The Mathematics of Mathematical Handwriting Recognition

Stephen M. Watt
University of Western Ontario

15 Sept 2010, TRICS, U. Western Ontario, Canada

The Pen as an Input Device

- Pen input for electronic devices is becoming important as an input modality.

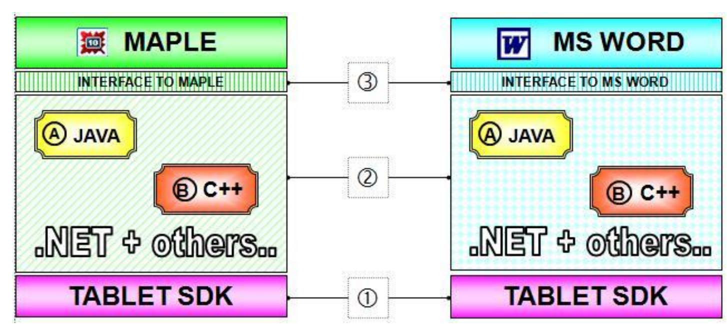
- Pens can be used where keyboards can't, on very small or very large devices, in wet or dirty environments, by people with repetitive stress injuries.
- They also allow much easier 2-dimensional input, e.g. for drawings, music or mathematics.

$$e^x = \int e^x dx = \sum_{i=0}^{\infty} \frac{x^i}{i!}$$

Our Work in Pen-Based Computing

Long-term ongoing projects

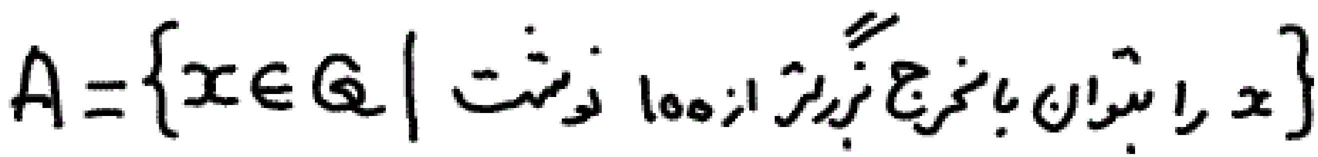
- Mathematical handwriting recognition
  - for computer algebra
  - for document processors
  - Geometric and statistical methods
- Real time ink processing
- Multi-modal ink – *Skype* add on
- Portability of ink data – *InkML*
- Portability of ink software



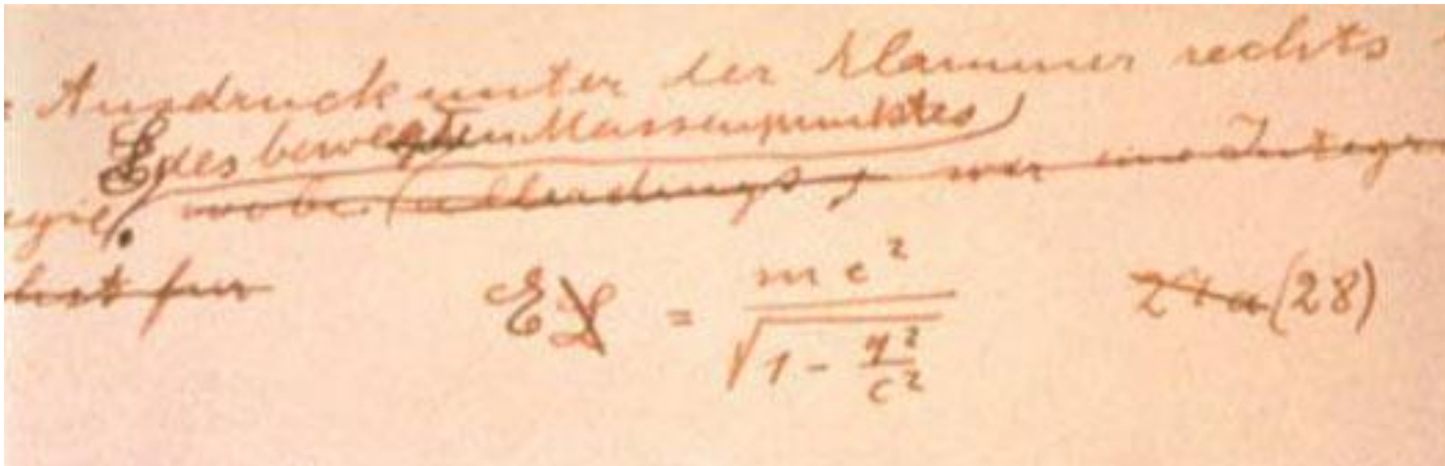
Our Work in Pen-Based Computing

Newer projects

- Personal handwritten fonts
- Handwriting neatening
- Calligraphic rendering with InkML
- Ink compression
- Multi-language ink handling



Pen-Based Math?



- Input for CAS and document processing.
- 2D editing.
- Computer-mediated collaboration.
- Killer app for pen.

Pen-Based Math!

- Does not require learning another language:

$$\sum_{i=0}^{r} g_{r-i} x^{i}$$

```
\sum_{i=0}^r g_{r-i} X^i
```

```
sum(g[r-i]*X^i, i = 0..r);
```

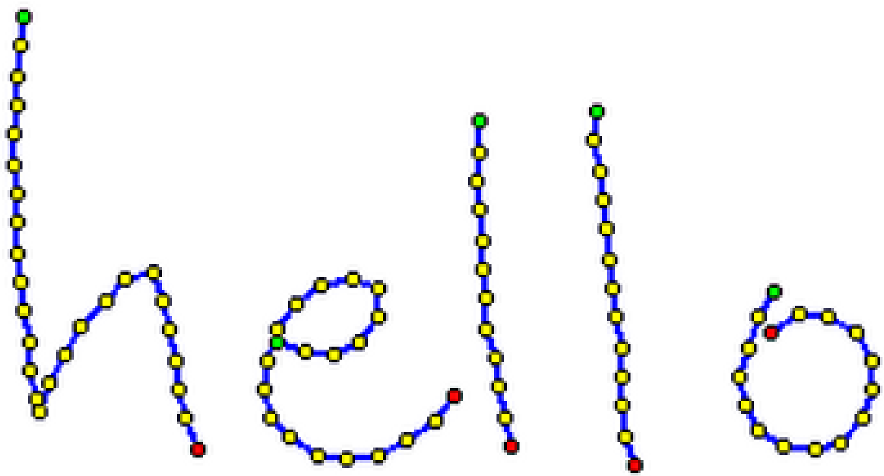
Pen-Based Math!

- Different than natural language recognition:
  - 2-D layout is a combination of writing and drawing.
  - Many similar few-stroke characters.
  - Many alphabets, used idiosyncratically.
  - Lots of symbols, each person uses only small subset.
  - No fixed dictionary.

Character Recognition

- Will concentrate on character recognition
- Several projects ignoring this problem
- Three statisticians go hunting

Digital Ink Formats



- Collected by surface digitizer or camera
- Sequence of (x, y) points in time
  sampled at some known frequency
  + possibly other info (angles, pressure, etc)
- Grouping into traces, letters, words + labelling

W3C Working Draft

Ink Markup Language (InkML)

W3C Working Draft 27 May 2010

This version:
http://www.w3.org/TR/2010/WD-InkML-20100527/
Latest version:
http://www.w3.org/TR/InkML
Previous version:
http://www.w3.org/TR/2006/WD-InkML-20061023
Editors:
Stephen M. Watt, University of Western Ontario
Tom Underhill, Microsoft
Authors:
Yi-Min Chee (until 2006 while at IBM)
Katrin Franke (until 2004 while at Fraunhofer Gesellschaft)
Max Froumentin (until 2006 while at W3C)
Sriganesh Madhvanath (until 2009 while at HP)
Jose-Antonio Magaña (until 2006 while at HP)
Gregory Russell (until 2005 while at IBM)
Muthuselvam Selvaraj (until 2009 while at HP)
Giovanni Seni (until 2003 while at Motorola)
Christopher Tremblay (until 2003 while at Corel)
Larry Yaeger (until 2004 while at Apple)

A non-normative version of this document showing changes made since the previous draft is also available.

Copyright ©2010 W3C® (MIT, ERCIM, Keio), All Rights Reserved. W3C liability, trademark and document use rules apply.

InkML Concepts

- Traces, trace groups
- Device information: sampling rate, resolution, etc.
- Pre-defined and application defined channels
- Trace formats, coordinate transformations
- Annotation text and XML

Usual Character Reco. Methods

- Smooth and re-sample data *THEN*
- Match against *N* models by sequence alignment
  OR
- Identify "features", such as
  - Coordinate values of sample points, Number of loops, cusps, Writing direction at selected points, *etc*

  Use a classification method, such as
  - Nearest neighbour, Subspace projection, Cluster analysis, Support Vector Machine

  THEN
- Rank choices by consulting dictionary

Difficulties

- Having many similar characters (e.g. for math) means comparison against all possible symbol models is slow.
- Determining features from points
  - Requires many *ad hoc* parameters.
  - Replaces measured points with interpolations
  - It is not clear how many points to keep,
    and most methods depend on number of points
  - Device dependent
- What to do since there is no dictionary?
- New ideas are needed!

Two Thoughts

- For HWR do we need all the trace data?
  - Do we need all the points?
  - Do we need full accuracy for all the points?

- What is classification?
  - H (English aitch, Greek eta, Russian en)
  - O (zero, oh, degree, …)
  - P, C, S (R, S, T)

Fundamental Thm of HWR

∀ A, *if a sample looks like an* A, *then it can be an* A.

Fundamental Thm of HWR

$\forall$ A, *if a sample looks like an* A, *then it can be an* A.

Corollaries:

- Classification gives a set of valid possibilities.
- Must be able to classify perturbed inputs.
- Can use approximation to represent traces more conveniently.

Orthogonal Series Representation

- **Main idea**:
  Represent coordinate curves as truncated orthogonal series.

- **Advantages**:
  - *Compact* – few coefficients needed
  - *Geometric*
    – the truncation order is a property of the character set
    – gives a natural metric on the space of characters
  - *Algebraic*
    – properties of curves can be computed algebraically
    (instead of numerically using heuristic parameters)
  - *Device independent*
    – resolution of the device is not important

Inner Product and Basis Functions

- Choose a functional inner product, e.g.

$$\langle f, g \rangle = \int_a^b f(t) g(t) w(t) dt$$

- This determines an orthonormal basis in the subspace of polynomials of degree *d*.
- Determine ϕ_i using GS on $\{1, t, t^2, t^3, \ldots\}$.
- Can then approximate functions in subspaces

$$A(t) \approx \sum_{i=0}^{d} \alpha_i \phi_i(t) \qquad \alpha_i = \langle A(t), \phi_i(t) \rangle$$

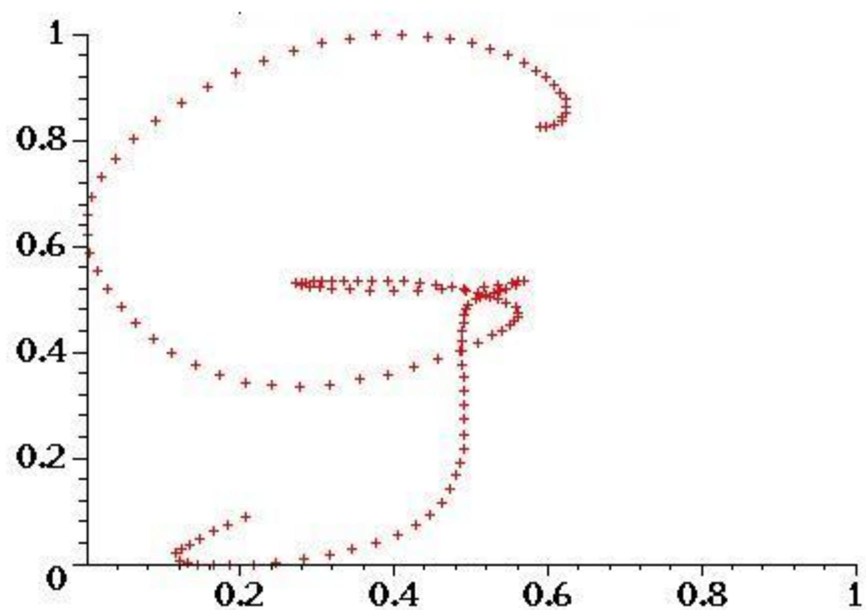
First Look: Chebyshev Series

- Initially used Chebyshev series
  [Char+SMW ICDAR 2007].

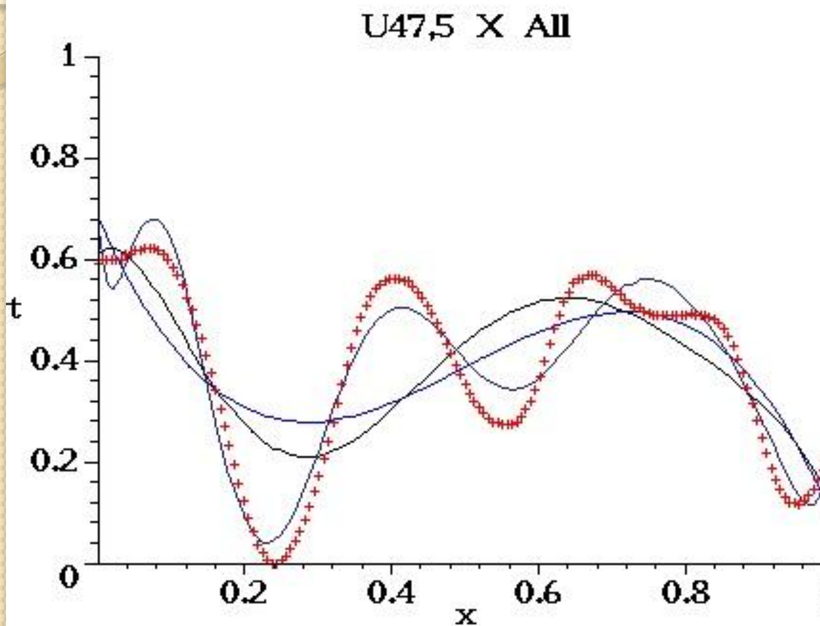
- Found could approximate closely
  (small RMS error) with series of order 10.

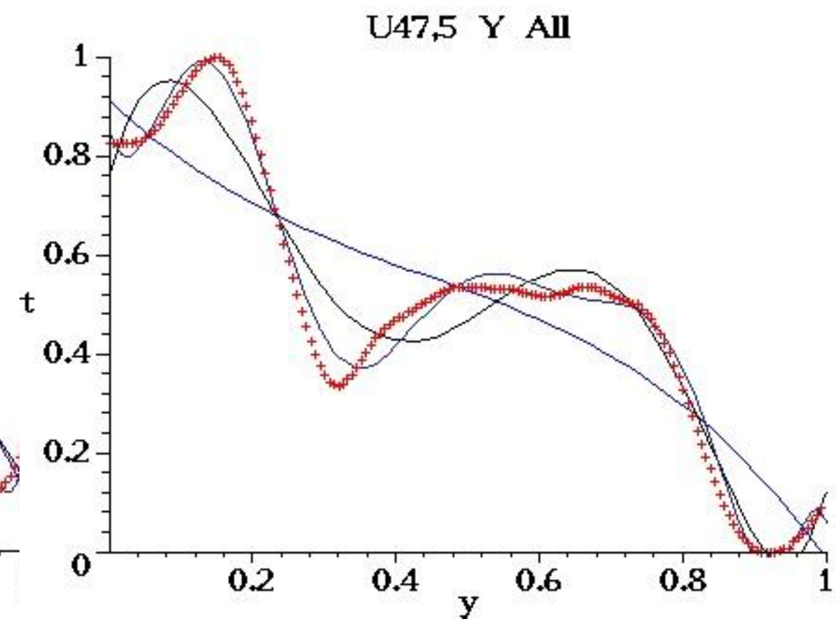
- Like symbols formed clusters.

Raw Data for Symbol G

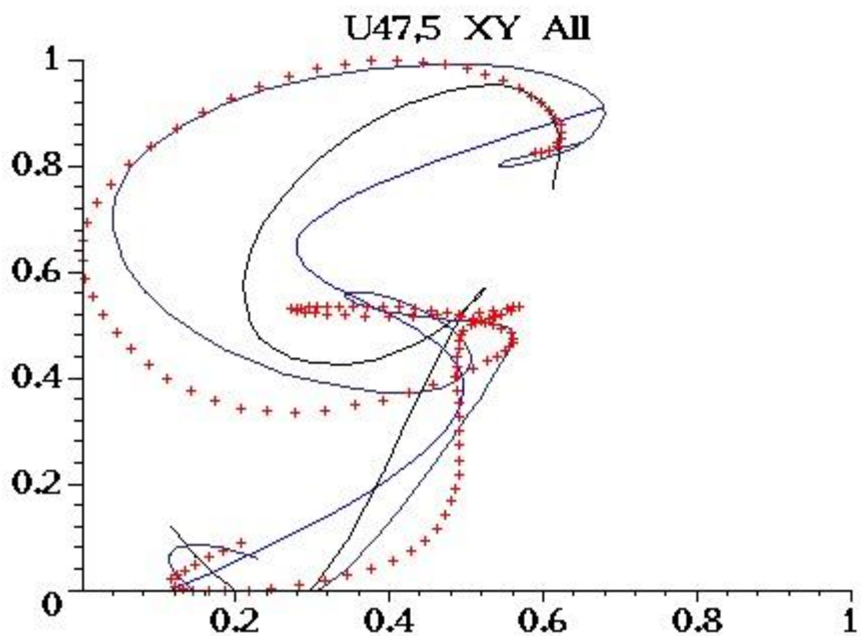


Coordinate fn approximations

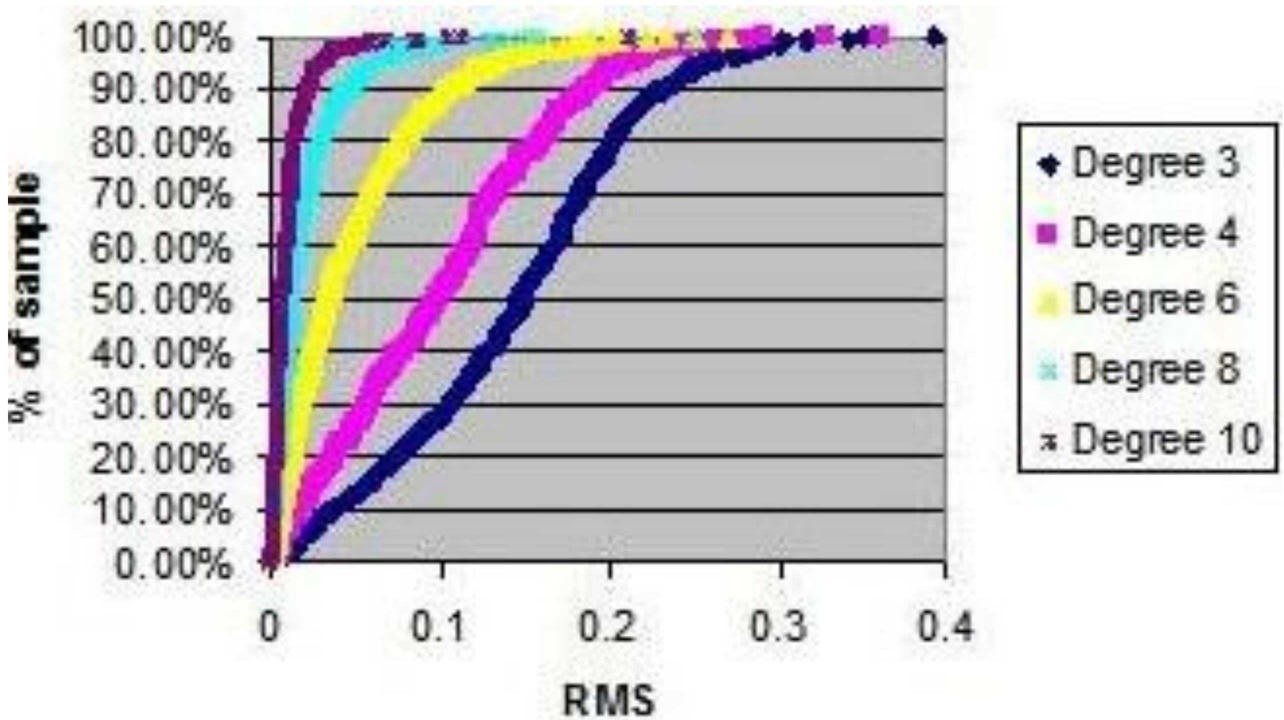




Chebyshev Approx to Character



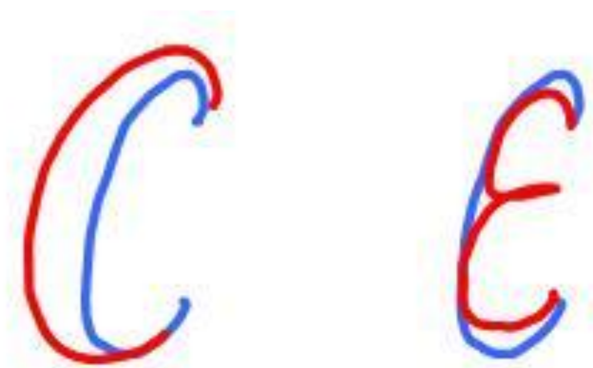
RMS Error


100.00%
90.00%
80.00%
70.00%
60.00%
50.00%
40.00%
30.00%
20.00%
10.00%
0.00%
% of sample
0
0.1
0.2
0.3
0.4
RMS
Degree 3
Degree 4
Degree 6
Degree 8
Degree 10

Problems

- Want fast response –
  how to work while trace is being captured.

- Low RMS does not mean similar shape.



Pb 1. On-Line Ink

- The main problem:
  In handwriting recognition, the human and the computer take turns thinking and sitting idle.

- We ask:
  Can the computer do useful work while the user is writing and thereby get the answer faster after the user stops writing?

- We show:
  The answer is "Yes"!

An On-Line Complexity Model

- Input is a sequence of n values received at a uniform rate.

- Characterize an algorithm by
  - $T_\Delta(n)$ no. operations as n-th input is seen
  - $T_F(n)$ no. operations after last input is seen

- Write on-line time complexity as

$$\mathrm{OL}_n[T_{\Delta(n)}, T_F(n)]$$

- E.g., linear insertion sort requires time

$$\mathrm{OL}_n[O(n), 0]$$

On-Line Series Coefficients (main idea)

- If we choose the right basis functions, then the series coefficients can be computed on line.
  [Golubitsky+SMW CASCON 2008, ICFHR 2008]
- The series coefficients are linear combinations of the moments, which can be computed by numerical integration as the points are received.

$$\langle P_n, x\rangle = \sum_{k=0}^{n} p_{n,k} \int_0^L \lambda^k x(\lambda) d\lambda$$

- This is the Hausdorff moment problem (1921), shown to be unstable by Talenti (1987).
- It is just fine, however, for the dimensions we need.

On-Line Series Coefficients (more details)

- Use Legendre polynomials P_i as basis on the interval $[-1,1]$, with weight function 1.

- Collect numerical values for $f(\lambda)$ on $[0, L]$.
  λ = arc length.
  L is not known until the pen is lifted.

- As the numerical values are collected, compute the moments $\int \lambda^i f(\lambda) d\lambda$.
- After last point, compute series coeffs for f with domain and range scaled to $[-1,1]$.
  These will be linear combinations of the moments.

Complexity

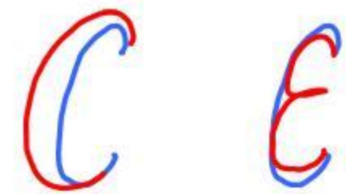
- The on-line time complexity to compute coefficients for a Legendre series truncated to degree d is then

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

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

- The time at pen up is *constant* with respect to the number of points.

Pb 2. Shape vs Variation



- The corners are not in the right places.

- Work in a jet space to force coords & derivatives close.

- Use a Legendre-Sobolev inner product

$$\langle f, g\rangle = \int_a^b f(t)g(t)dt + \mu_1 \int_a^b f'(t)g'(t)dt + \mu_2 \int_a^b f''(t)g''(t)dt + \cdots$$

- 1^{st} jet space $\Rightarrow$ set $\mu_i = 0$ for $i > 1$.
  Choose μ_1 experimentally to maximize reco rate.
  Can be also done on-line.
  [Golubitsky + SMW 2008, 2009]

Life in an Inner Product Space

- With the Legendre-Sobolev inner product we have
  - Low dimensional rep for curves (10 + 10 + 1)
  - Compact rep of samples ~ 160 bits [G+W 2009]
  - A useful notion of distance between curves
    that is very fast to compute
  - >99% linear separability and convexity of classes

- Training data of 50,000 math char samples.
  Use 75% for training 25% test. 10X cross validation.

Convexity of Classes

- Can separate N classes with $N(N-1)$ SVM planes.
- Each class is then (mostly) within its own convex polyhedral cell.
- Can classify either by
  - SVM majority voting + run-off elections (96%)
  - Distance to convex hull of k nearest neighbours (97.5%).
  - On-line computation.

Comparison of Candidate to Models

- Some classification methods compute the distance between the input curve and models.

  E.g. Elastic matching takes time up to quadratic in the number of sample points and linear in the number of models.

- Many tricks and heuristics to improve on this.

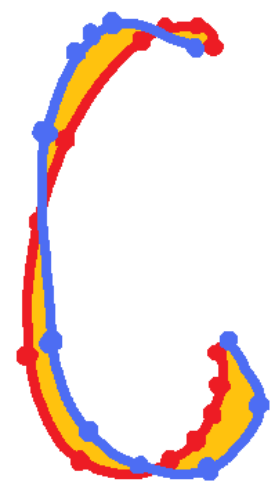
  E.g. Limit amount of dynamic time warping,
  pre-classify based on features, ...

Distance Between Curves

- Approximate the variation between curves by some fn of distances between points.
- May be coordinate curves or curves in a jet space.

- Sequence alignment
- Interpolation ("resampling")

- Why not just calculate the area?
- This is very fast in ortho series representation.



Distance Between Curves

$$\bar{x}(t) = x(t) + \xi(t) \qquad \xi(t) = \sum_{i=0}^{\infty} \xi_i \phi_i(t), \qquad \phi_i \text{ orthonormal on } [a,b] \text{ with } w(t) = 1.$$

$$\bar{y}(t) = y(t) + \eta(t) \qquad \eta(t) = \sum_{i=0}^{\infty} \eta_i \phi_i(t)$$

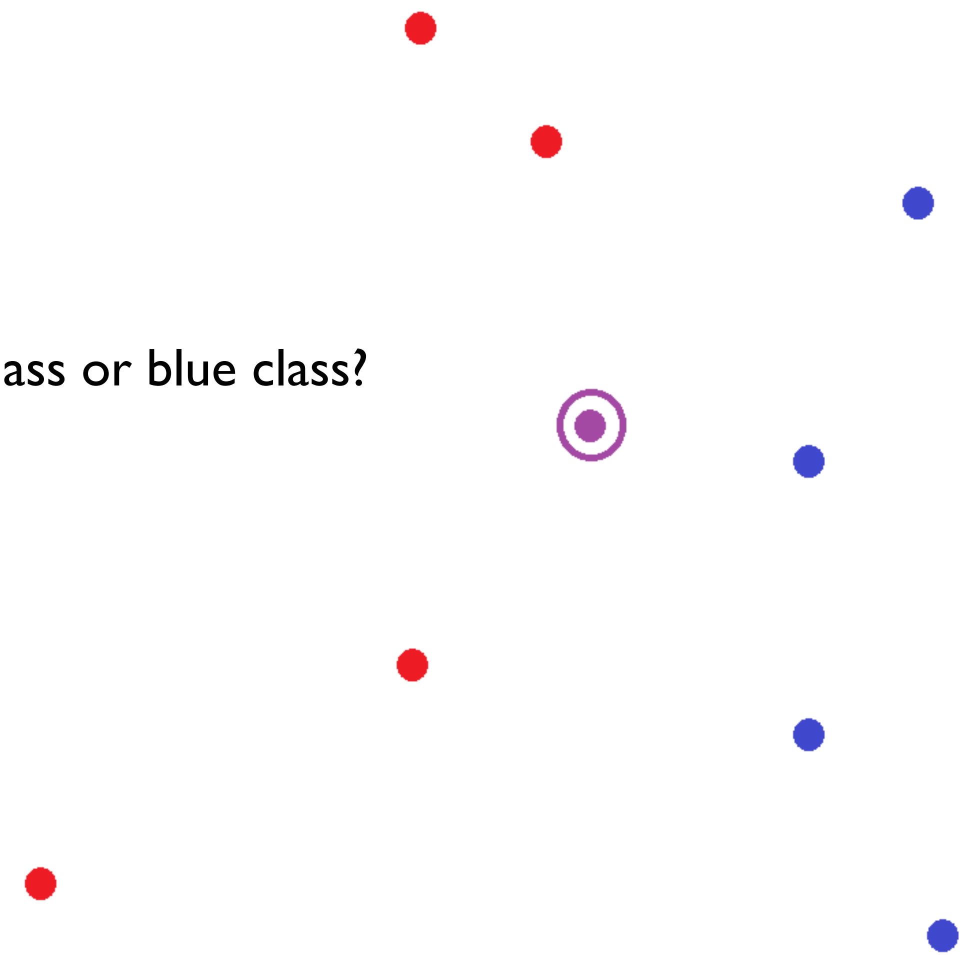
$$\begin{aligned}
&\rho^2(C, \bar{C}) \\
&\quad = \int_a^b \left[\left(x(t) - \bar{x}(t)\right)^2 + \left(y(t) - \bar{y}(t)\right)^2\right] dt \\
&\quad = \int_a^b [\xi(t)^2 + \eta(t)^2] dt \\
&\quad \approx \int_a^b \left[\sum_{i=0}^{d} \xi_i^2 \phi_i^2(t) + \text{cross terms} + \sum_{i=0}^{d} \eta_i^2 \phi_i^2(t) + \text{cross terms}\right] dt \\
&\quad = \sum_{i=0}^{d} \xi_i^2 + \sum_{i=0}^{d} \eta_i^2
\end{aligned}$$

Comparison of Candidate to Models

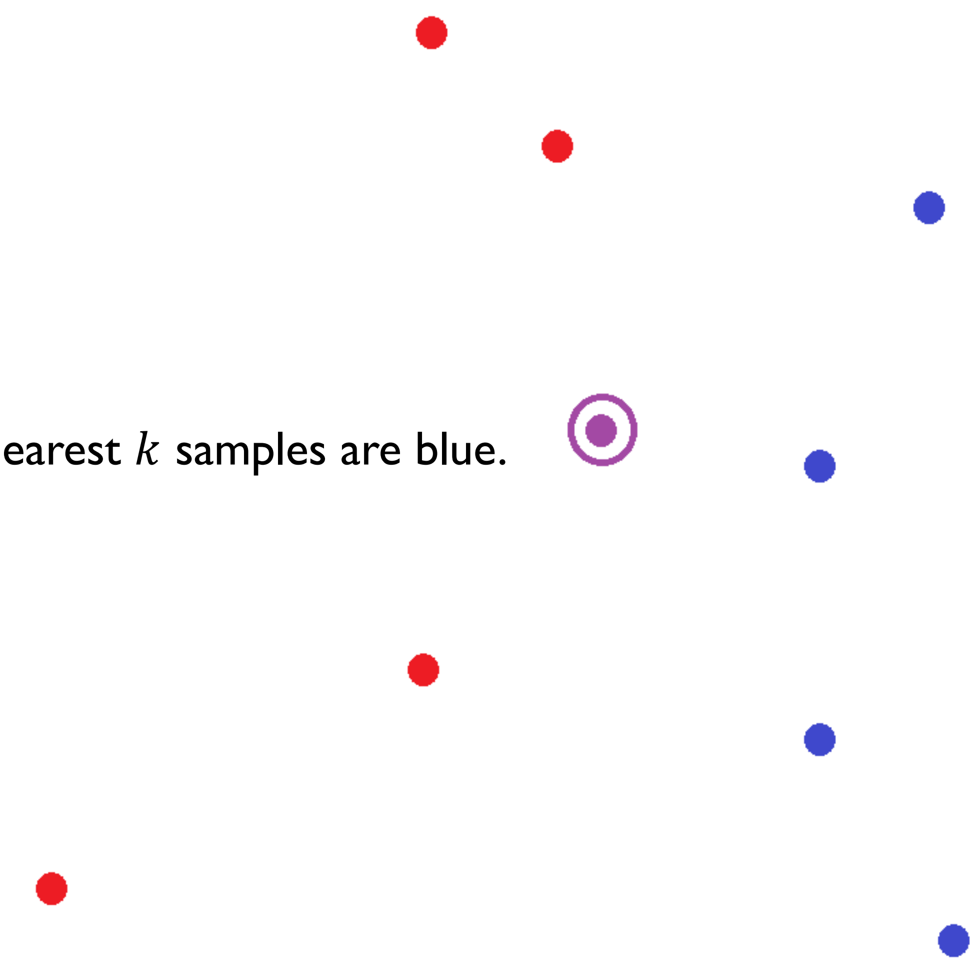
- Use Euclidean distance in the coefficient space.
- *Just as accurate* as elastic matching.
- *Much less expensive.*
- Linear in d, the degree of the approximation.
  $< 3\,d$ machine instructions (30ns) *vs* several thousand!
- Can trace through SVM-induced cells incrementally.
- Normed space for characters gives other advantages.

Choosing between Alternatives

Red class or blue class?



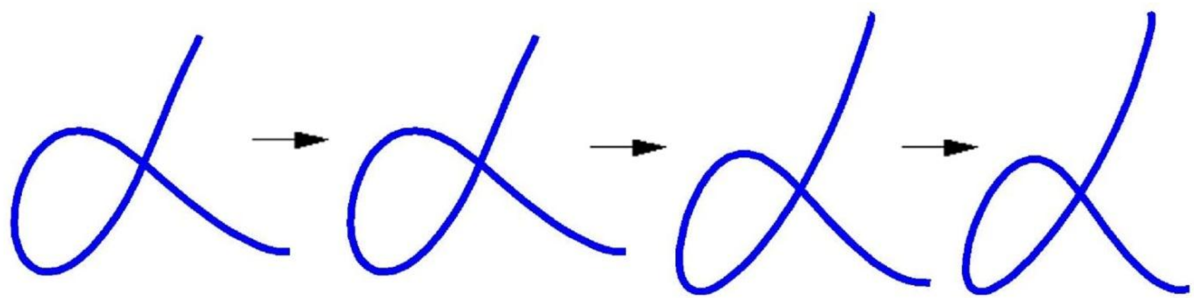
Choosing between Alternatives



The nearest k samples are blue.

Geometry

- Linear separability ⇒
  Linear homotopies within a class (Fund Thm of HWR)

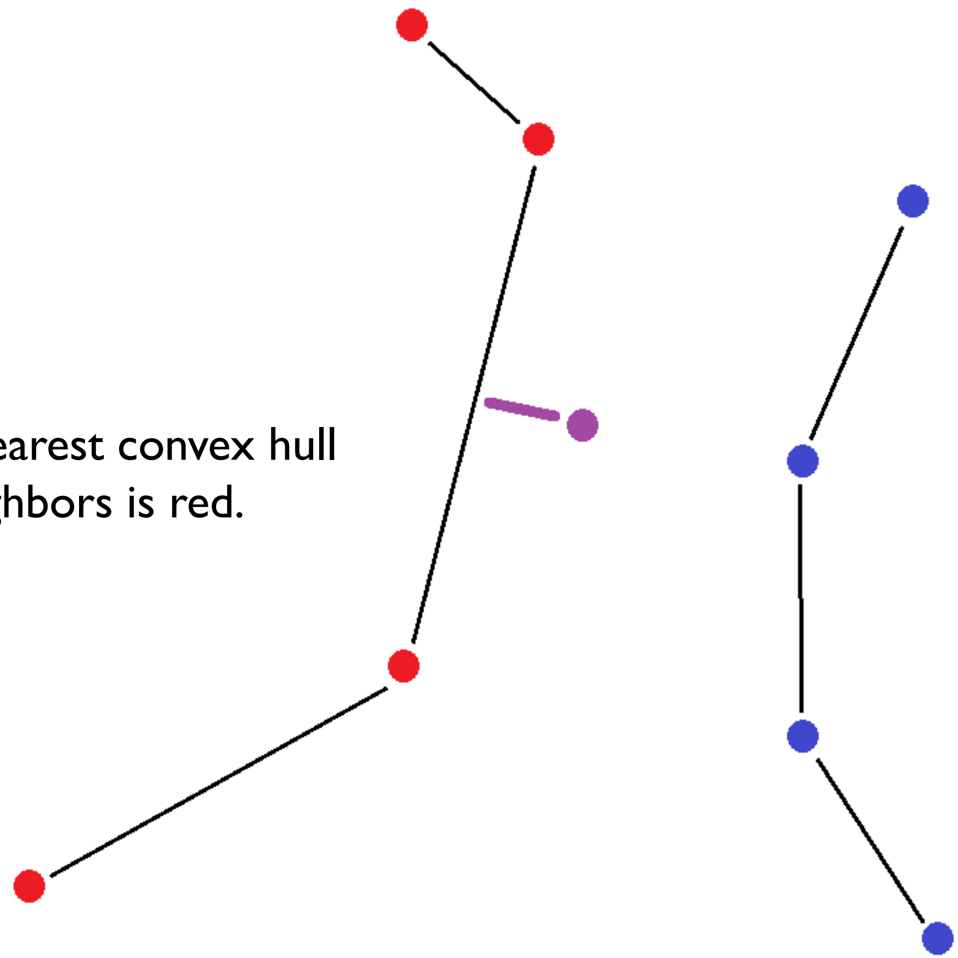


$$C = (1 - t)\,A + t\,B$$

- Can compute distance of a sample to this line
- Distance to convex hull of a set of models gives best recognition [Golubitsky+SMW 2009,2010]

Choosing between Alternatives

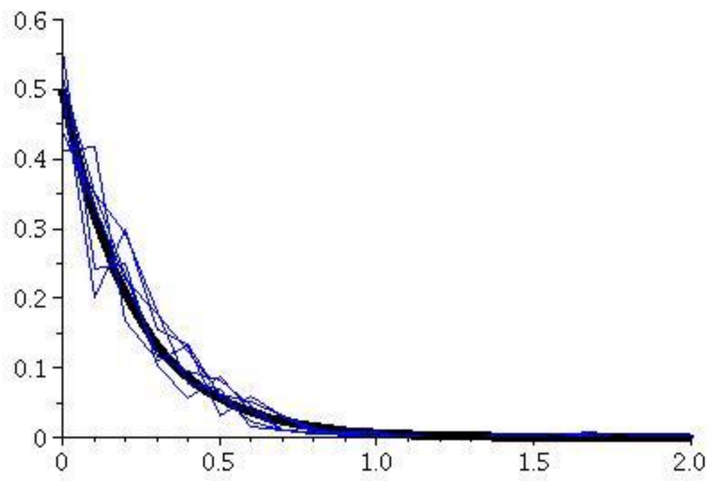
The nearest convex hull of neighbors is red.



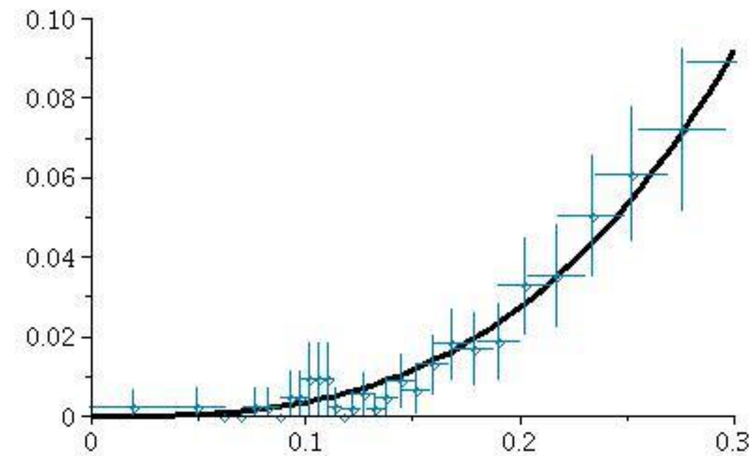
Recognition Summary

- Database of samples ⇒ set of LS points
- Character to recognize ⇒ Integrate moments as being written
  - Lin. trans. to obtain one point in LS space
  - Classify by distance to convex hull of k-NN.

Error Rates as Fn of Distance



SVM

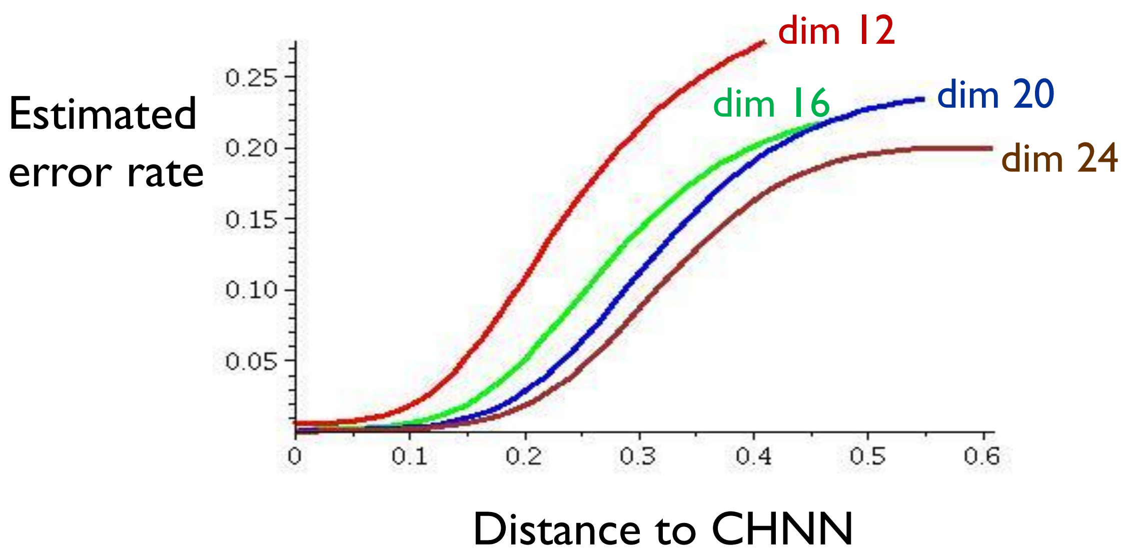


Convex Hull

- Error rate as fn of distance gives confidence measure for classifiers [MKM – Golubitsky + SMW 2009]

Cumulative Frequency Plots

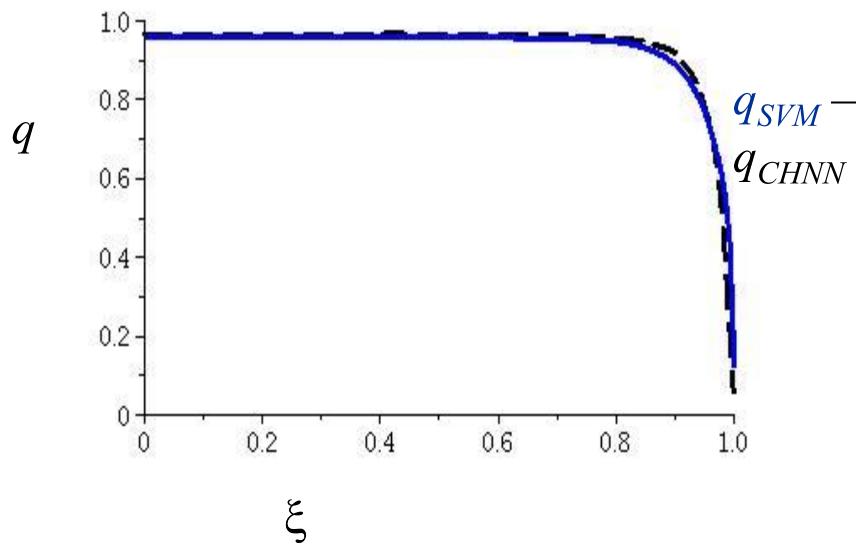
- $N(d) = \#$ samples with CHNN distance $< d$
- $e(d) = \#$ misclassified samples with CHNN distance $< d$
- Fit $f_{\alpha,\beta,\gamma,\delta}(d) = \left(\alpha d^{\beta} + \gamma\right)^{-1} + \delta$ to $N(d)$ and $e(d)$
- Obtain error rate as $e'(d)/N'(d)$

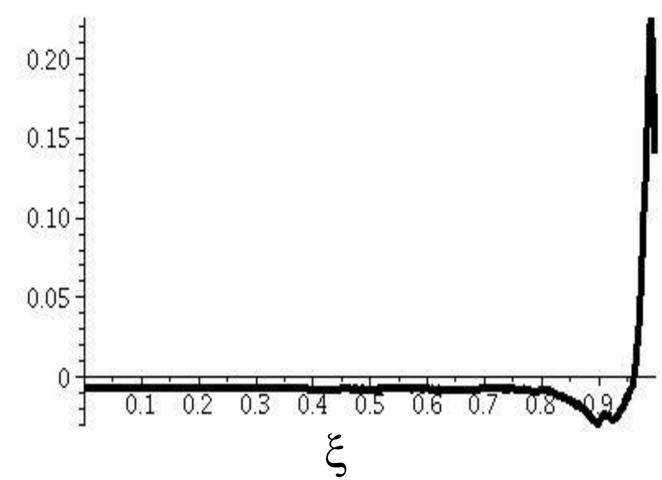


Quality of Confidence Measures

- X^+ set of correctly classified samples
- X^- set of misclassified samples
- $f(x)$ confidence values

$$q(\xi) = \frac{\#\{x \in X^+ \mid f(x) > \xi\} + \#\{x \in X^- \mid f(x) < \xi\}}{\#(X^+ \cup X^-)}$$





Ambiguities

$$\sum_{i} i^{2}$$

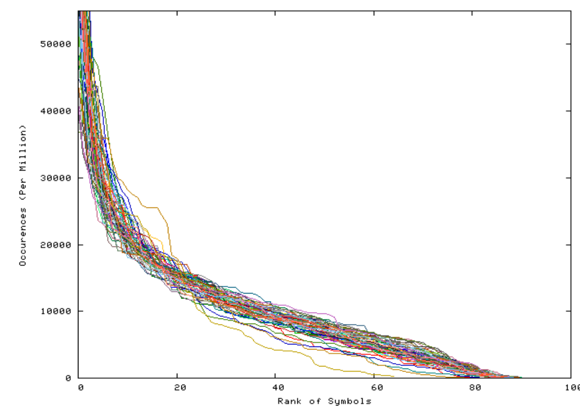
Ambiguities

Ambiguities

$$\dot{z} + z = \sin \omega t$$

Combining with Frequency Info

- Empirical confidence on classifiers allows geometric recognition of isolated symbols to be combined with statistical methods.
- Domain-specific *n*-gram information:
  - Research mathematics – 20,000 articles from arXiv [MKM -- So+SMW 2005]
  - 2nd year engineering math – most popular textbooks [DAS -- SMW 2008]
  - Inverse problem – identifying area via *n*-gram freq! [DML -- SMW 2008]



Deciding with Confidence Measure

Symbol X in an Expression E

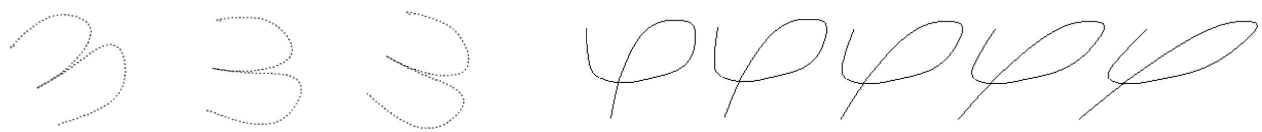
Symbol Recognizer:
$X \in \text{Class}_1$ **with Conf** ξ_1

Context-Based Predictor:
$X \in \text{Class}_2$ **with Conf** ξ_2

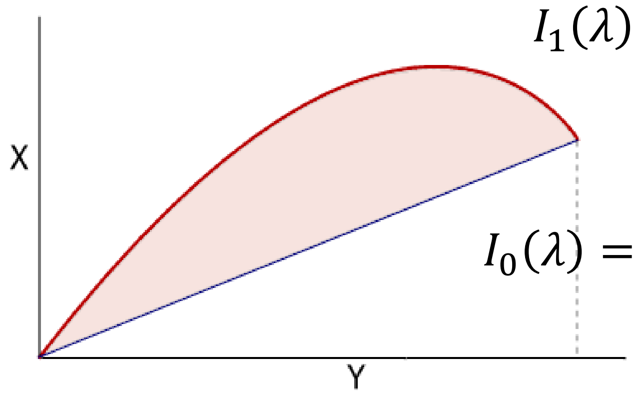
$X \in \text{Class}_i \Leftrightarrow \xi_i = \max(\xi_1, \xi_2)$

Orientation and Shear

- Reco when writing at an angle, or with slanted chars.



- Instead of taking ortho series of coordinates *x* and *y*, use ortho series of "integral invariants", a concept from algebraic geometry. [Golubitsky, Mazalov, SMW 2009 rotn, 2010 shear]



$I_1(\lambda) = area$

$I_0(\lambda) = radius$

$I_{k>1}(\lambda) = more\ complicated\ integrals$

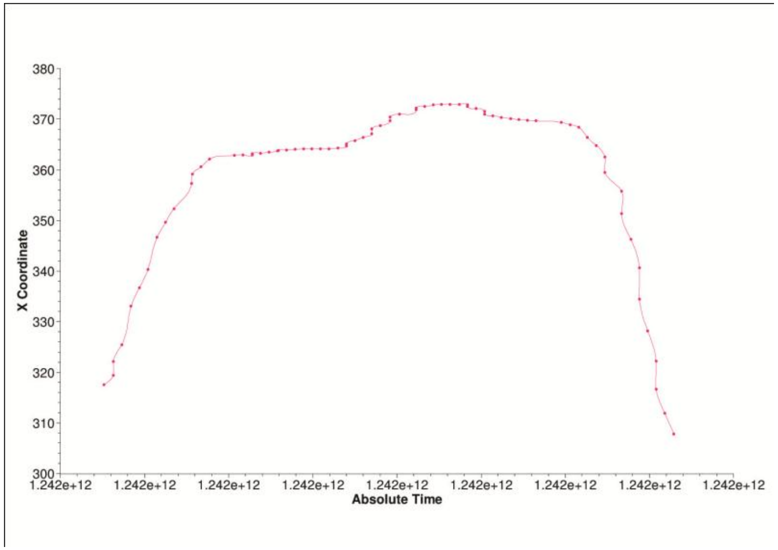
Ortho Series for Compact Ink Rep

- InkML
- Points, differences, 2nd differences
- Stream of series coefficients instead.
  [ICFHR: Mazalov+Watt 2010]

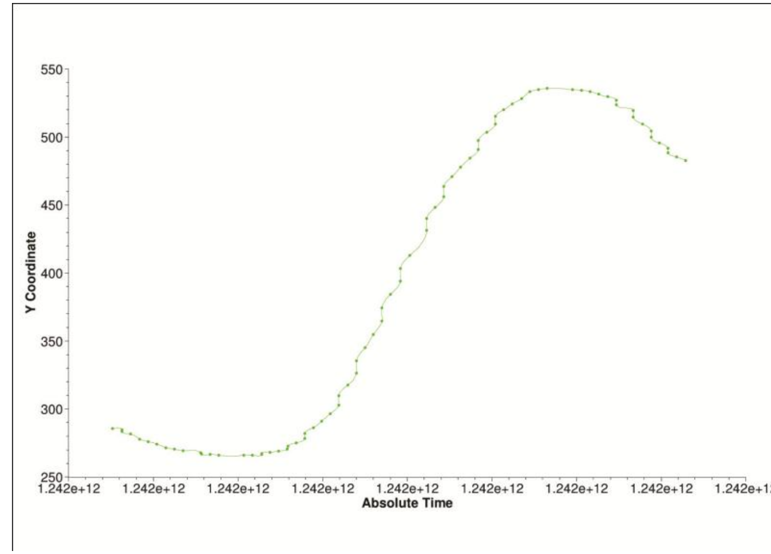
The Mathematics of Calligraphy

Single Stroke

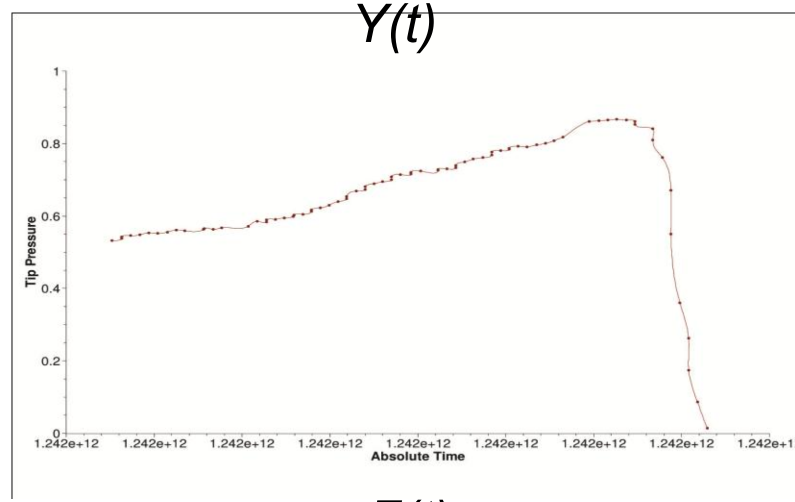
Device Data



X(t)



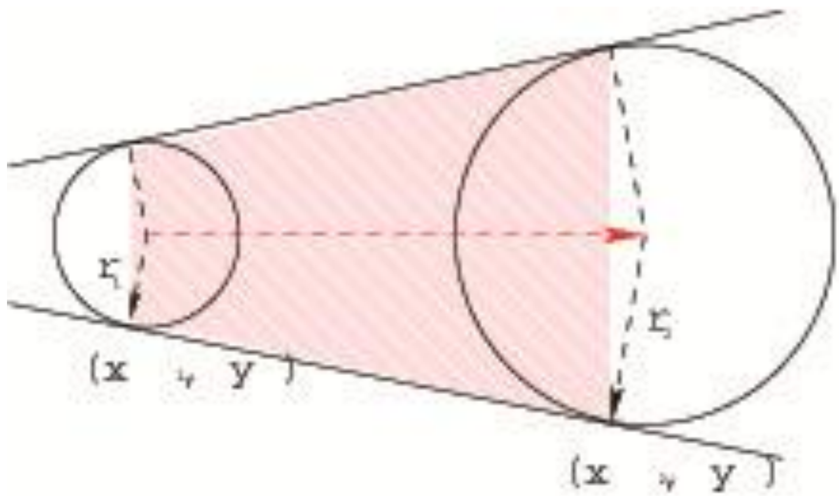
Y(t)



F(t)

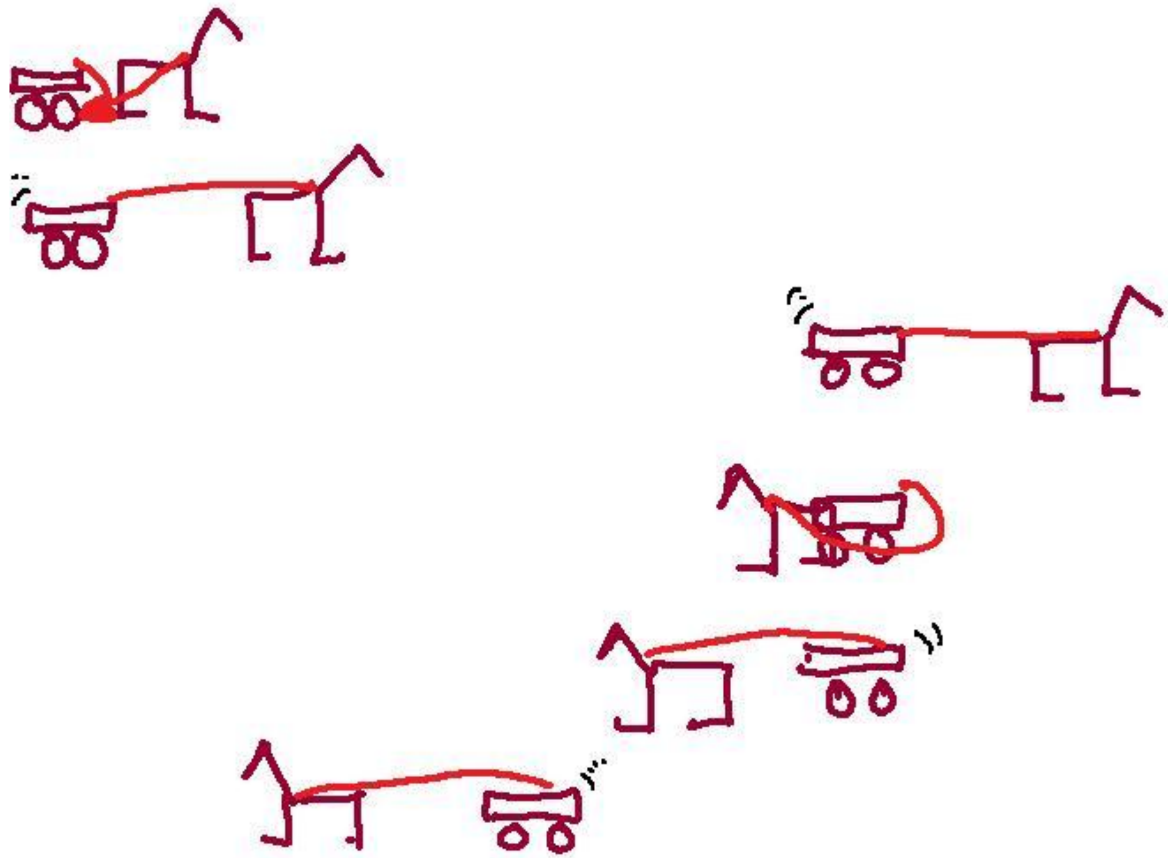
Simple Brush Models – Round Tip

- Round brush head, radius proportional to pressure



[Theng 2009]

A Donkey Pulls a Cart



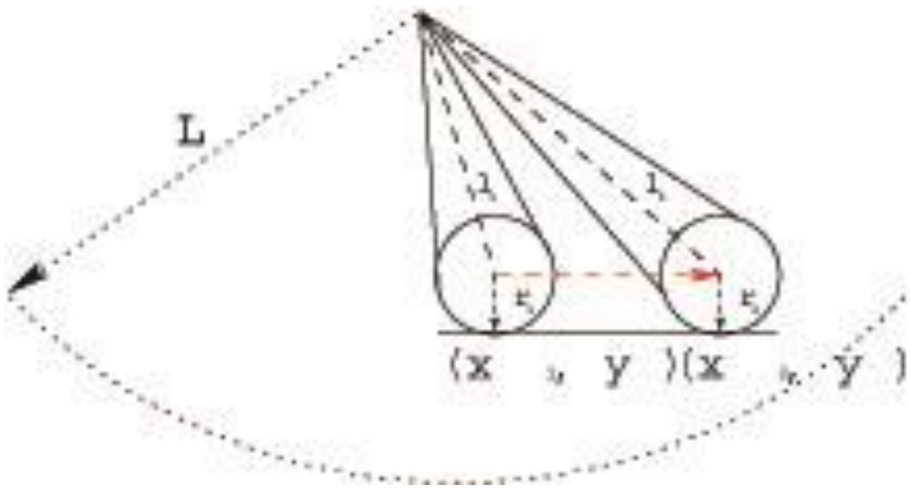
[SMW: Conf in Honor of 90th Birthday of Wu Wen Tsun]

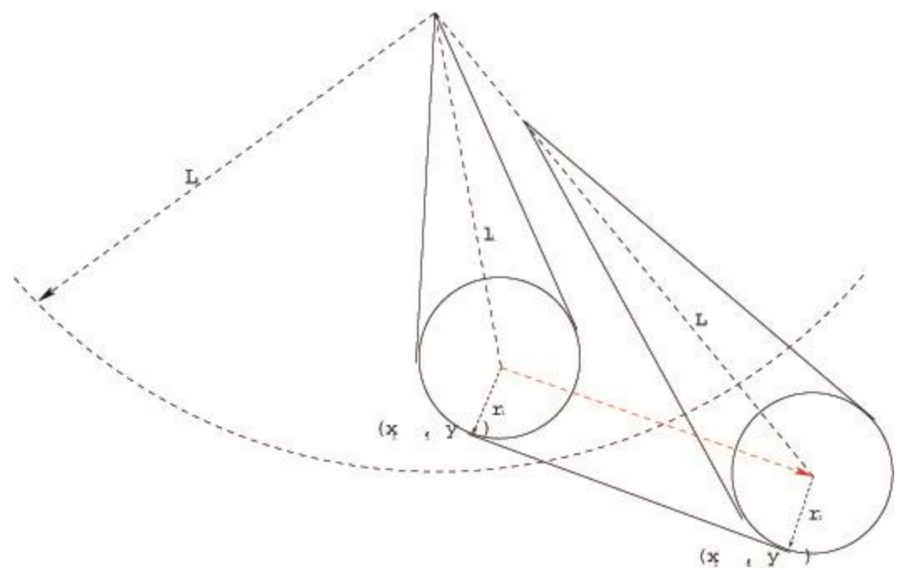
Simple Brush Models – Teardrop Tip

- Shape of brush contact with canvas is a teardrop, with the round head of the brush and trailing tail.
- Size of head is proportional to pressure.
- Tail has length between zero and some maximum, and is dragged following the head.
- Adopted into InkML.

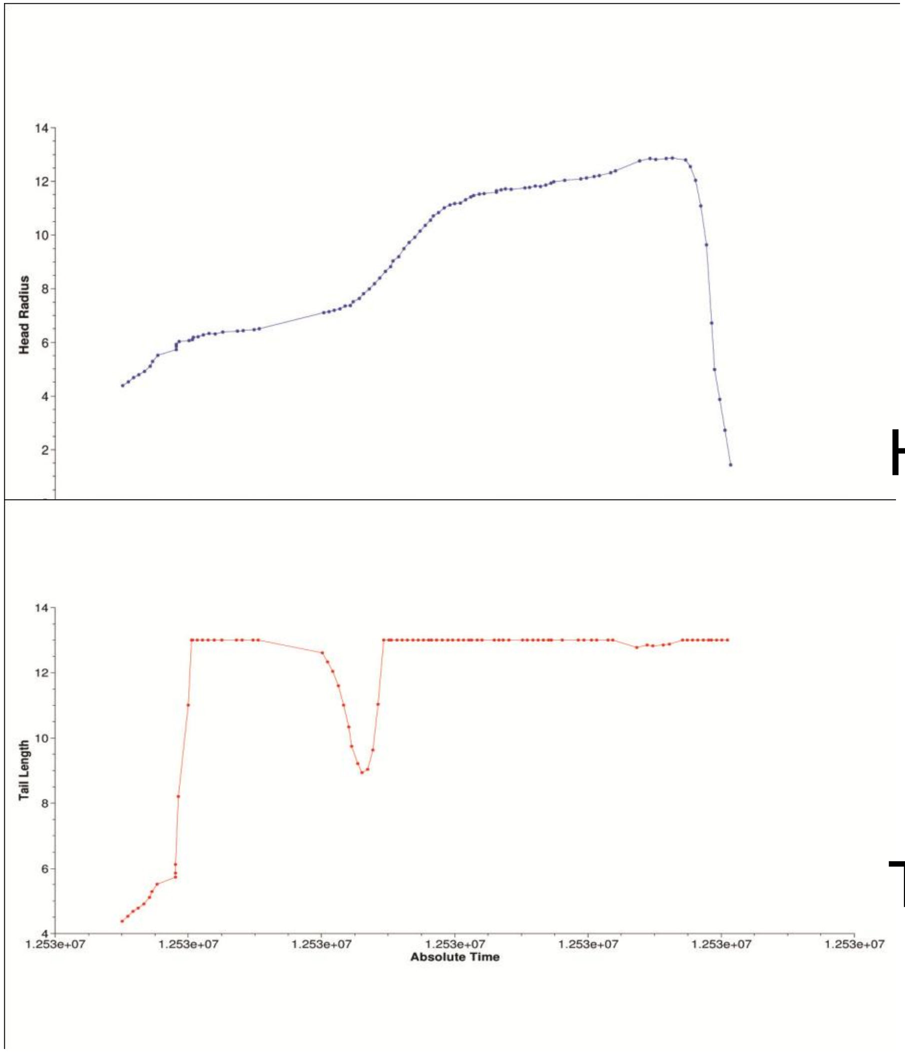
[Rui Hu 2009]

A Head Pulls a Tail





Modelled Parameters



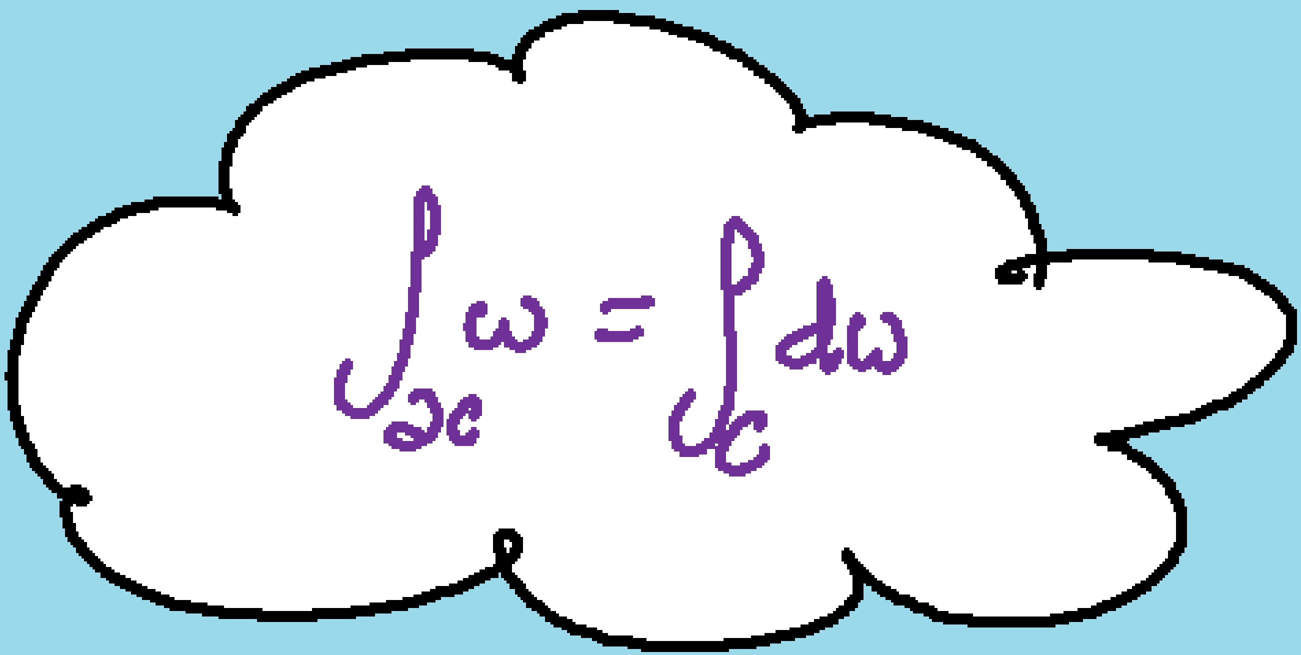
Head Radius

Tail Length

More sophisticated models

- Brush as a collection of parameterized teardrops, overlapping.
- Physical properties:
  - Brush stiffness
  - Fixed volume of brush head
- Inverse problem:
  - Given calligraphic form, determine brush position/angle/pressure path.
- Representation:
  - Ortho series for x, y, θ, R, l
  - Allows recognition and faithful rendering,
  - Use brush parameters in trace format.

Writing on Clouds



Writer-Dependency

- Users do have writing styles
- Some symbols will be written idiosyncratically (*i.e.* wrong)
- => Multi-writer point set has
  some classes that are too big (in LS space)
  some classes that are too small.
- => Allow users to add + remove points.

Where to keep the user profile?

- Today people use multiple devices.
  - Laptop, office computer, home computer, telephone, smart white board
- Cloud-based storage

Profile server

- User accounts
- Individual ink profile server
- Ink-based apps (e.g. our Skype ink chat) will retrieve profile
  and save user-accepts/rejects of reco.

The Quid Pro Quo

- *For the user:*
  Applications will get very good at recognizing individual's handwriting.

- *For us:*
  We get lots of data.

Next Directions

- More applications using the user profiles.

- Use new data:
  - Identify common writing styles
  - Identify correlations among symbol variants
  - Better writer *independent* math recognition.

Conclusions

- Ask what are we really trying to do.
- Work with ink traces objects as curves, rather than as collections of sample points.

- Admits powerful analytic tools.
- Have useful geometry on space of curves.

- Gives device/resolution independence.
- Gives faster algorithms.
- Gives useful insights.

- Gives framework for output as well.