

One of the obstacles to computing Chebyshev series directly in a stroke-online manner is the algebraic form of their weight function $w(t) = 1/\sqrt{1-t^2}$. We therefore consider instead the Legendre polynomials, a family of orthogonal polynomials with the simplest possible weight function.

The Legendre polynomials may be defined as

$$P_n(t) = \frac{1}{2^n n!}\frac{\mathrm{d}^n}{\mathrm{d}t^n}(t^2-1)^n. \qquad (1)$$

These are orthogonal on $[-1,1]$ with respect to the weight function $w(t) = 1$. These are usually normalized so that $P_n(1) = 1$ and we have

$$\langle P_i, P_j\rangle = \int_{-1}^{1} P_i(\tau)P_j(\tau)\mathrm{d}\tau = \frac{2}{2n+1}\delta_{ij},$$

where $\delta_{ij} = 1$ if $i = j$ and zero otherwise.

When we collect digital ink we receive function values for the coordinates $x_i(\lambda)$ for values of the parameter λ from 0 to some point L. Typically L is known only at the time the pen is lifted.

We show in the following sections how a function defined on $[0, L]$ can be scaled to domain $[-1, 1]$ and range $[a, b]$ and can have its Legendre series coefficients computed to solve our stated problem. We prove the following:

Theorem (Online Legendre series).
Given a sequence of n values of a function $f(\lambda)$ at equally spaced values of λ, received one at a time, it is possible to compute the first d coefficients of the truncated Legendre series for f, normalized to a desired range and domain, in online time $\mathrm{OL}_n[O(d), O(d^2)]$.

We then show by experimental results how representing characters as truncated Legendre series provides models of the same practical quality as the Chebyshev series used previously.

5 Numerical Moment Computation

For our purposes, the moments of a function f are defined over an unbounded half-line since the curve may be traced over an arbitrary length:

$$\mu_k(f,\ell) = \int_0^{\ell} \lambda^k f(\lambda)\mathrm{d}\lambda. \qquad (2)$$

The moments of the function may be determined on-line numerically either by hardware integrators or a software implementation.

In our application we assume that discrete sample values of f are received as a real-time signal. We use these values to compute approximate values for the moment integrals. To have a constant bound on the number of arithmetic operations for each sample value of f, we use a finite order integration method. We assume the sample points are equally spaced with $\Delta\lambda = 1$.

Through numerical experiments with curves having 200 sample points we have found that the standard first and second order Newton-Cotes integration (trapezoids and Simpson's rule) do not yield sufficiently accurate values for the moments μ_k as k increases. Furthermore, methods that require *a priori* knowledge of L cannot be applied in this context. We have found that instead a specialized first order integration method gives suitable results.

To compute the integral $\int_0^L \lambda^k f(\lambda)\mathrm{d}\lambda$, we use the following approximation for each interval $[i, i+1]$ and sum the results:

$$\int_i^{i+1} \lambda^k f(\lambda)\mathrm{d}\lambda \approx \frac{(i+1)^{k+1} - i^{k+1}}{k+1} \times \frac{f(i+1)+f(i)}{2}. \qquad (3)$$

This formula represents the exact integral of λ^k on $[i, i+1]$ multiplied by the mean value of the function at the end points.

Combining formulas (3) for all i from 0 to $L-1$ and rearranging the terms, we obtain the following integration formula:

$$\int_0^L \lambda^k f(\lambda) d\lambda \approx \sum_{i=1}^{L-1} \frac{i^{k+1}}{k+1} \times \frac{f(i-1)-f(i+1)}{2} + \frac{L^{k+1}}{k+1} \times \frac{f(L-1)+f(L)}{2}.$$

We can update the the sums $\sum_{i=1}^{L-1}(\ldots)$ for all orders 0 to d online, as soon as the value $f(i+1)$ is available, in $2(d+2)$ arithmetic operations. Thus, the online complexity is

$$T_\Delta = 2(d+2),$$

which is constant in the number of points L. We then need $2(d+2)$ additional operations to compute the last term when the value $f(L)$ arrives and pen is lifted up, which is part of the computation after the last input analysed in the next section.

This integration method can easily be adapted for points non-uniformly separated in time or parametrized by arc-length rather than time. Since parametrization by arc length is intrinsic to the curve (i.e., not affected by the variations in the speed with which the curve is traced out) and invariant with respect to the Euclidean transformations, it may be preferable to the time parametrization for the purpose of character recognition and classification. Also, higher order methods based on this idea can be used instead if desired.

6 Legendre Coefficients from Moments

After a curve is traced, we will have computed its moments over some length L, with L known only at the time the pen is lifted. The problem is now to scale L to a standard interval and compute the truncated Legendre series coefficients for the scaled function from the moments of the unscaled function, $\mu_k(f, L)$. We show how this can be done, and that it requires a number of arithmetic operations depending only on the desired number of terms of the Legendre series.

Suppose $f(\lambda)$ is defined on $[0, L]$ and let $\hat{f}(\tau) = f\big((\tau+1)L/2\big)$ be the scaled version defined on $[-1, 1]$. If $\hat{f}(\tau) = \sum_{k=0}^{\infty} \hat{\alpha}_k P_k(\tau)$, we may compute the $\hat{\alpha}_k$ as follows:

$$\begin{aligned}
\hat{\alpha}_k &= \big(k+\frac{1}{2}\big) \int_{-1}^{1} \hat{f}(\tau) P_k(\tau)\, \mathrm{d}\tau \\
&= \frac{2k+1}{L} \int_0^L f(\lambda) P_k(2\lambda/L - 1)\, \mathrm{d}\lambda \\
&= \frac{2k+1}{L} \sum_{i=0}^{k} [t^i] P_k(2t-1) \int_0^L f(\lambda)\, (\lambda/L)^i\, \mathrm{d}\lambda \\
&= \frac{2k+1}{L} \sum_{i=0}^{k} \frac{[t^i] P_k(2t-1)}{L^i} \mu_i(f, L).
\end{aligned}$$

Here $[t^i](\cdot)$ denotes the coefficient of t^i. The polynomials $\tilde{P}_k(t) = P_k(2t-1)$ are known as the "shifted Legendre polynomials" and are given explicitly by

$$\tilde{P}_n(t) = (-1)^n \sum_{i=0}^{n} \binom{n}{i} \binom{n+i}{i} (-t)^i.$$

We therefore have

$$\hat{\alpha}_k = (-1)^k \frac{2k+1}{L} \times \sum_{i=0}^{k} \left(\frac{-1}{L}\right)^i \binom{k}{i} \binom{k+i}{i} \mu_i(f, L) \quad (4)$$

Note that the coefficients $(-1)^i \binom{k}{i} \binom{k+i}{i}$ are independent of the problem and may be computed as constants in advance.

Given the first k moments of f and the first $k-1$ powers of L, we may compute $\hat{\alpha}_k$ and L^k in a number of arithmetic operations depending only on k: Computing L^k can be done with one multiplication. The first term of the sum is given. Adding each of the k subsequent terms to the sum can be done with three operations (one division, one multiplication and one addition). Multiplication by $(2k+1)/L$ can be done with two additional operations if $2k+1$ is a pre-computed constant. In all, we have $3k+3$ operations to compute α_k and L^k.

All the coefficients of a truncated Legendre series of order d from can be computed from the

moments with $\sum_{i=0}^{d}(3i+3) = 3(d+2)(d+1)/2$ arithmetic operations. This is constant in the number of points, L. Since the required approximation order is fixed and small (typically less than 10), the quadratic appearance of d can be ignored.

To scale $\hat{f}(\tau)$ to range over $[a, b]$ on domain $[-1, 1]$, we record $f_{\min}$ and $f_{\max}$, the minimum and maximum values attained by $f(\lambda)$. We then define

$$\begin{aligned}\hat{\hat{f}}(\tau) &= \frac{b-a}{f_{\max}-f_{\min}}\hat{f}(\tau) + \frac{af_{\max}-bf_{\min}}{f_{\max}-f_{\min}} \\ &= \sum_{i=0}^{\infty}\hat{\hat{\alpha}}_i P_i(\tau),\end{aligned}$$

where

$$\hat{\hat{\alpha}}_i = \hat{\alpha}_i\frac{b-a}{f_{\max}-f_{\min}} + \delta_{i0}\frac{af_{\max}-bf_{\min}}{f_{\max}-f_{\min}}. \quad (5)$$

This takes an additional $d + 7$ operations, so we have

$$T_F = \frac{3}{2}d^2 + \frac{11}{2}d + 10.$$

7 Experimental Results

Our first experiment asks how well the truncated Legendre series computed from moments approximate the coordinate functions $X(t)$ and $Y(t)$ of handwritten character curves. The samples comprised 987 single stroke characters from a collection of 239 mathematical symbols taken from 9 test users. Users provided samples only for the symbols with which they were familiar and which they would normally write from time to time.

We computed the moments of their coordinate functions using the algorithm from Section 5, and then the coefficients of the truncated Legendre series using the algorithm from Section 6. We estimated the root mean square deviation

$$\mathrm{RMS}(\phi) = \sqrt{\sum_{t=0}^{L}\frac{||(X(t),Y(t)) - \phi(t)||_2{}^2}{L+1}}$$

between the handwritten character curves $(X(t), Y(t))$ scaled to range $[0, 1]$ and the corresponding truncated Legendre series $\phi(t)$ of degrees 3, 4, 6, 8, 10, 15, 17, 18, 19 computed with double precision. The resulting distribution of RMS, shown in Table 1 and Figure 1, is nearly identical to the one obtained for truncated Chebyshev series in [2]. The mean square deviation decreases until degree 16–17, after which it begins to increase due to the fact that higher degree coefficients are computed inaccurately with double precision.

We repeated the same computations for coordinate functions parametrized by arc length. As Table 2 and Figure 3 show, the RMS errors for parametrization by arc length are similar to those by time.

An example of a character and its approximations by truncated Legendre series is shown in Figure 2. The time-parametrized approximation is indistinguishable from the original already for degree 8, unlike the one parametrized by arc length, which tends to cut the singular points (cusps and corners) of the original curve even for the optimal degree 17. The reason for such behavior is that the coordinate functions X and Y parametrized by time are smooth everywhere, and hence better approximated by polynomials, than the coordinate functions parametrized by arc length, which are singular at the cusp (see Figure 4). The mismatch between the character curve and its approximations by arc length parametrized polynomials is not reflected in the mean square errors, because around the singular points (where the curves disagree most) the differential of the arc length vanishes, and hence the contribution of this region to the RMS is small regardless of the accuracy of approximation there.

The computations were carried out with double precision floating point numbers. To estimate round-off errors, we also computed the expansion coefficients exactly, using rational arithmetic, and found the maximal absolute and average relative errors (defined as the sum of absolute errors divided by the sum of absolute values), summarized in Table 3. These results show that double precision floating point provides sufficient accuracy for computing coefficients up to degree 13–14, even though the errors grow exponentially with the degree.

The maximal absolute value that appears during the computations for degree 10 is 4×10^{31}, and the minimal non-zero absolute value

Degree	$< 90\%$	$< 95\%$	$< 99\%$
3	.310	.330	.361
4	.240	.264	.323
6	.149	.175	.234
8	.075	.099	.153
10	.034	.048	.097
15	.012	.016	.040
17	.011	.014	.044
18	.013	.017	.060
19	.041	.067	.181

Table 1: *Legendre series RMS cutoffs by degree (curves parametrized by time).*

Degree	$< 90\%$	$< 95\%$	$< 99\%$
3	.279	.300	.340
4	.189	.223	.292
6	.106	.135	.173
8	.057	.073	.115
10	.036	.047	.074
15	.020	.024	.044
17	.017	.021	.049
18	.017	.021	.067
19	.038	.054	.221

Table 2: *Legendre series RMS cutoffs by degree (curves parametrized by arc length).*

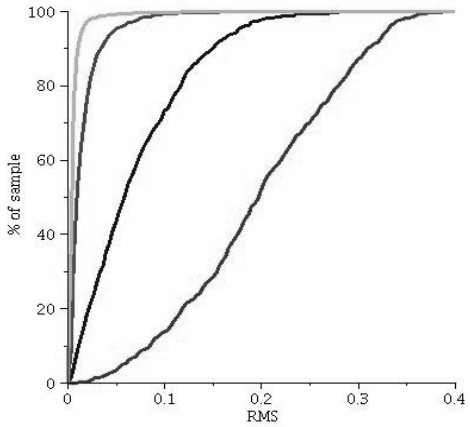

Figure 1: *RMS error for time parametrization. % of sample with* RMS *error* $\leq$ *given amount for series truncated at degrees 3, 6, 10, 15. Curves are in order by degree, with 3 lowest.*

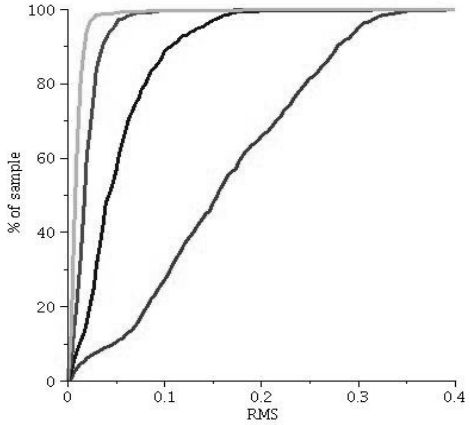

Figure 3: *RMS error for arc length parametrization. % of sample with* RMS *error* $\leq$ *given amount for series truncated at degrees 3, 6, 10, 15. Curves are in order by degree, with 3 lowest.*

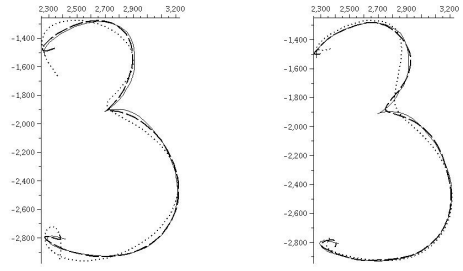

Figure 2: *Approximations of character "3" (solid) with Legendre series. Left: time parametrization, degrees 7 (dotted) and 8 (dashed). Right: arc length parametrization, degrees 8 (dotted) and 17 (dashed).*

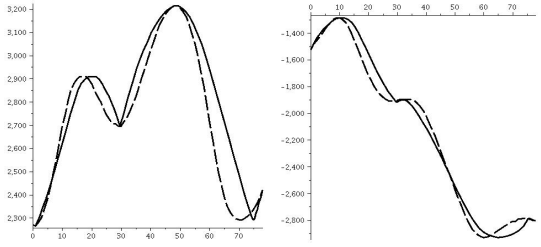

Figure 4: *Coordinate functions X (left) and Y (right) parametrized by time (dashed) and arc length (solid). The parameters are scaled to span the same interval.*

Degree	Abs. Error	Rel. Error
0	1×10^{-11}	7×10^{-18}
1	4×10^{-11}	4×10^{-14}
2	5×10^{-10}	3×10^{-13}
3	2×10^{-9}	2×10^{-12}
4	1×10^{-8}	1×10^{-11}
5	5×10^{-8}	4×10^{-11}
6	3×10^{-7}	5×10^{-10}
7	2×10^{-6}	3×10^{-9}
8	9×10^{-6}	2×10^{-8}
9	5×10^{-5}	2×10^{-7}
10	3×10^{-4}	2×10^{-6}
11	1×10^{-3}	1×10^{-5}
12	1×10^{-2}	1×10^{-4}
13	6×10^{-2}	9×10^{-4}
14	3×10^{-1}	5×10^{-3}
15	2×10^{0}	4×10^{-2}
16	1×10^{1}	3×10^{-1}

Table 3: *Maximal absolute and relative round-off errors for coefficients by degree.*

is 1.0. Therefore, if we were to perform the computations in fixed point arithmetic, 128 bits would be required to compute expansions of degree 10. With 64-bit fixed point arithmetic, only expansions of degree 4 can be computed, which do not approximate the character curves accurately enough. Thus, floating point numbers are more adequate for our purpose.

Floating point computations with parametrization by arc length, instead of time, cause similar round-off errors (not shown here). Since arc length is usually an irrational number, we could not compute the expansions with respect to arc length exactly with rational arithmetic and used quadruple precision (30 decimal digits) floating point numbers instead.

We conclude that truncated Legendre series computed from moments with double precision approximate handwritten character curves just as closely as truncated Chebyshev series of the same degree; and the latter, as has been shown in [2], are visually indistinguishable from the samples for degree 10. In addition, our experiment has confirmed that for such small degrees computation of Legendre coefficients from moments can be carried out reliably with double precision, even though in general this problem is ill-conditioned.

8 Conclusions and Future Work

This article has examined the problem of online stroke analysis. We have presented a framework in which the computational complexity of these online algorithms can be meaningfully modeled, characterizing computation that can be performed as data is received.

We have demonstrated that a truncated Legendre series representation of ink strokes can be computed in online time that is constant with respect to the number of data points, normalizing the domain and range of the coordinate functions as desired.

We have shown experimentally that this Legendre series representation provides models of the same quality as earlier work with truncated Chebyshev series. That is, for most characters, their approximations by Legendre series of order 10 look indistinguishable from the originals.

Having established the utility of this representation, it is left to future work to derive further characteristics of the ink strokes from the analytic representation, to build stroke classifiers (*e.g.* using cluster proximity or SVMs) based on the series coefficients, and to compare them with other methods (such as elastic matching) in accuracy and speed.

Acknowledgements

We thank Bruce Char for the Maple worksheet to compute RMS errors. Stephen Watt thanks NSERC, the Fields Institute, the MITACS Network of Centres of Excellence, Microsoft Canada and Maplesoft for support.

About the Authors

Oleg Golubitsky is a postdoctoral fellow at the Ontario Research Centre for Computer Algebra (ORCCA) at the University of Western Ontario. His interests lie in the areas of mathematical computing, with recent attention to algorithms for differential ideals.

Stephen M. Watt is a professor at the University of Western Ontario and director of ORCCA. His research interests lie in the areas of computer algebra, pen-based computing and programming languages.

References

[1] S.D. Bruda and S. G. Akl. Real-time computation: A formal definition and its applications. *Journal of Computers and Applications*, 25:247–257, 2003.

[2] Bruce W. Char and Stephen M. Watt. Representing and characterizing handwritten mathematical symbols through succinct functional approximation. In *Proc. International Conference on Document Analysis and Recognition (ICDAR)*, pages 1198–1202. IEEE Computer Society, 2007.

[3] D. Fasino. Spectral properties of Hankel matrices and numerical solutions of finite moment problems. *Journal of Computational and Applied Mathematics*, 65(1-3):145–155, 1995.

[4] Felix Hausdorff. Summationsmethoden und Momentfolgen. I. *Mathematische Zeitschrift*, 9:74–109, 1921.

[5] Felix Hausdorff. Summationsmethoden und Momentfolgen. II. *Mathematische Zeitschrift*, 9:280–299, 1921.

[6] Pierluigi Novi Inverardi, Giorgio Pontuale, Alberto Petri, and Aldo Tagliani. Hausdorff moment problem via fractional moments. *Applied Mathematics and Computation*, 144(1):61–74, 2003.

[7] G. Talenti. Recovering a function from a finite number of moments. *Inverse Problems*, 3:501–517, 1987.

[8] H. Yamada. Real-time computation and recursive functions not real-time computable. *IRE Transactions on Electronic Computers*, 11:753–760, 1962.