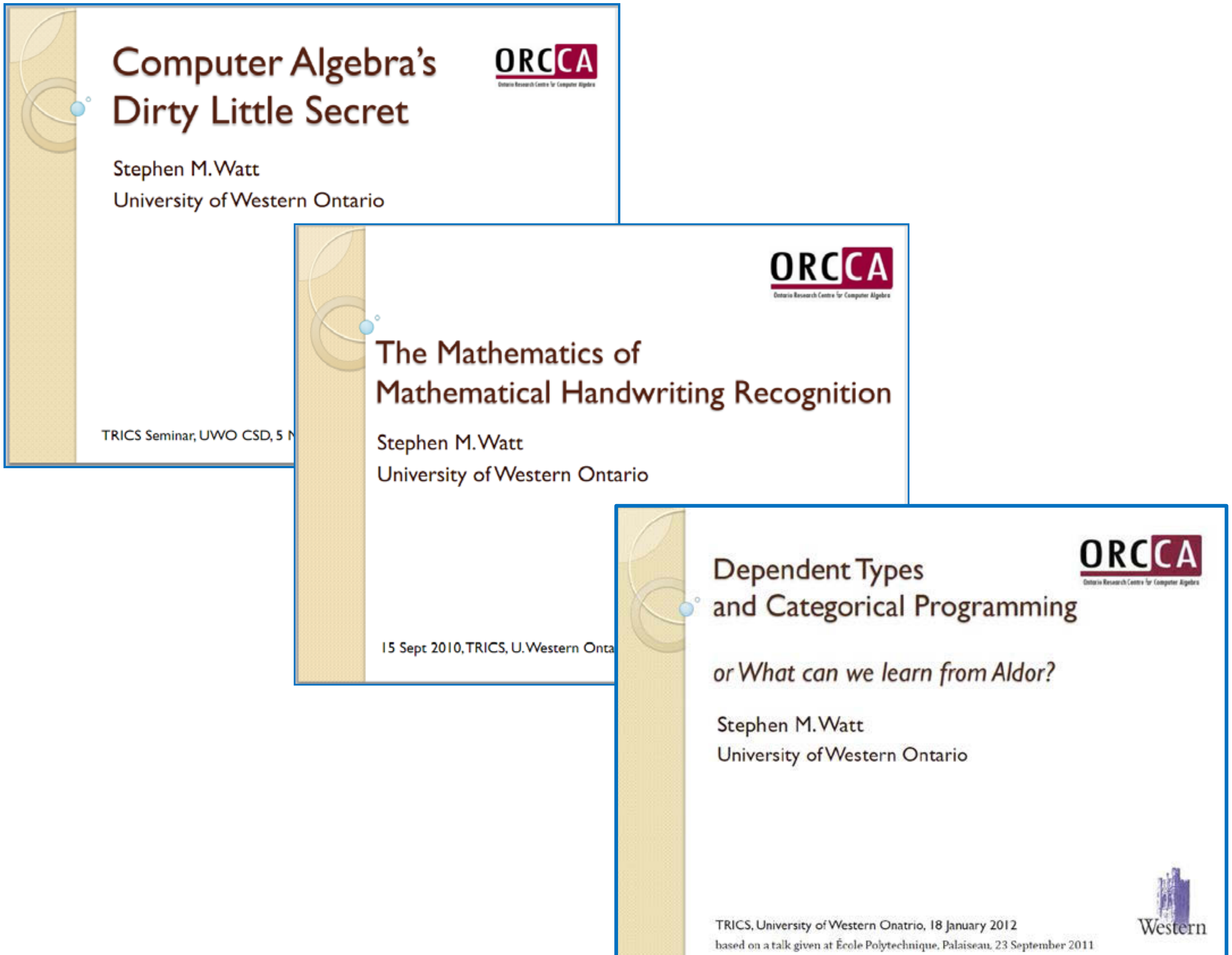
Tools for Online Technical Collaboration

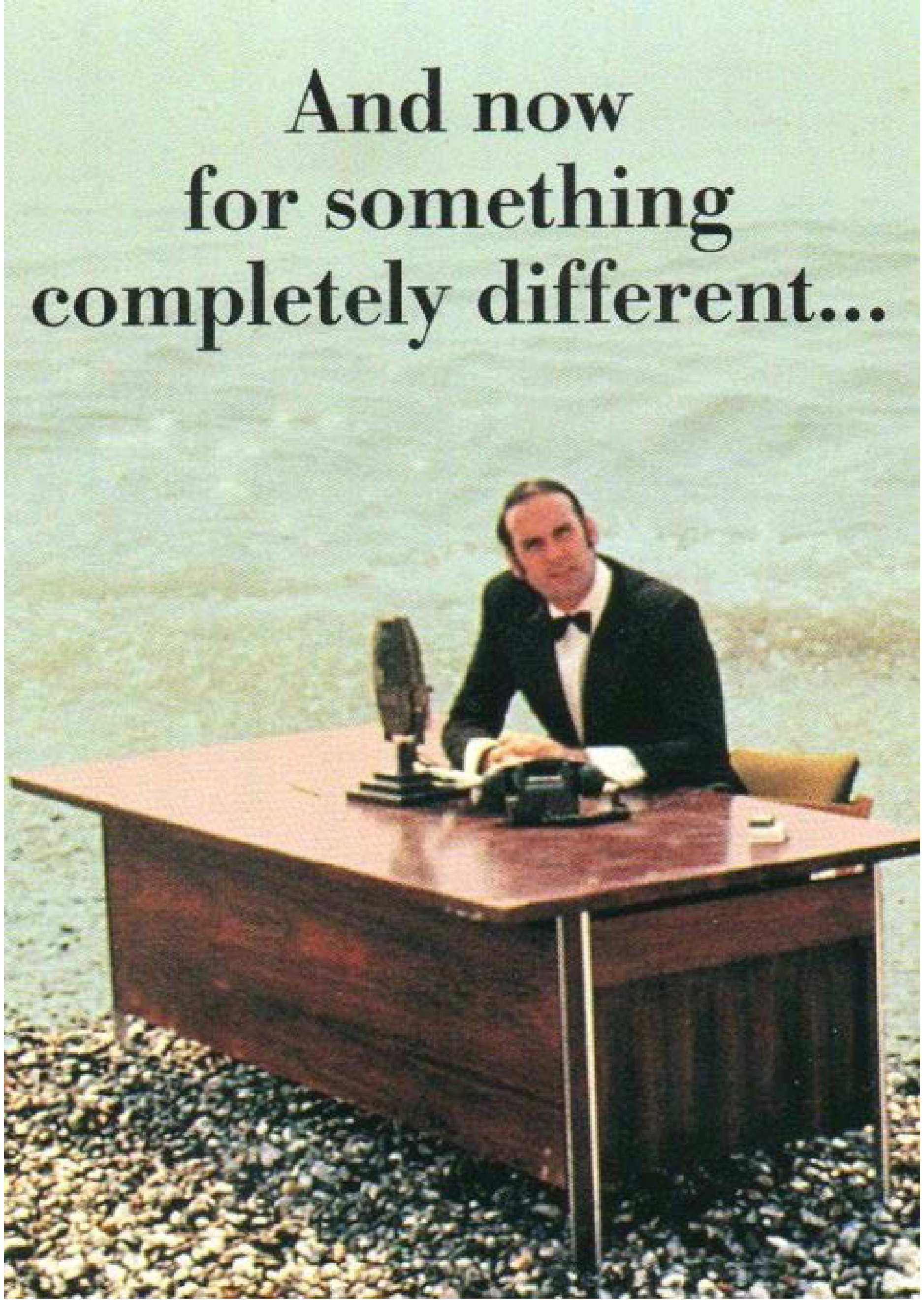
Stephen M. Watt
University of Western Ontario

TRICS, University of Western Onatrio, 10 September 2014

Preliminary version of a talk to be given at West University of Timisoara, Romania, 22 September 2014

Previous TRICS!




And now
for something
completely different...

Menu

Appetizers

Collaboration

Technical Content vs Pictures

Mains

Digital Ink

Mathematical Handwriting

Deserts

Previous Software

Present Generation

Collaboration

Collaborative Software

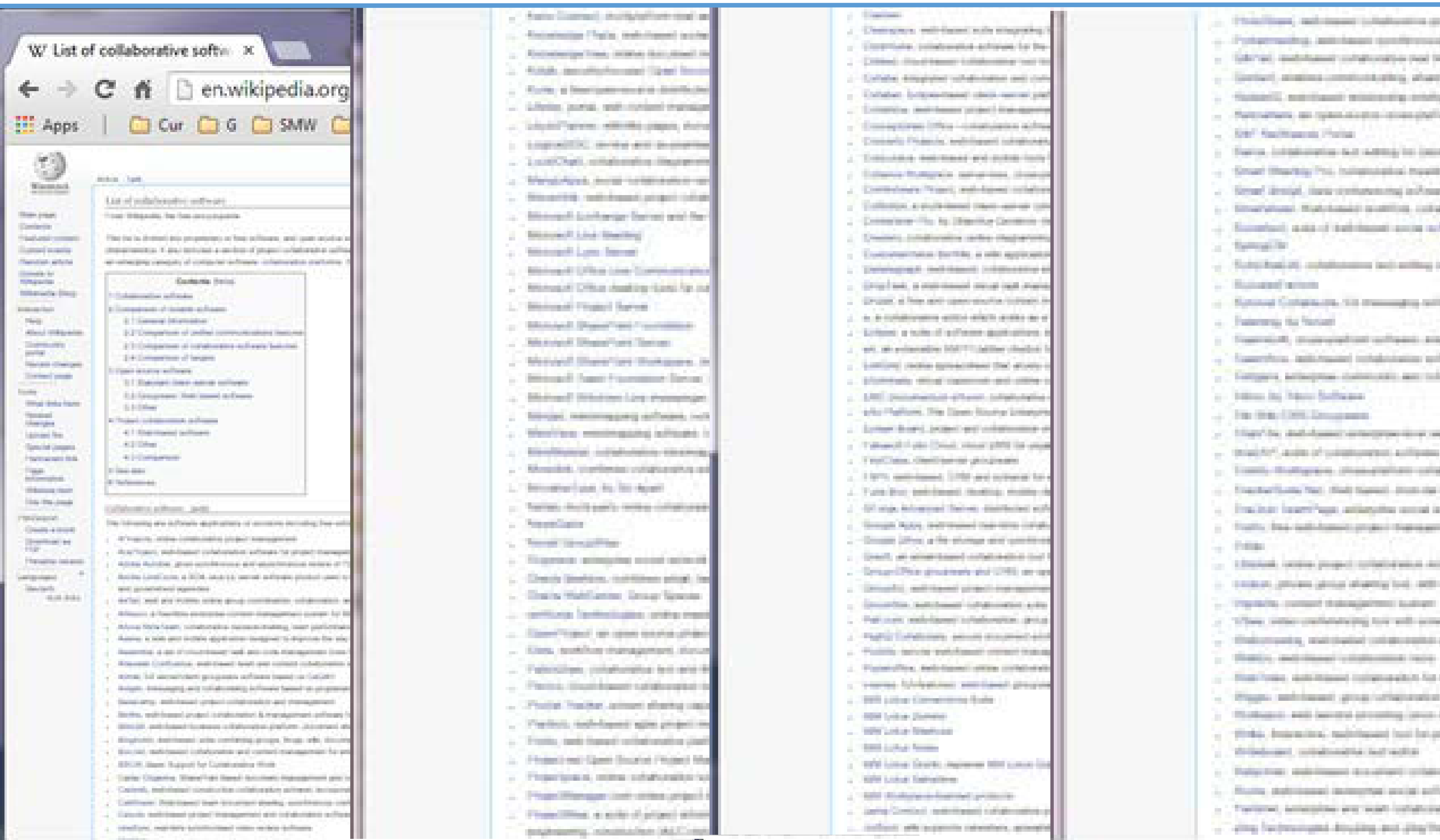
Google Docs

DESKAWAY
SIMPLIFY WORK

Dropbox

Dabbleboard
The whiteboard reinvented
Visualize, explore, collaborate

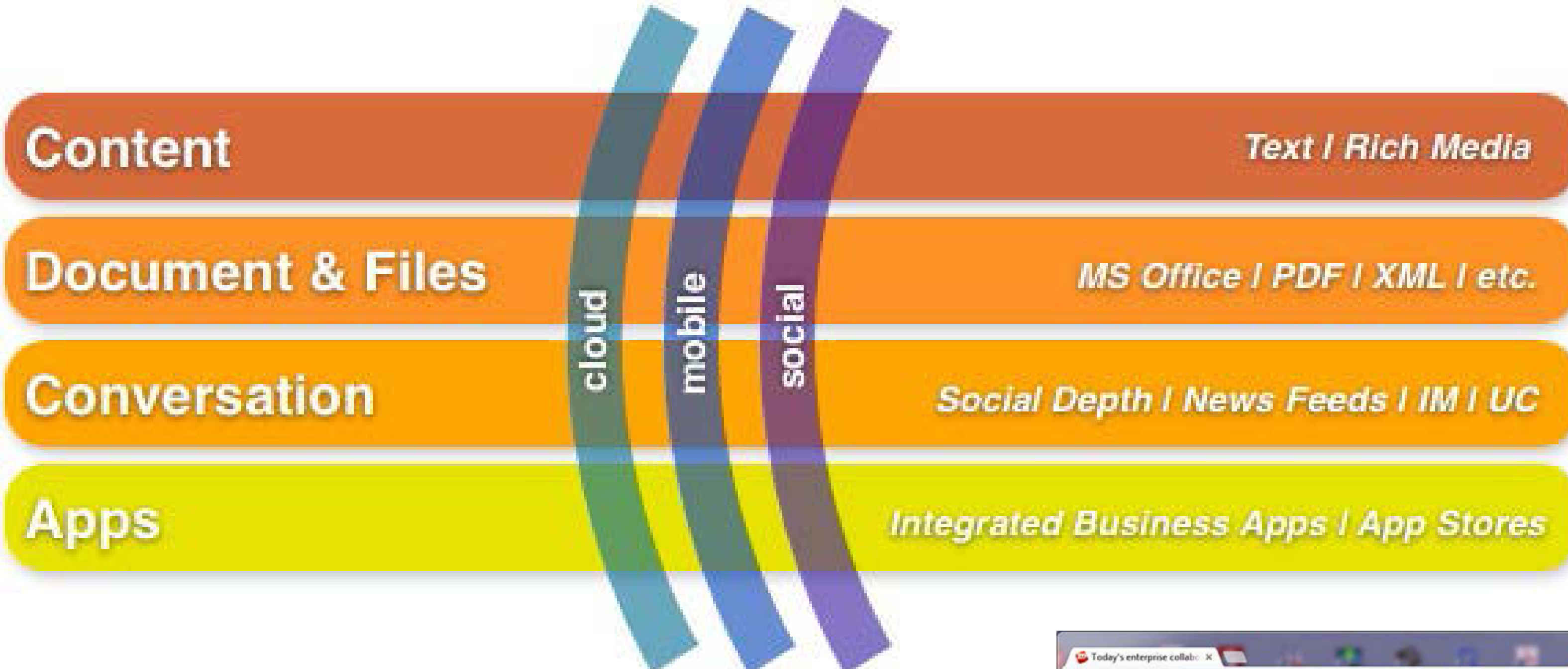
Lots of Collaborative Software



Common Features

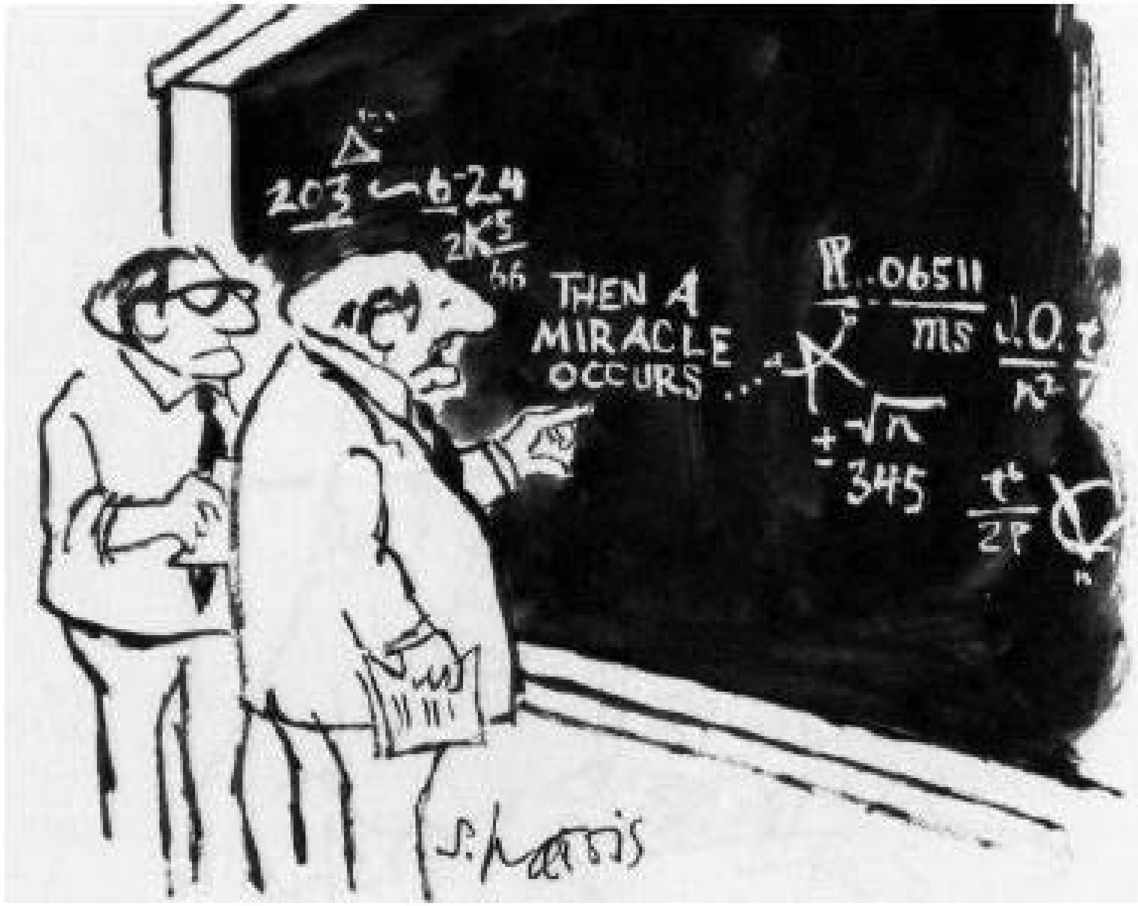
- Slide shows
- Whiteboarding
- Voice chat
- Video chat
- Image capture

Expected Enterprise Collaboration Features
Circa 2013



From http://zdnet.com/blog/hinchcliffe

Technical Collaboration



"I think you should be more explicit here in step two."

from *What's so Funny about Science?* by Sidney Harris (1977)

Technical Collaboration

Missing:

- Mathematics
- Diagrams, graphs
- Geometric figures
- Technical knowledge base
- Document markup

- Scientific software connections (Maple, Mathematica, GeoGebra, R,...)

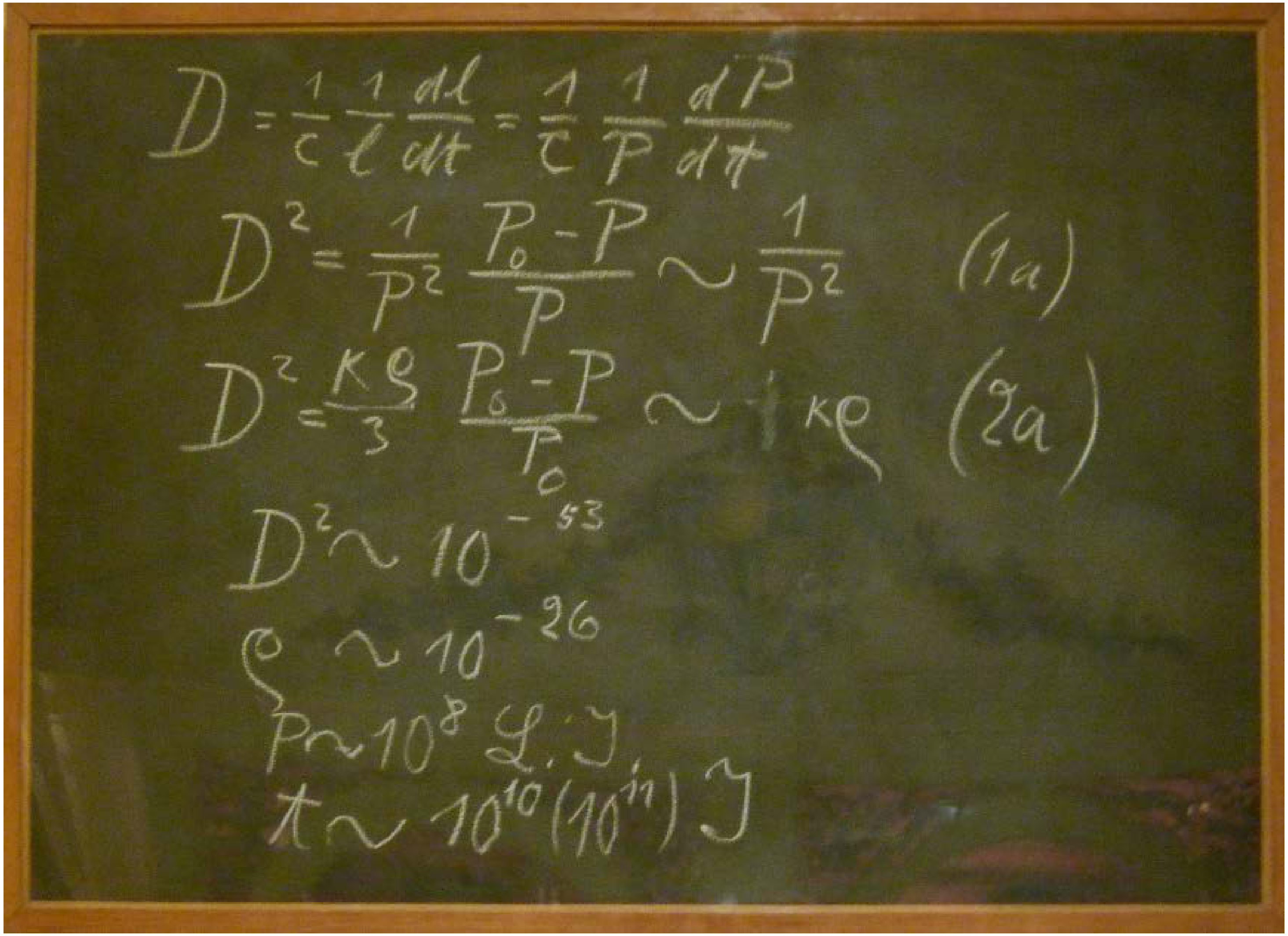
Technical Collaboration

Isn’t a shared whiteboard, with the ability to save images enough????

Technical Collaboration

The Treachery of Images

(La trahison des images)


$D = \frac{1}{c} \frac{1}{l} \frac{dl}{dt} = \frac{1}{c} \frac{1}{P} \frac{dP}{dt}$
$D^2 = \frac{1}{P^2} \frac{P_0 - P}{P} \sim \frac{1}{P^2}$ (1a)
$D^2 = \frac{\kappa\rho}{3} \frac{P_0 - P}{P_0} \sim \kappa\rho$ (2a)
$D^2 \sim 10^{-53}$
$\rho \sim 10^{-26}$
$P \sim 10^8$ L.J.
$t \sim 10^{10} (10^{11})$ J

Einstein's Blackboard

- Einstein to receive honorary doctorate at Oxford, May 1931.
- Lecture at Rhodes House.
- Board retrieved and preserved by Edmund ("Ted") Bowen.
- Nice to look at, but content is **trapped**.

Digital Ink

- Location, time information, sometimes also pressure and angles.
- Capture online pen strokes, *not* images.

- Suitable for
  - **Recognition** algorithms
  - **Semantic** grouping
  - **Annotation**
  - **Manipulation**: search, transformation, archival.

- Problem: Multiple vendor-specific formats.

W3C Recommendation

Ink Markup Language (InkML)

W3C Recommendation 20 September 2011

This version:

http://www.w3.org/TR/2011/REC-InkML-20110920/

Latest version:

http://www.w3.org/TR/InkML

Previous version:

http://www.w3.org/TR/2011/PR-InkML-20110510/

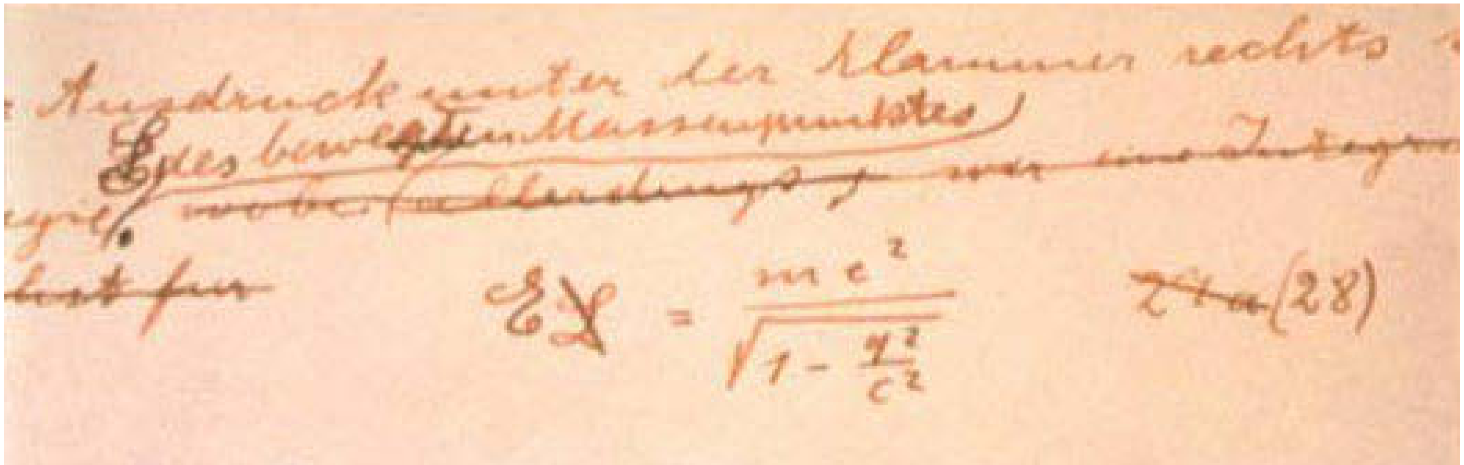
Editors:

Stephen M. Watt, University of Western Ontario and Maplesoft
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)
Grégory Pakosz (until 2007 while at Vision Objects)
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)

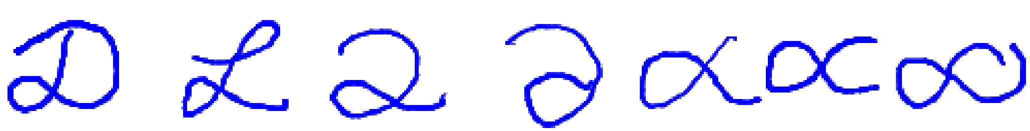
Pen-Based Math

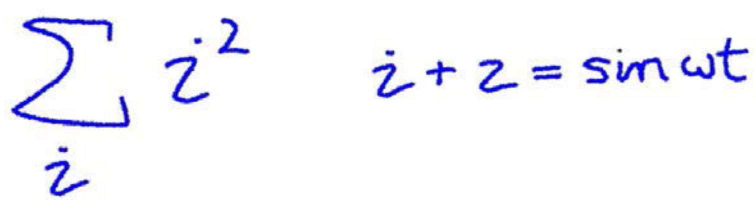


- Input for CAS and document processing.
- 2D editing.
- Computer-assisted collaboration.

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.
  - Many symbols, each person uses a subset.
  - No fixed dictionary for disambiguation.





Character Recognition

- A story about a UI proposal
- A story about three statisticians
- Concentrate on character recognition
- Several projects ignore this problem

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*

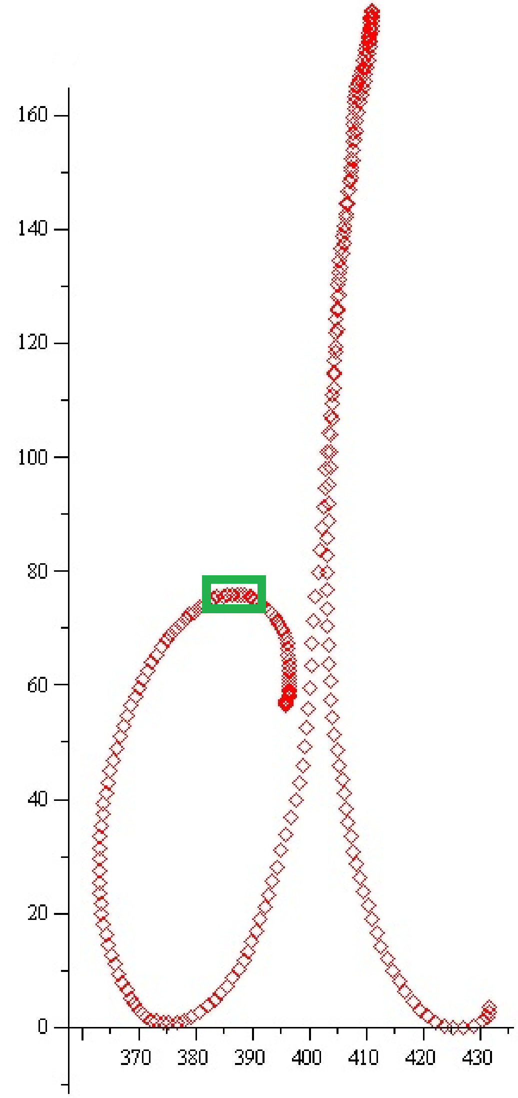
  Use a classification method, such as
  - Nearest neighbour, Subspace projection, Cluster analysis, Support Vector Machine

  THEN
- Rank choices by consulting dictionary

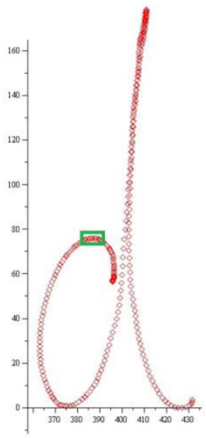
Difficulties

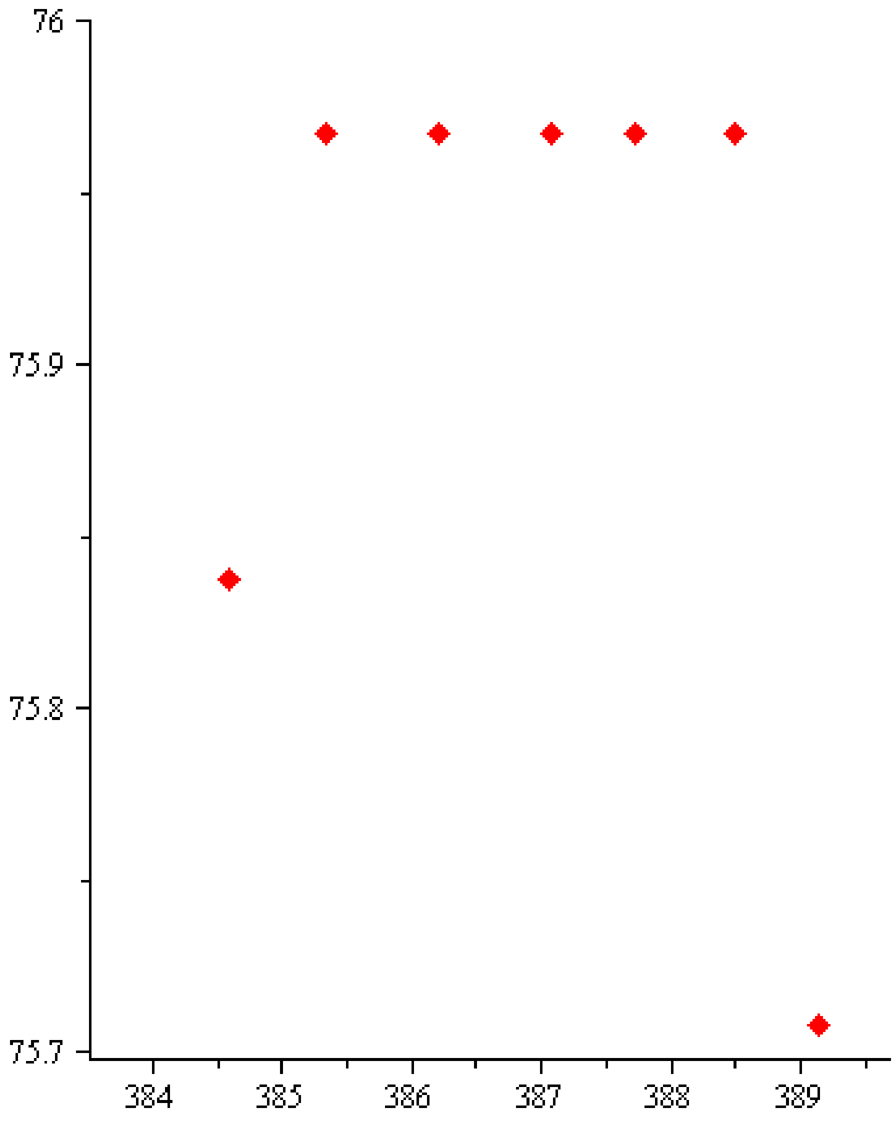
- Having many similar characters (e.g. for math) means comparison against all possible symbol models is slow.
- Determining features from points
  - Requires many *ad hoc* parameters.
  - Replaces measured points with interpolations
  - It is not clear how many points to keep,
    and most methods depend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

What the Computer Sees



What the Computer Sees





Orthogonal Series Representation

- **Main idea**:
  Represent traces as curves, not discrete points and *c*oordinate curves as truncated orthogonal series.

Orthogonal Series

- Start with inner product on a space of functions, e.g.

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

- Functions $\phi_i(t)$ give an orthogonal basis if we can write

$$f(t) = \sum_{i=0}^{\infty} f_i \, \phi_i(t) \qquad \text{and} \qquad \langle \phi_i, \phi_j \rangle = 0 \text{ if } i \neq j$$

Then $f_i = \langle f, \phi_i \rangle / \langle \phi_i, \phi_i \rangle$.

- If sum is truncated, f is approximated.

- Obtain orthogonal basis from any basis set, e.g. $\{1, t, t^2, \dots\}$, by Gram-Schmidt process.

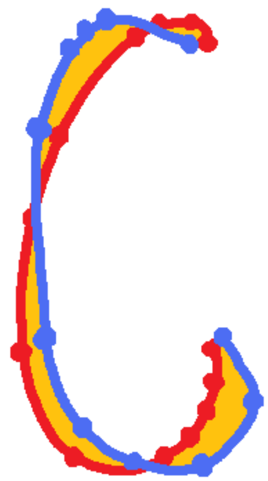
Orthogonal Series Representation

- **Main idea**:
  Represent traces as curves, not discrete points and 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

Distance Between Curves

- **Elastic matching:**
- Approximate the variation between curves by some fn of distances between sample 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

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

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

$$\rho^2(C,\bar{C}) = \int_a^b \left[\left(x(t) - \bar{x}(t)\right)^2 + \left(y(t) - \bar{y}(t)\right)^2 \right] dt = \int_a^b [\xi(t)^2 + \eta(t)^2] dt$$

$$\approx \int_a^b \left[\sum_{i=0}^{d} \xi_i^2 \phi_i^2(t) + \text{cross terms} + \sum_{i=0}^{d} \eta_i^2 \phi_i^2(t) + \text{cross terms} \right] dt$$

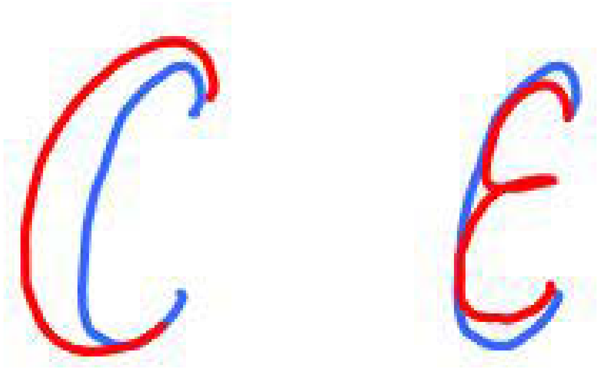
$$= \sum_{i=0}^{d} \xi_i^2 + \sum_{i=0}^{d} \eta_i^2$$

- *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!

Problems

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

- Low RMS does not mean similar shape.



Problem 1. On-Line Coefficients

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

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

- *The answer is "Yes"!*

Problem 1. On-Line Coefficients

- Use modified Legendre polynomials P_i as basis
  on the interval $[0, 1]$, 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.

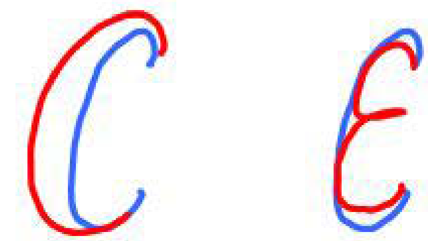
- *As the sample points are collected*, numerically integrate the moments $\int \lambda^i f(\lambda) d\lambda$.

- After last point, compute series coefficients for f
  with domain and range scaled to $[0,1]$.
  This uses a single linear transformation of the moments.

Problem 1. On-Line Coefficients

- Approach works for any inner product with linear weighting.
- This is the Hausdorff moment problem (1921), shown to be unstable by Talenti (1987).
- It is just fine, however, for the dimensions we need.

Problem 2. Shape vs Variation

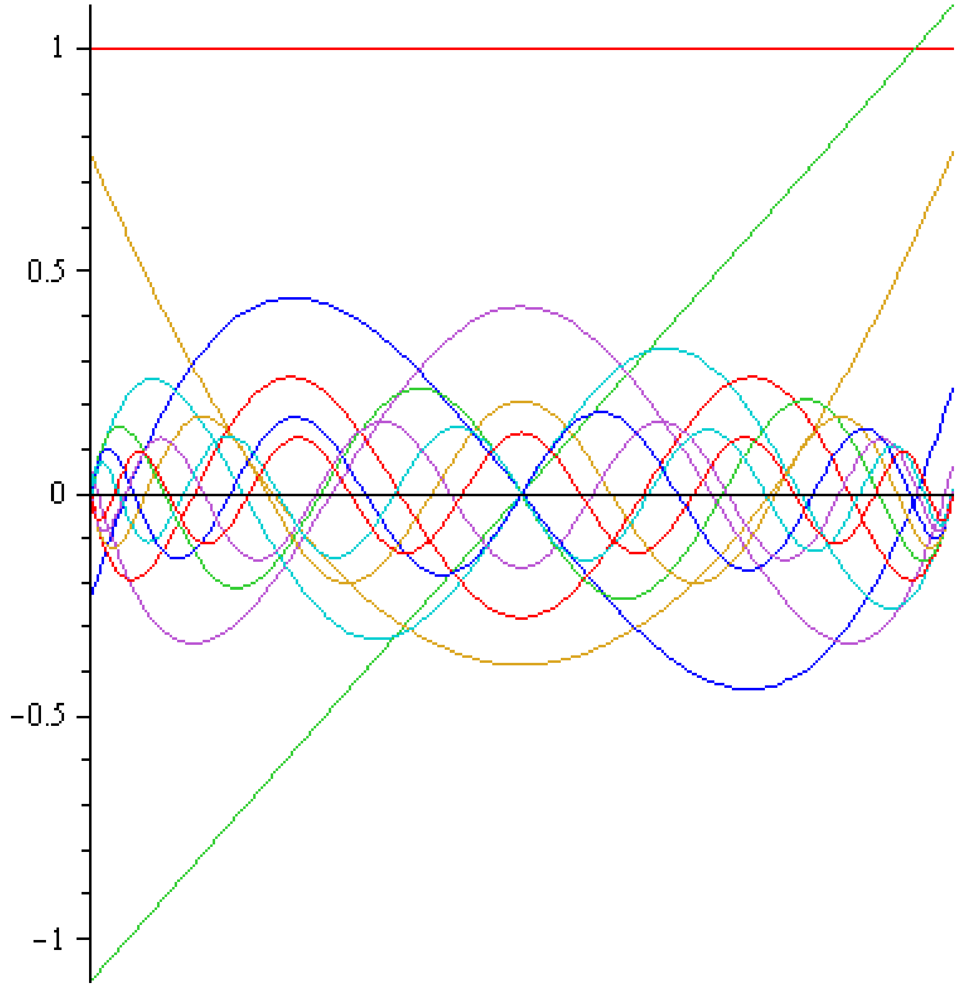


- The corners are not in the right places.
- Work in jet space to force coords & derivs to be close.
- Legendre-Sobolev inner product.

$$\langle f, g \rangle_{LS} = \int_a^b f(t)g(t)dt + \mu_1 \int_a^b f'(t)g'(t)dt + \mu_2 \int_a^b f^{\prime\prime(t)} g^{\prime\prime(t)} dt + \cdots$$

- 1^{st} jet space sufficient.
  - Choose μ_1 experimentally to maximize reco rate.
  - Can be also done on-line. [Golubitsky + SMW 2008, 2009]

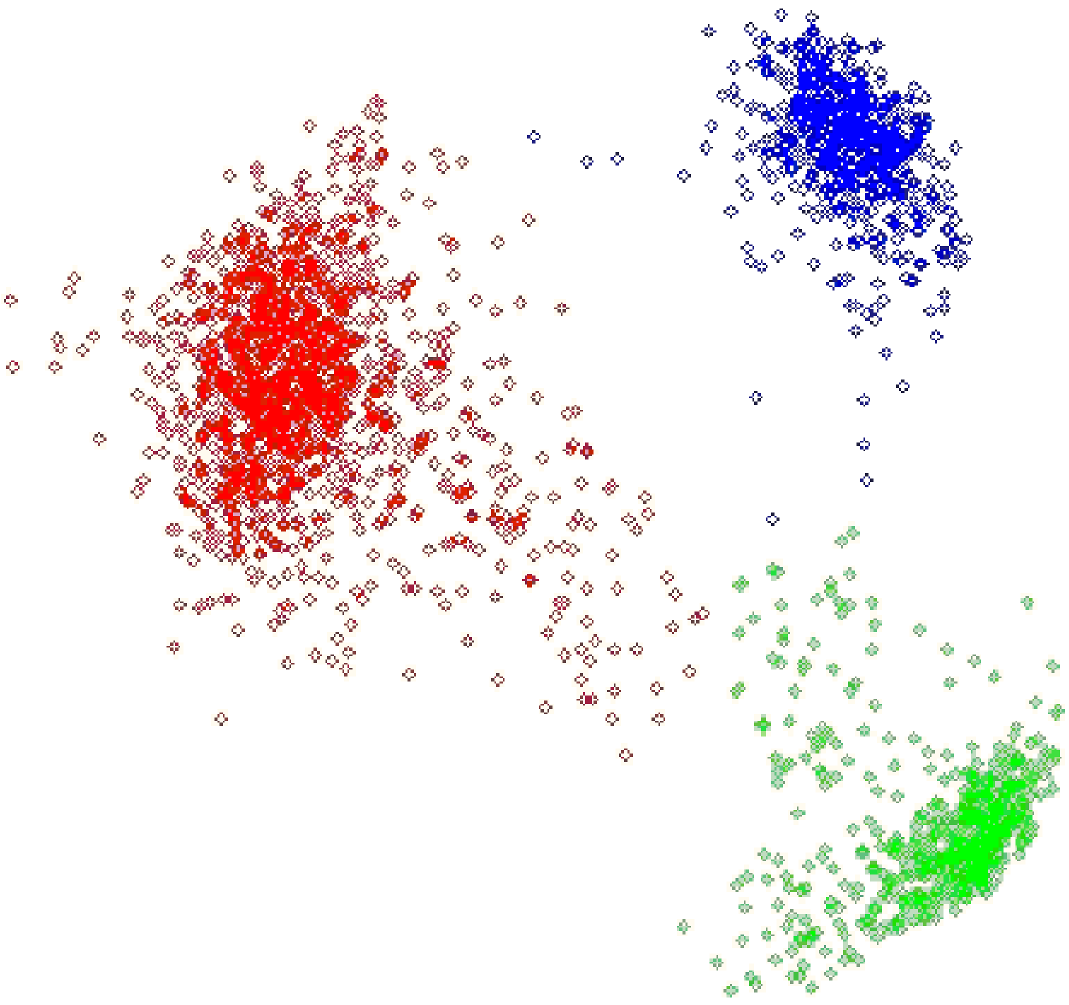
Legendre-Sobolev Basis



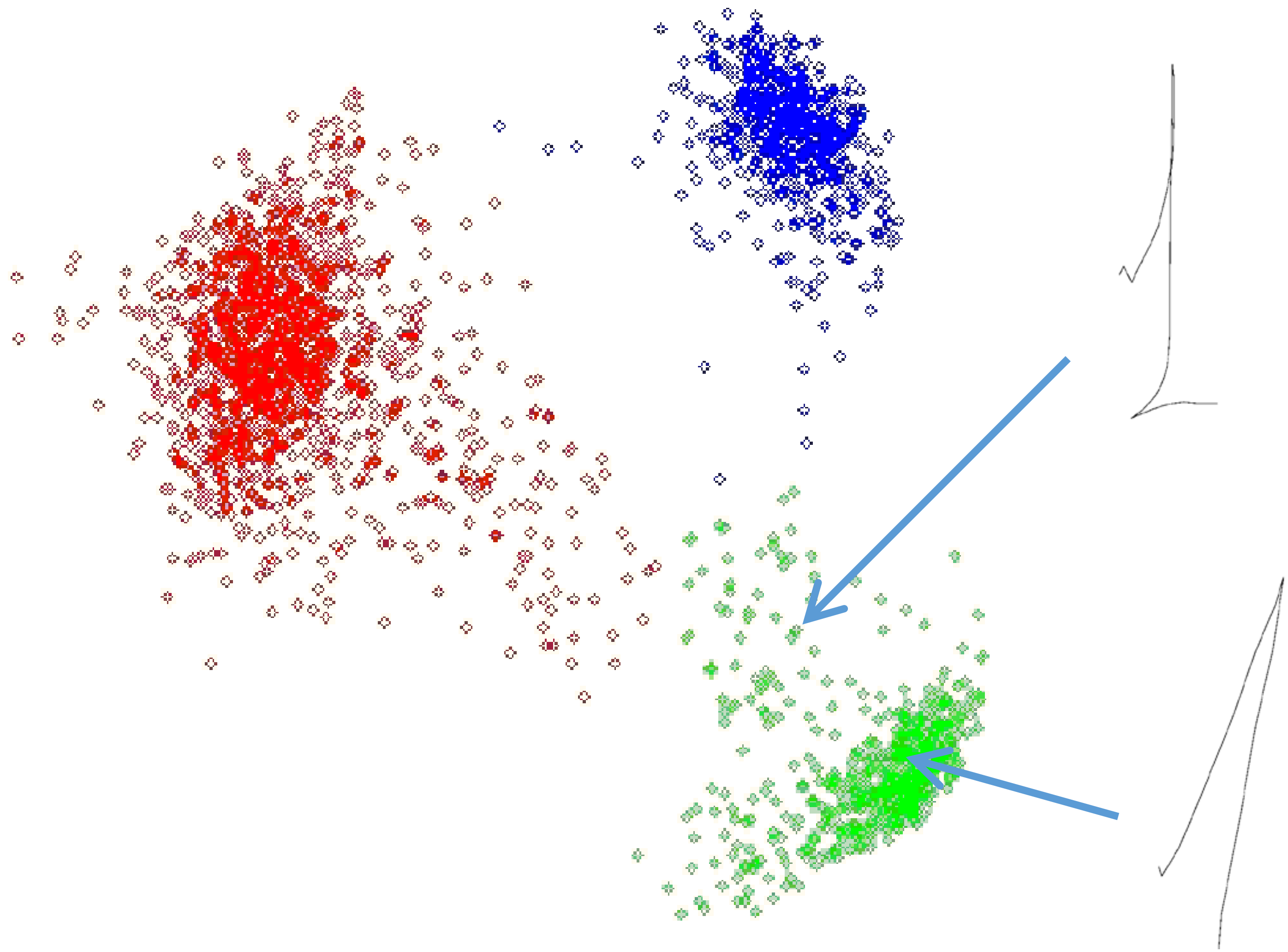
Life in an Inner Product Space

- 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]
  - **>99% linear separability => convexity of classes**
  - A useful notion of distance between curves *that is very fast to compute*

Linear Separability



Linear Separability

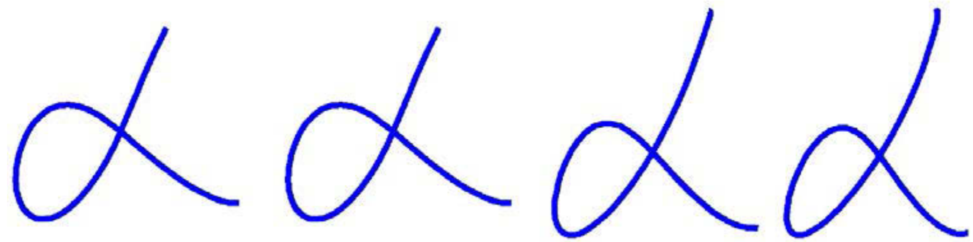


Comparison of Sample to Models

- Use Euclidean distance in the coefficient space.
- Can trace through SVM-induced cells incrementally.
- Normed space for characters gives other advantages.

The Joy of Convexity

- Convexity ⇒ Linear homotopies stay within a class



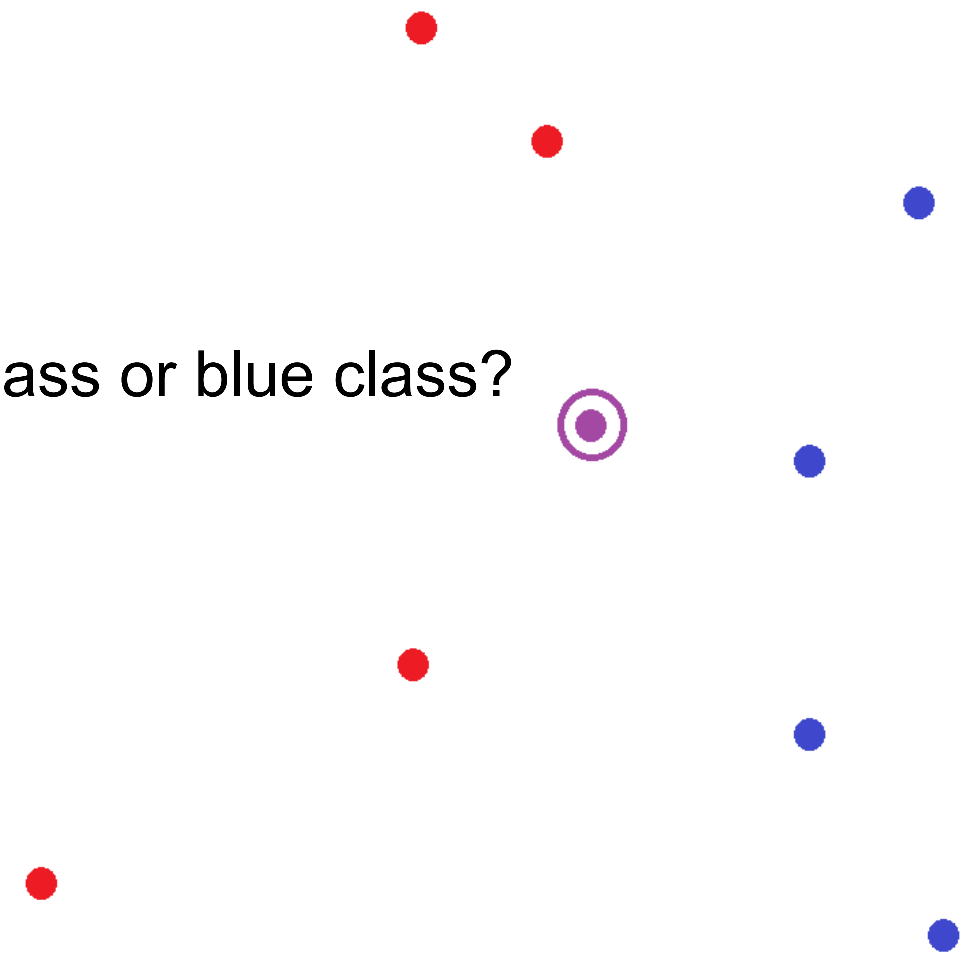
$$C = (1 - t)\,A + t\,B$$

- Can compute distance of a sample to this line

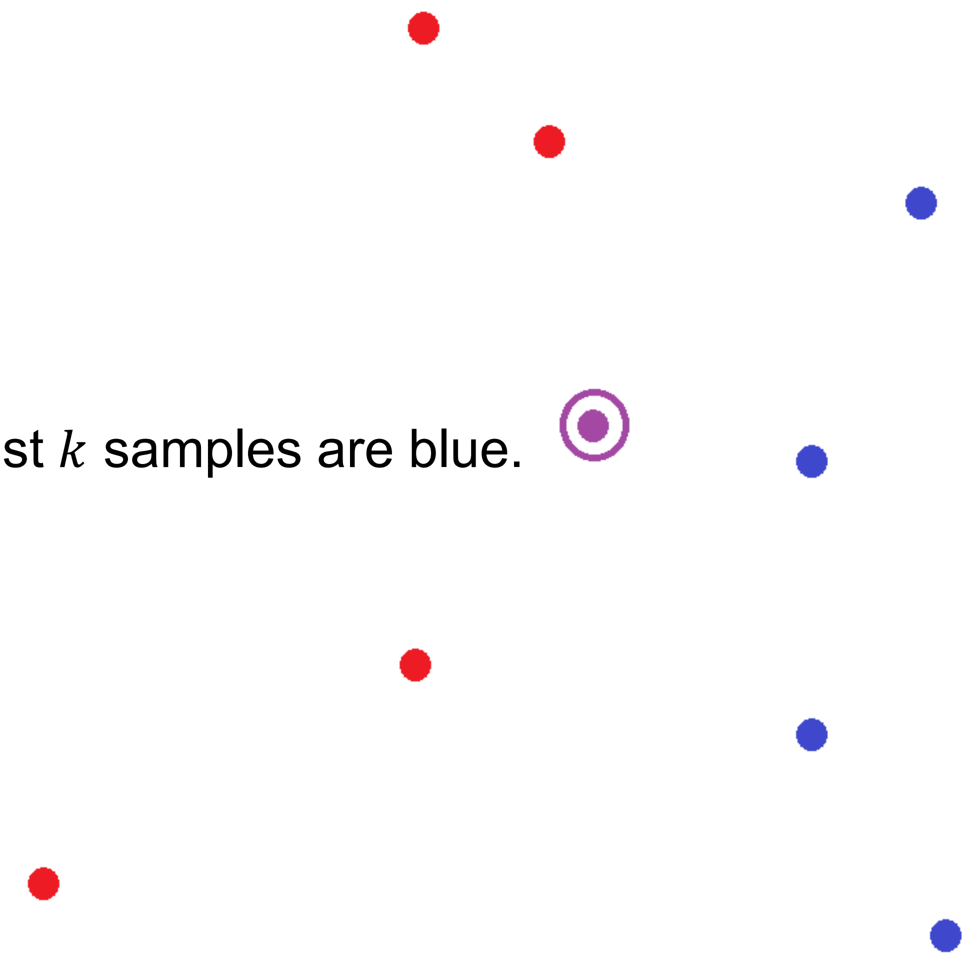
- Distance to **convex hull** of nearest neighbors in class gives best recognition [Golubitsky+SMW 2009,2010]

Choosing between Alternatives

Red class or blue class?

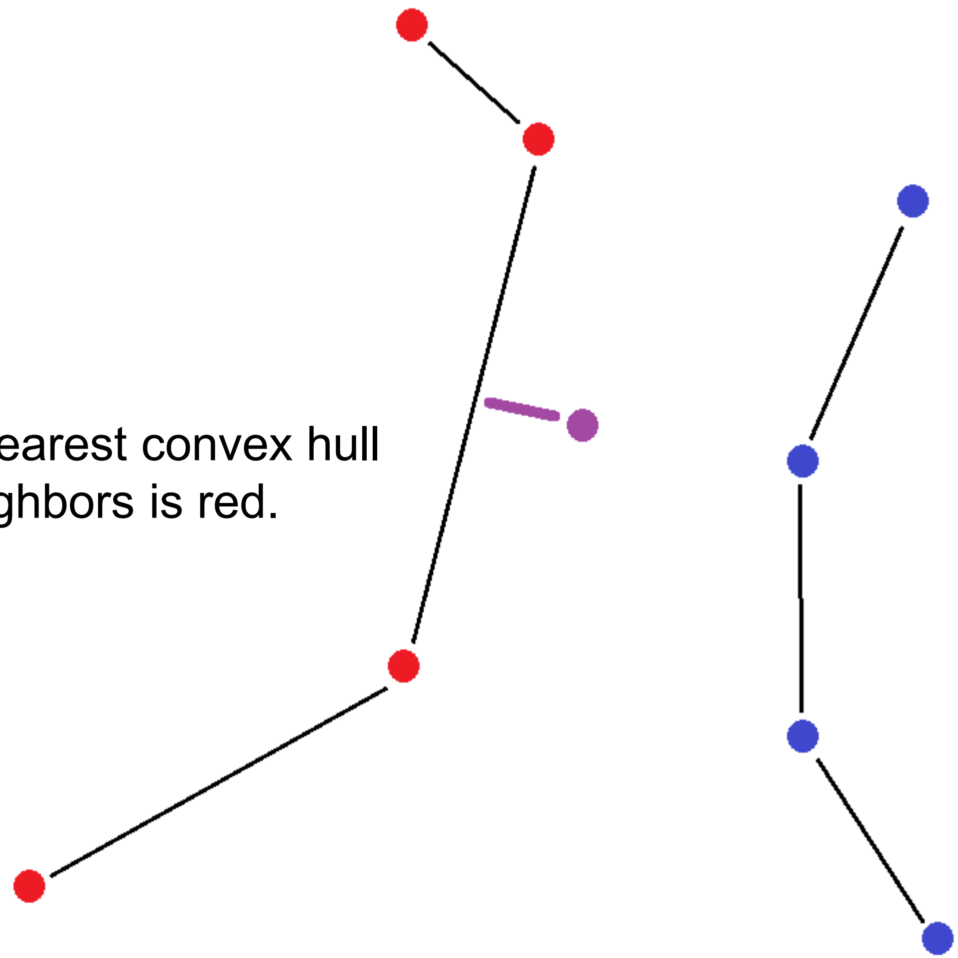


Choosing between Alternatives



The nearest k samples are blue.

Choosing between Alternatives

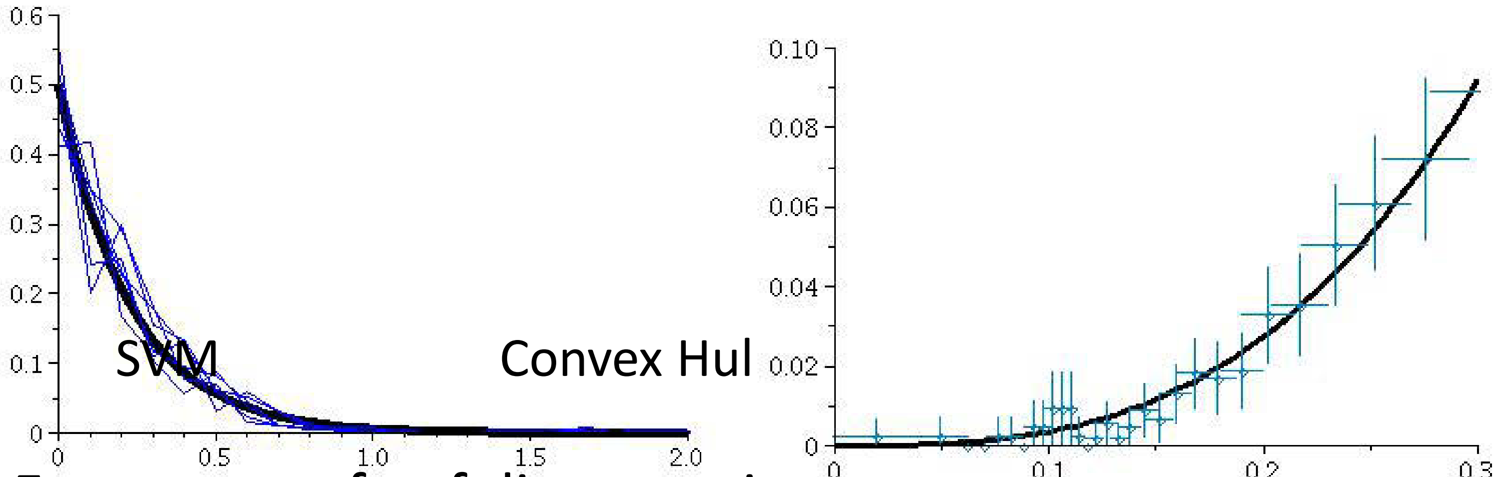


The nearest convex hull
of neighbors is red.

Training

- Using CHKNN allows training with relatively few samples. (Dozens vs Thousands per class)

Error Rates as Fn of Distance

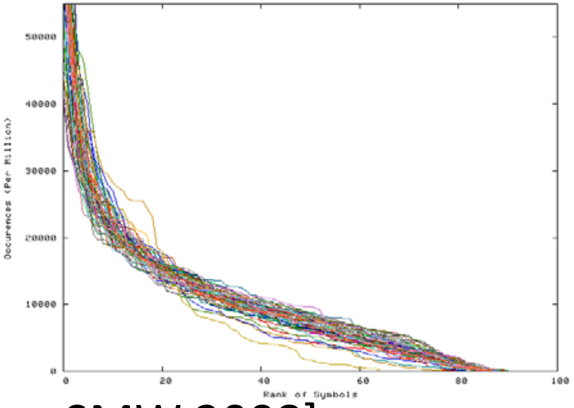


- Error rate as fn of distance gives confidence measure for classifiers [MKM – Golubitsky + SMW 2009]

Combining with Statistical 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]



Baseline Estimation

- Figure out baseline from the characters,
  rather than the other way around, which is more ususal.

- We can locate some important features by identifying special points.

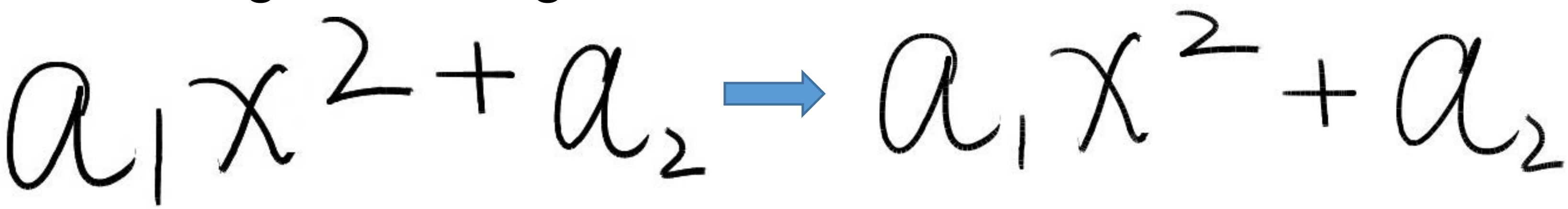
We refer to a point such as this, that determines the height of a metric line, as a *determining point*.

Baseline Estimation

- Juxtaposition ambiguity

P9 Pq pq p9

- Handwriting neatening

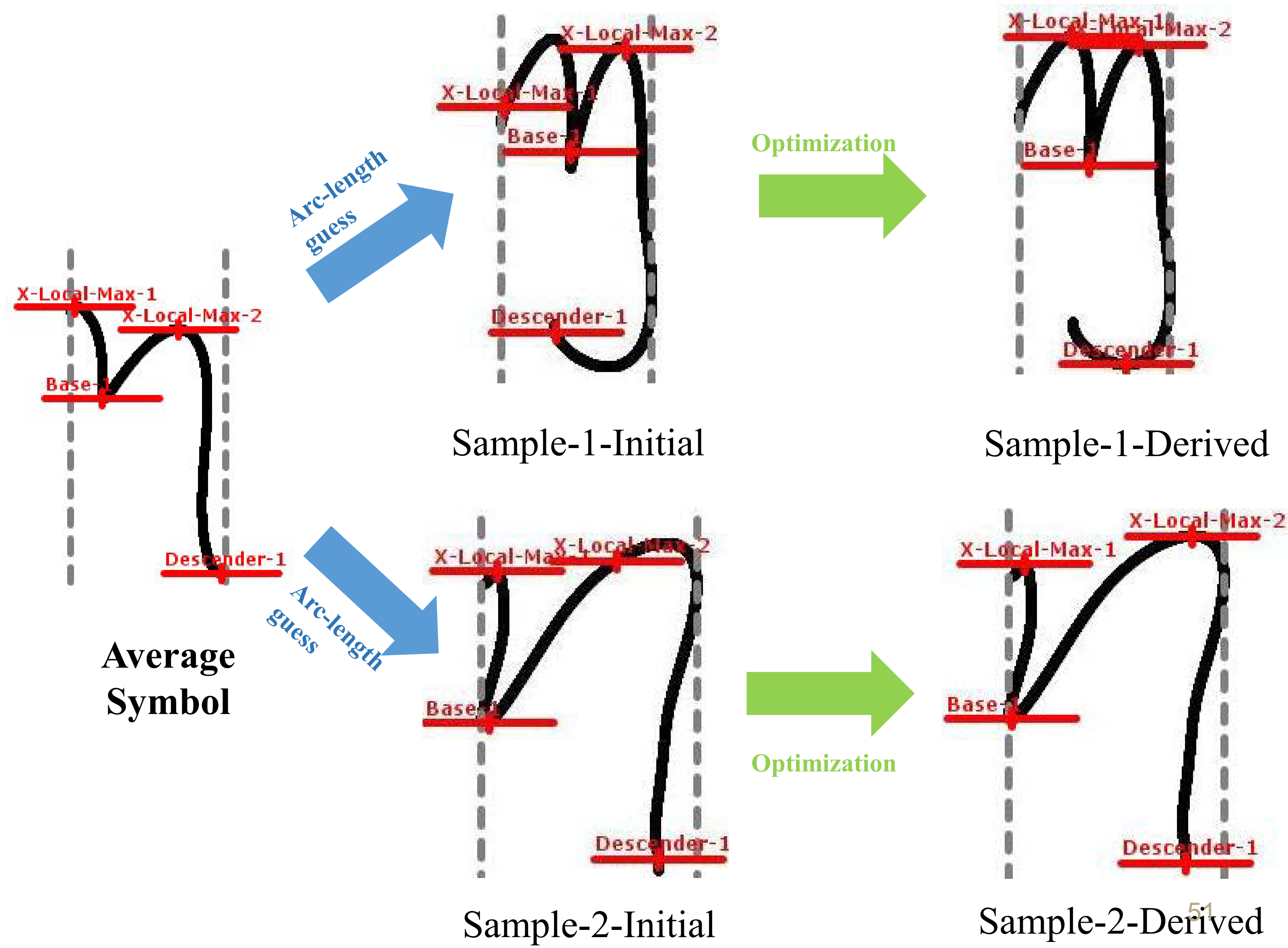


Average Symbol

- The average symbol of a set of known samples for a class can be computed as the average point in the functional space,

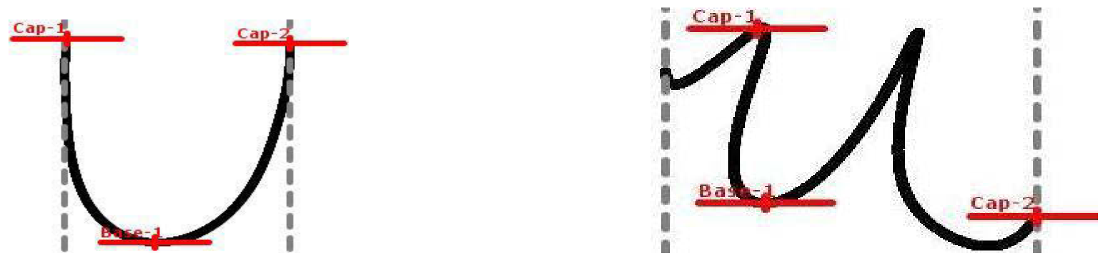
$$\bar{C} = \sum_{i=1}^{n} C_i / n$$

Deriving from a Reference Symbol



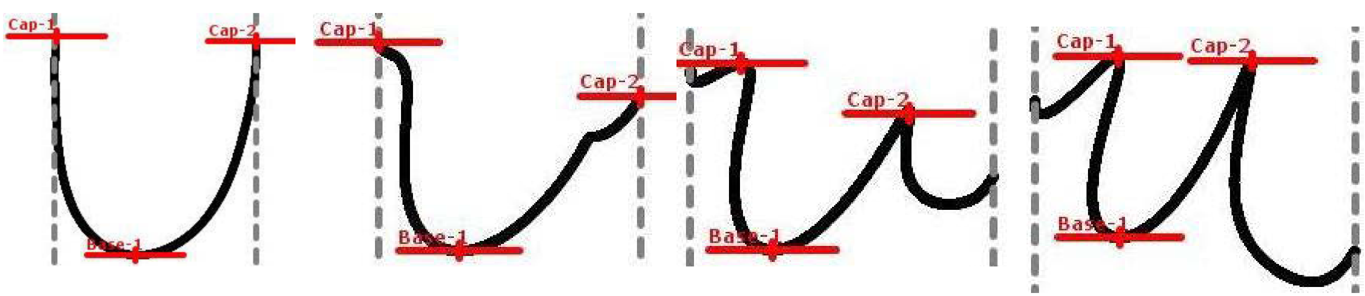
Using Homotopy

- Some samples are far away from the reference symbol.



Average Target

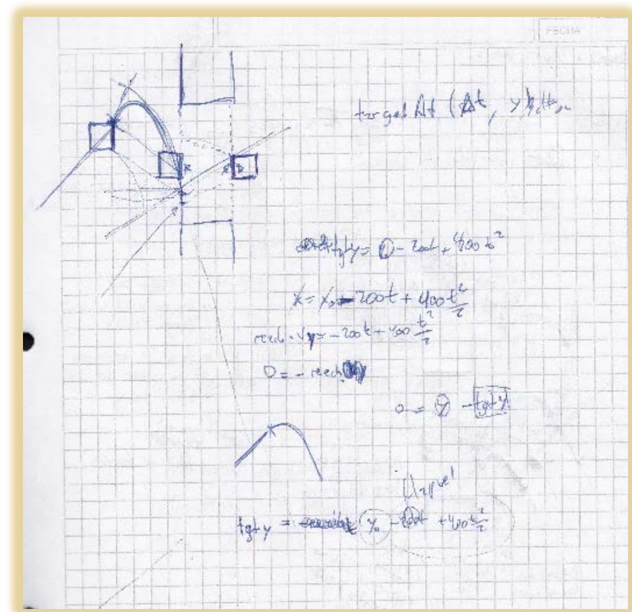
- We use a homotopy between the reference symbol and the target sample in a multi-step method.



Average Step-1 Step-2 Target

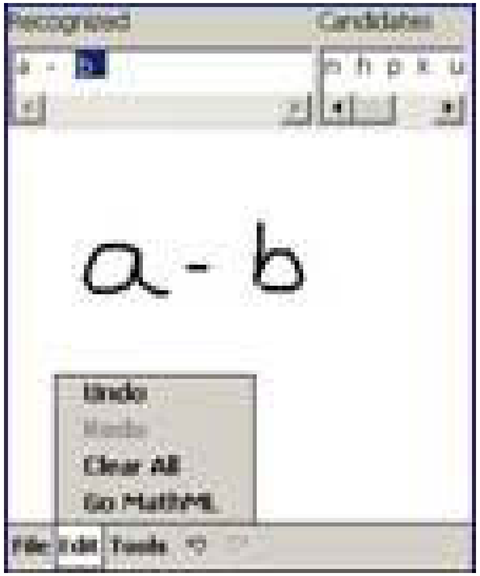
Prior Generations of Software

- 2000 Cross Pad:



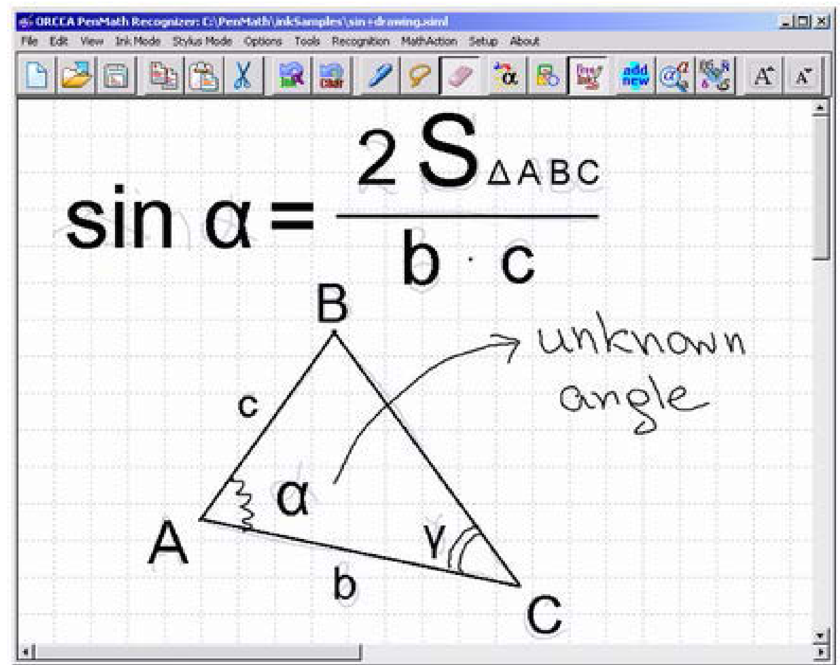
Prior Generations of Software

- 2002 Pocket PC:



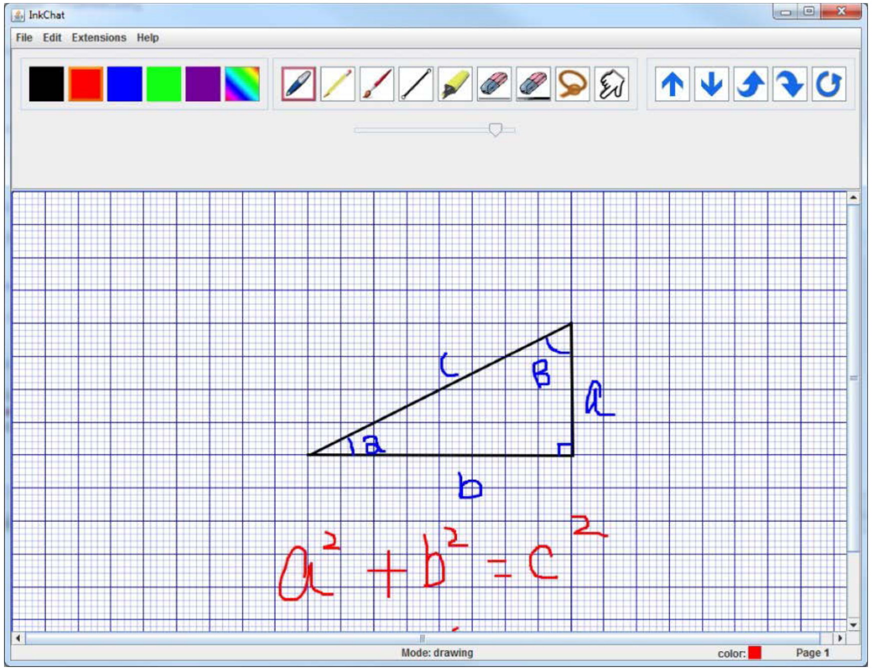
Prior Generations of Software

- 2002-2008 Tablet PC:



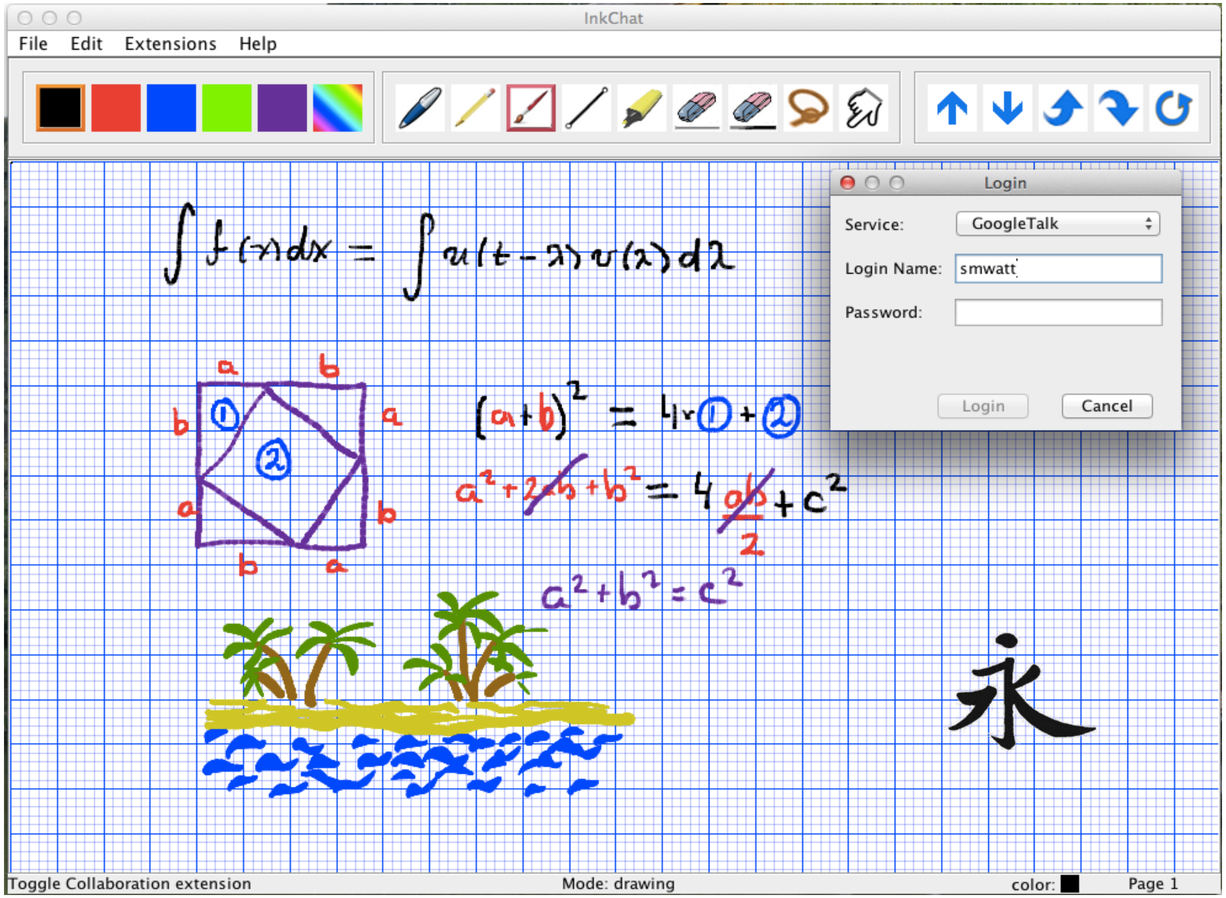
Prior Generations of Software

- 2008-2013 Java Application:



InkChat (Java Version)

- Skype and GTalk add-on to the Java application.



Problems

- Requires installation:
  - Big hassle for someone to use only once in a while or on all their machines.
- Limited portability:
  - Users expect versions on Android, iOS, Windows, Mac OSX, Linux, etc…
  - Incompatible software bases
  - Flakey, moving APIs
- Need to support multiple devices.
  - Nowadays a single user will want to work across many devices.

Solution

- Use browser infrastructure.

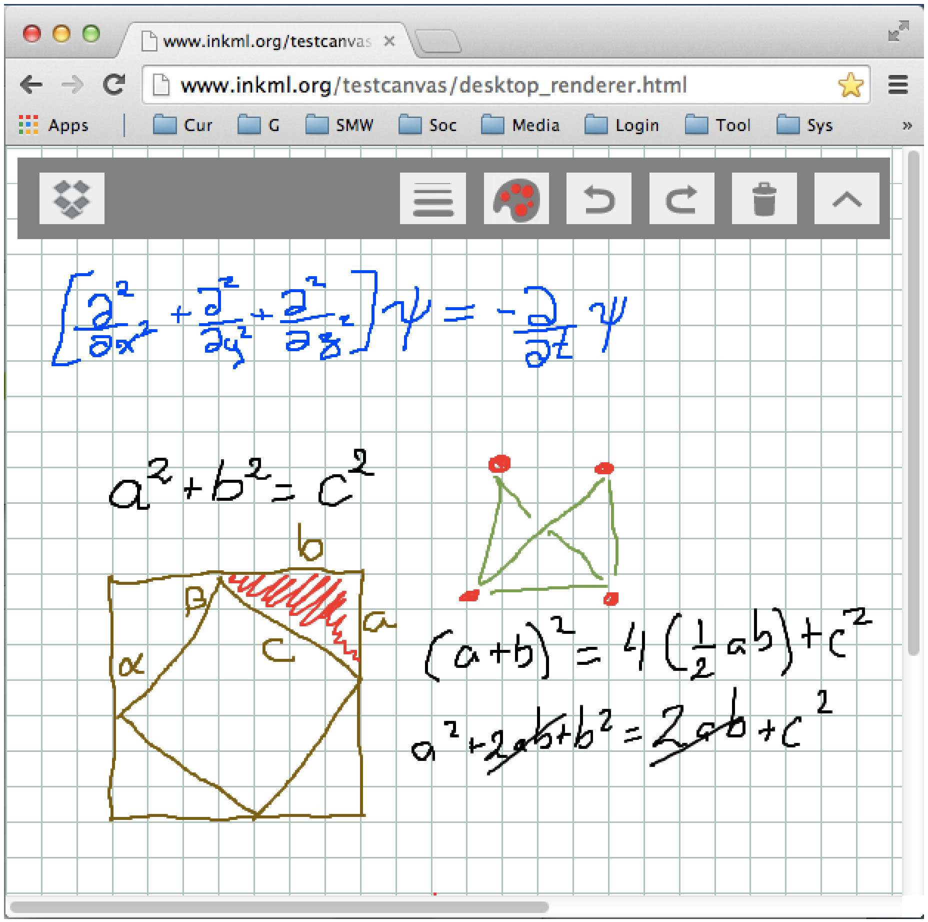
Solution

- Use browser infrastructure.

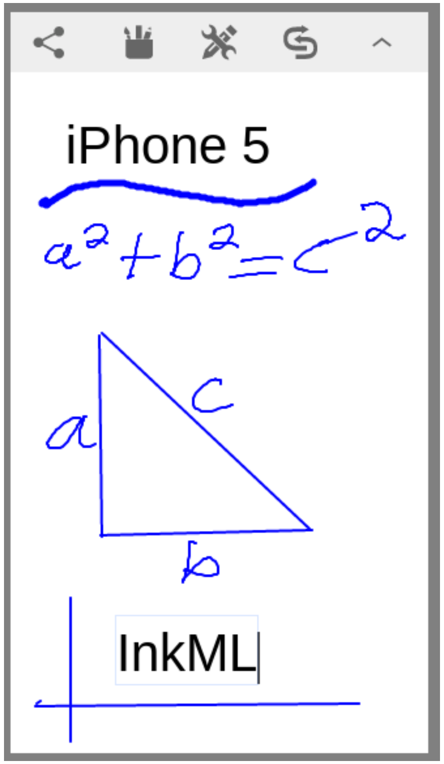
- JavaScript is not a great language for large projects, but…..
  - It is ubiquitous: Telephones, tablets, laptops, …
  - Libraries for many UI elements
  - Our new recognition algorithms are fast enough ☺

- Rapid development:
  - Prototype developed in 3 months by 3 students.

Current Generation



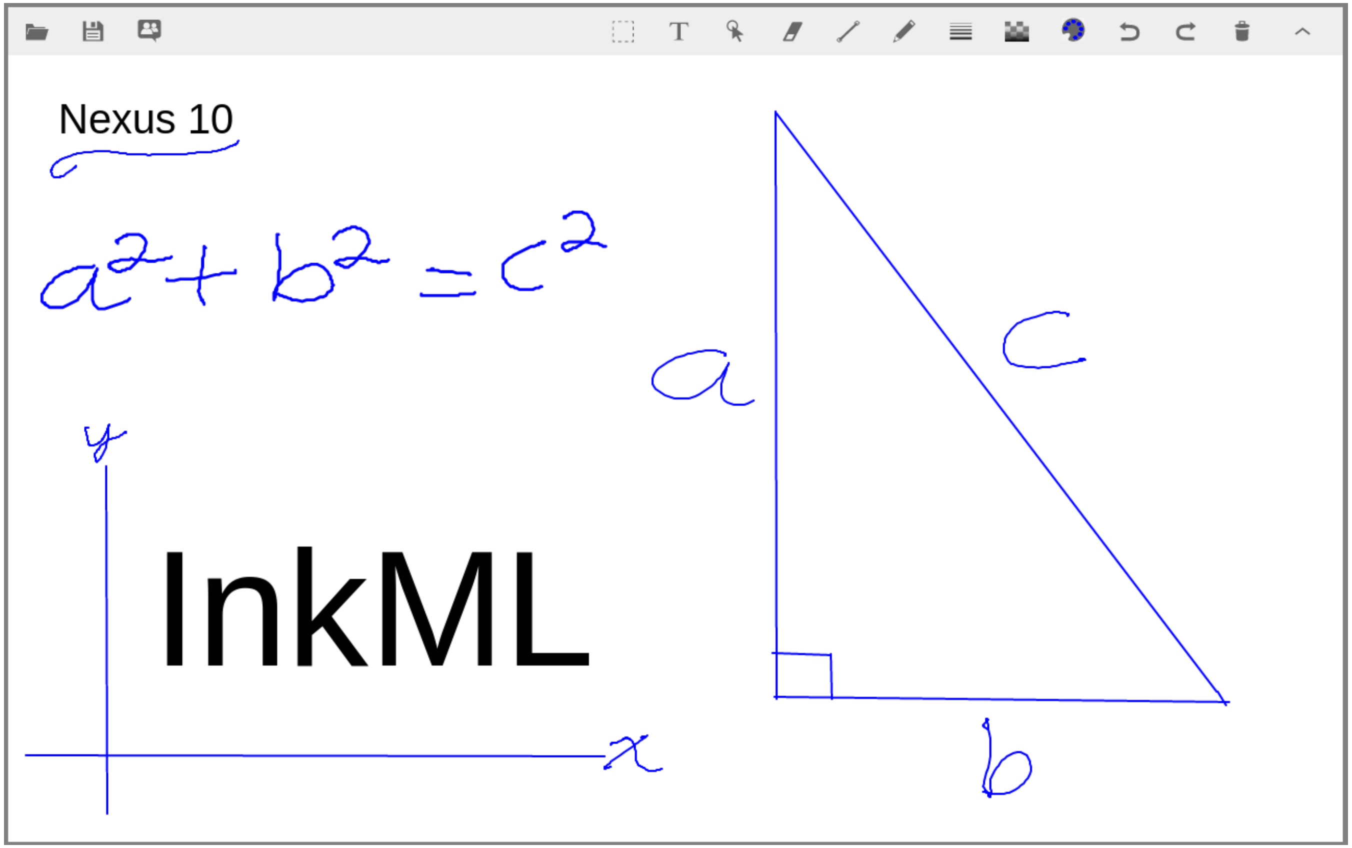
Desktop



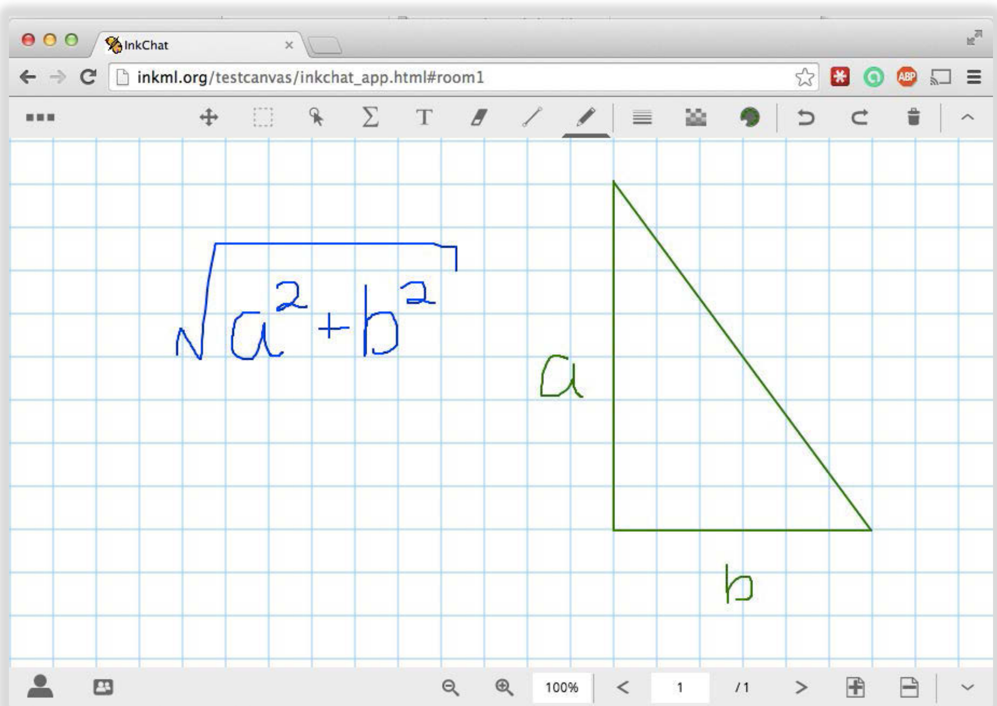
Telephone

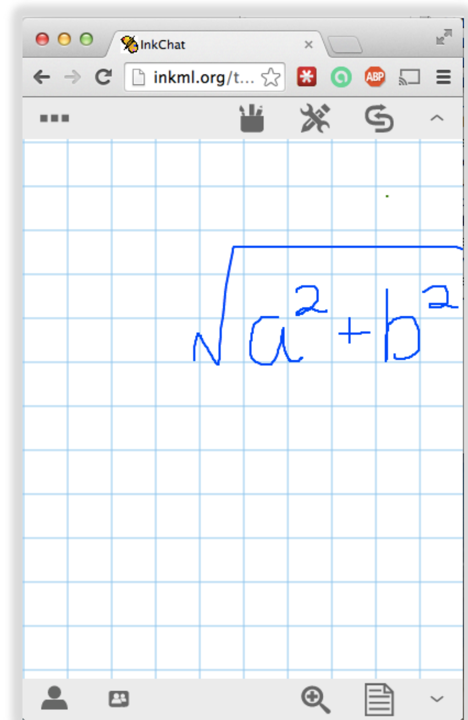
Current Generation

Tablet

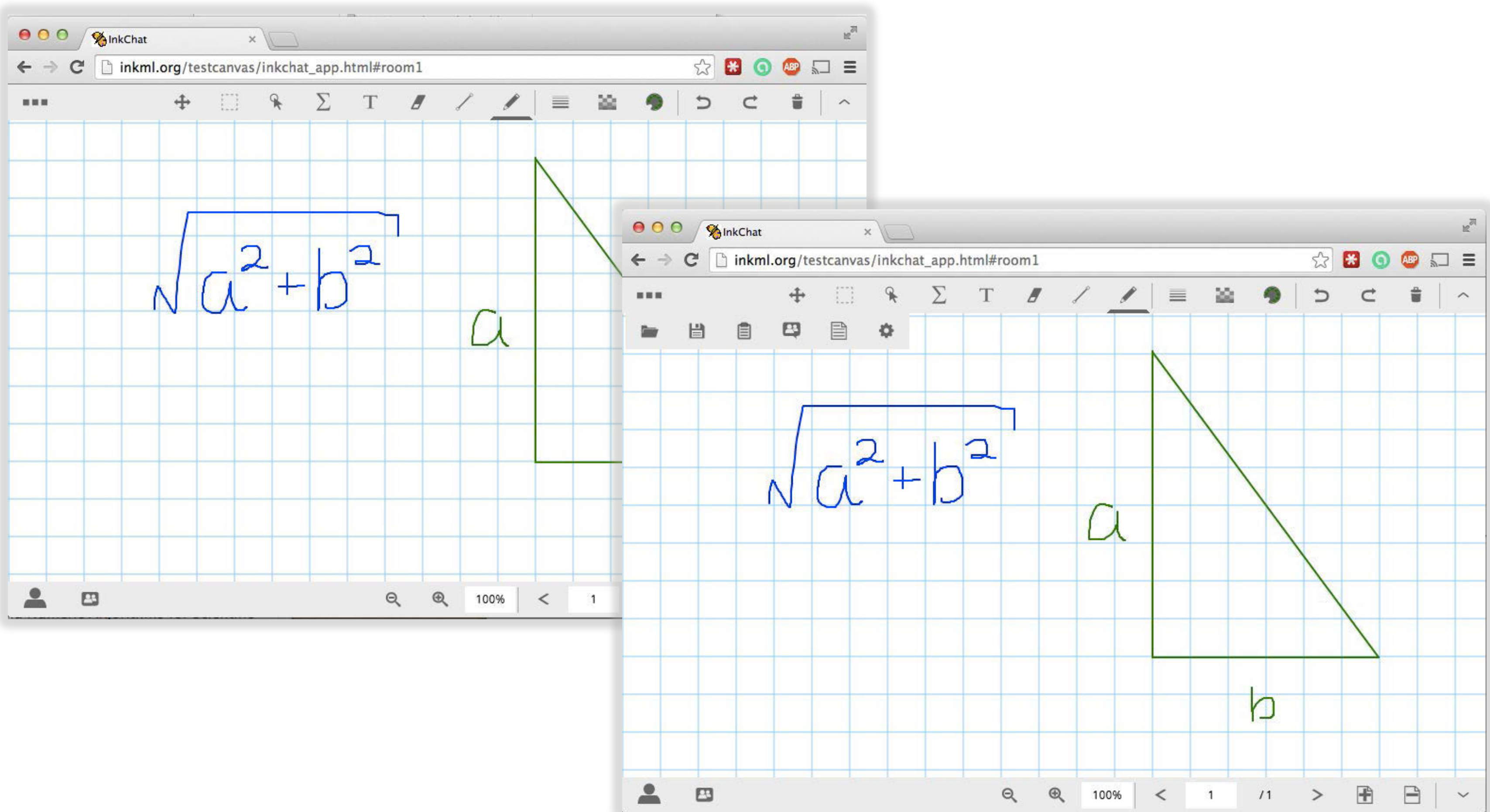


Simple Interface
with device-adapted menus

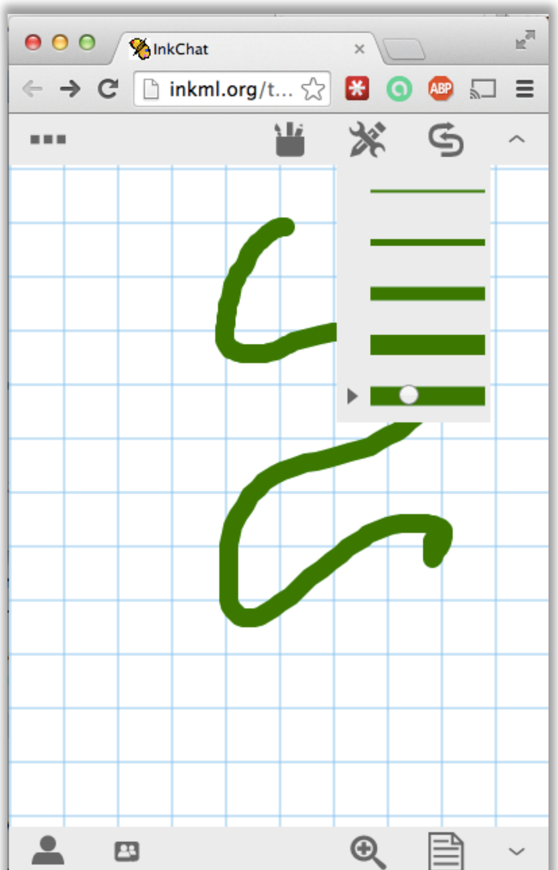


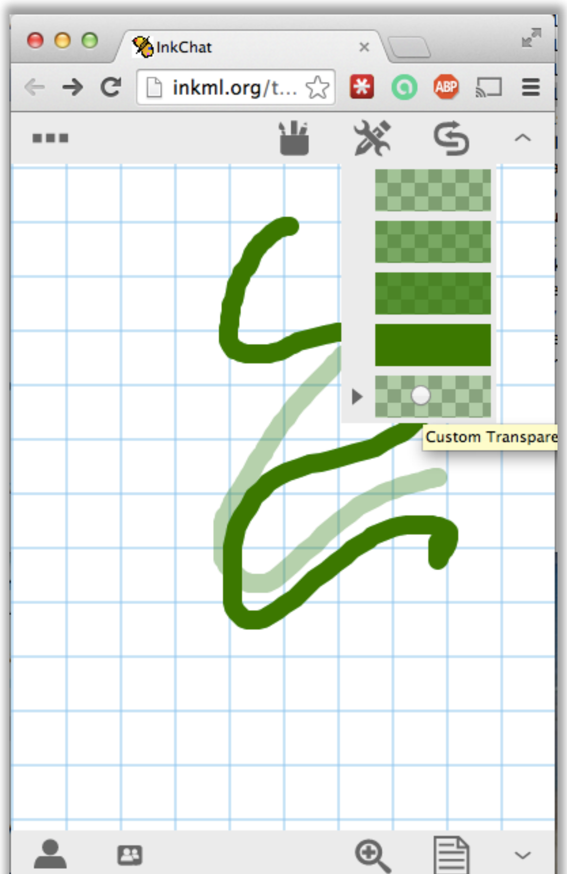


Simple Interface with device-adapted menus

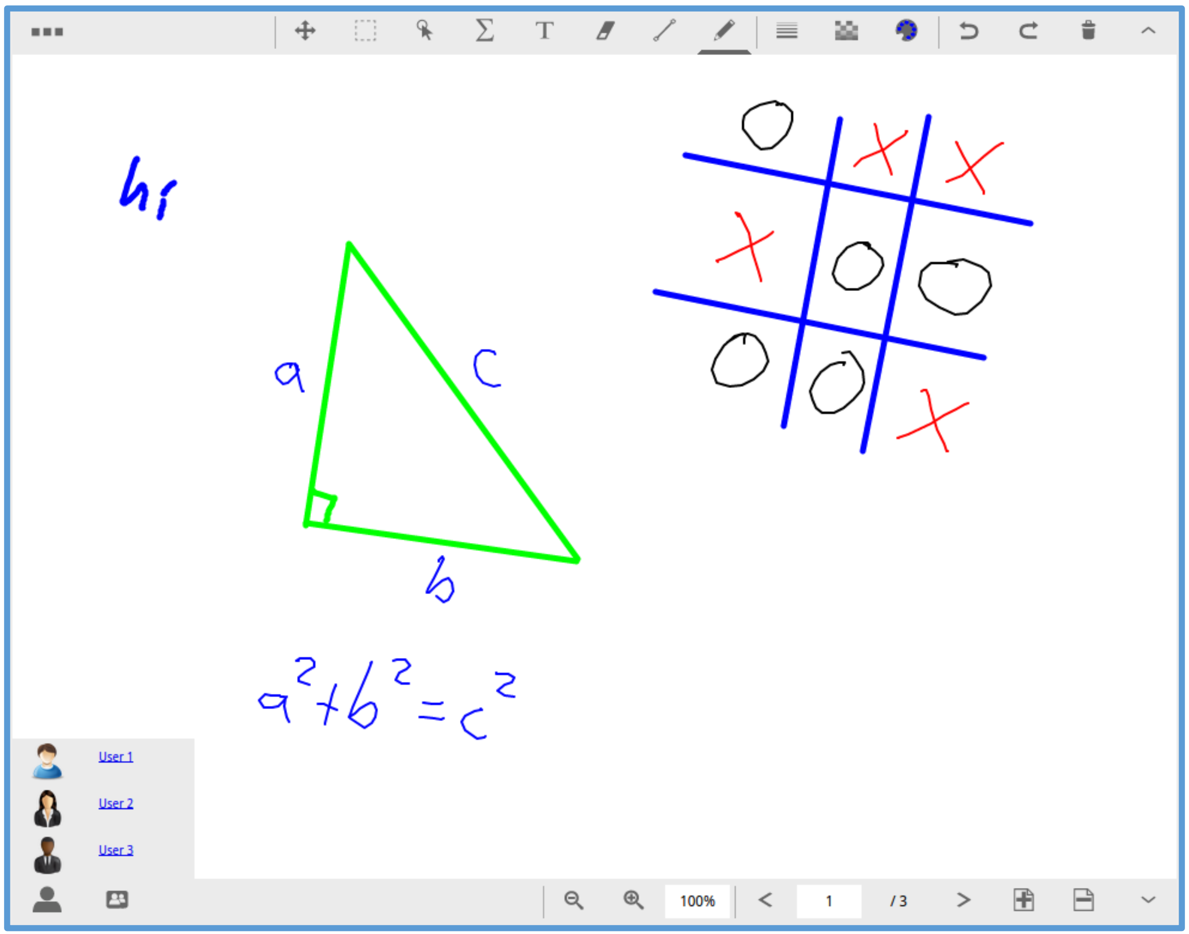


Ink Controls

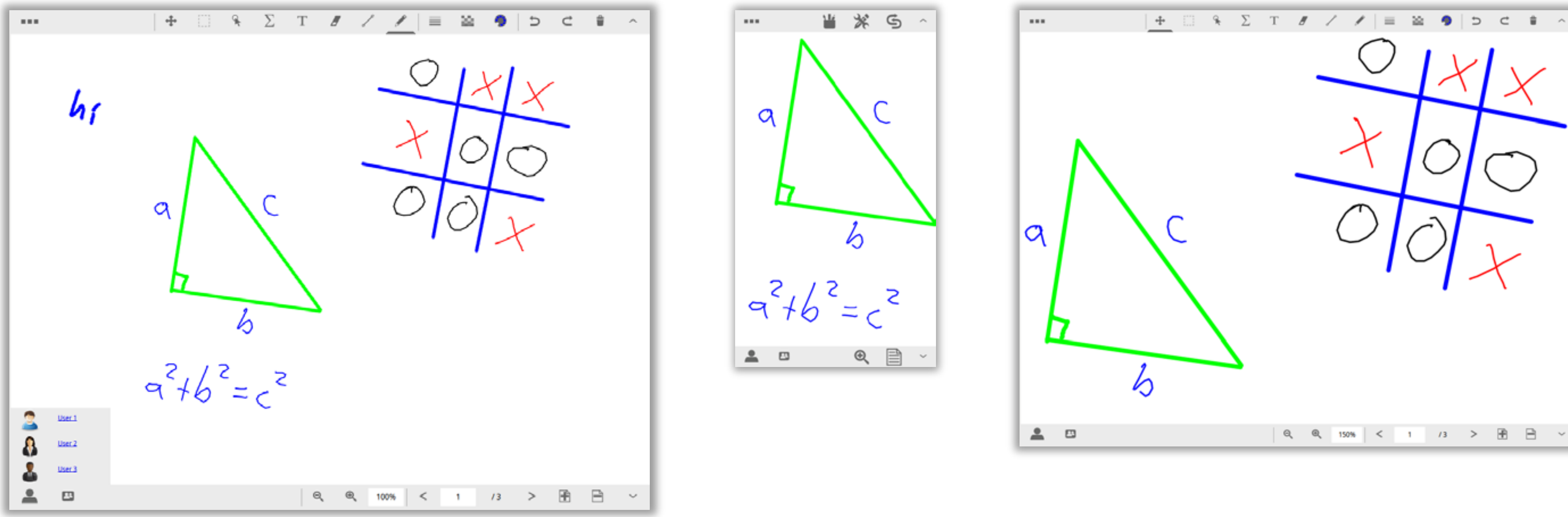




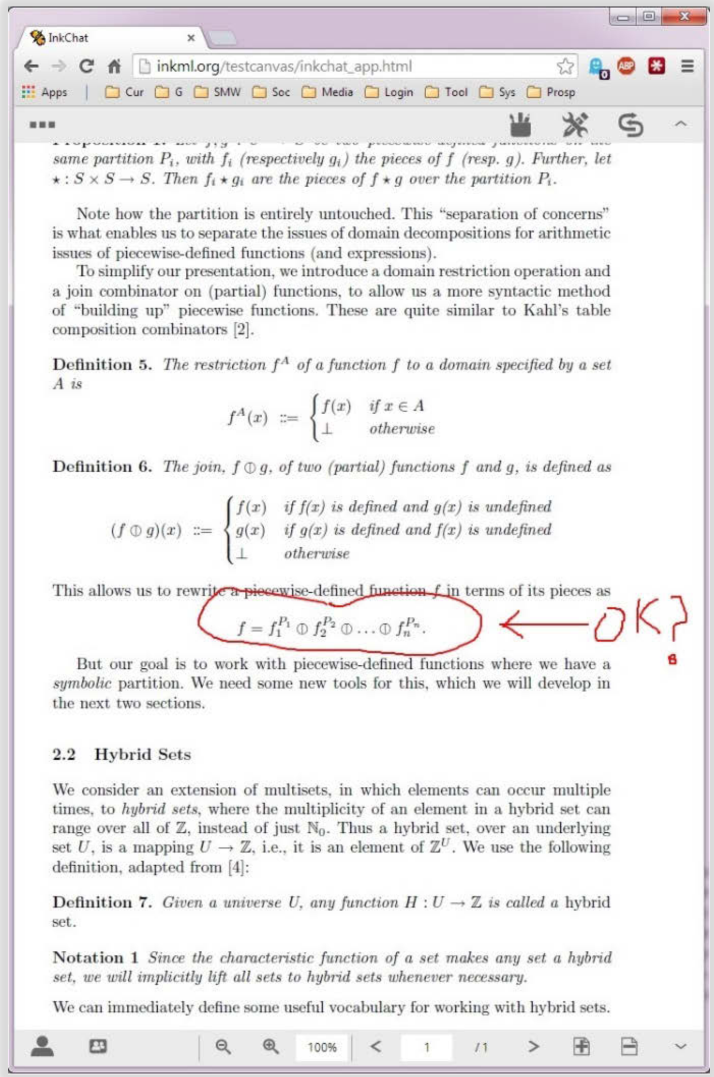
Collaboration:
Multiple Users Connected to Same URI



Collaboration:
Different Viewports from Different Devices



Collaboration:
Pointers for Discussions



same partition P_i, with f_i (respectively g_i) the pieces of f (resp. g). Further, let $\star : S \times S \to S$. Then $f_i \star g_i$ are the pieces of $f \star g$ over the partition P_i.

Note how the partition is entirely untouched. This "separation of concerns" is what enables us to separate the issues of domain decompositions for arithmetic issues of piecewise-defined functions (and expressions).

To simplify our presentation, we introduce a domain restriction operation and a join combinator on (partial) functions, to allow us a more syntactic method of "building up" piecewise functions. These are quite similar to Kahl's table composition combinators [2].

Definition 5. *The restriction f^A of a function f to a domain specified by a set A is*

$$f^A(x) ::= \begin{cases} f(x) & \text{if } x \in A \\ \bot & \text{otherwise} \end{cases}$$

Definition 6. *The join, $f \oplus g$, of two (partial) functions f and g, is defined as*

$$(f \oplus g)(x) ::= \begin{cases} f(x) & \text{if } f(x) \text{ is defined and } g(x) \text{ is undefined} \\ g(x) & \text{if } g(x) \text{ is defined and } f(x) \text{ is undefined} \\ \bot & \text{otherwise} \end{cases}$$

This allows us to rewrite a piecewise-defined function f in terms of its pieces as

$$f = f_1^{P_1} \oplus f_2^{P_2} \oplus \ldots \oplus f_n^{P_n}.$$

But our goal is to work with piecewise-defined functions where we have a *symbolic* partition. We need some new tools for this, which we will develop in the next two sections.

2.2 Hybrid Sets

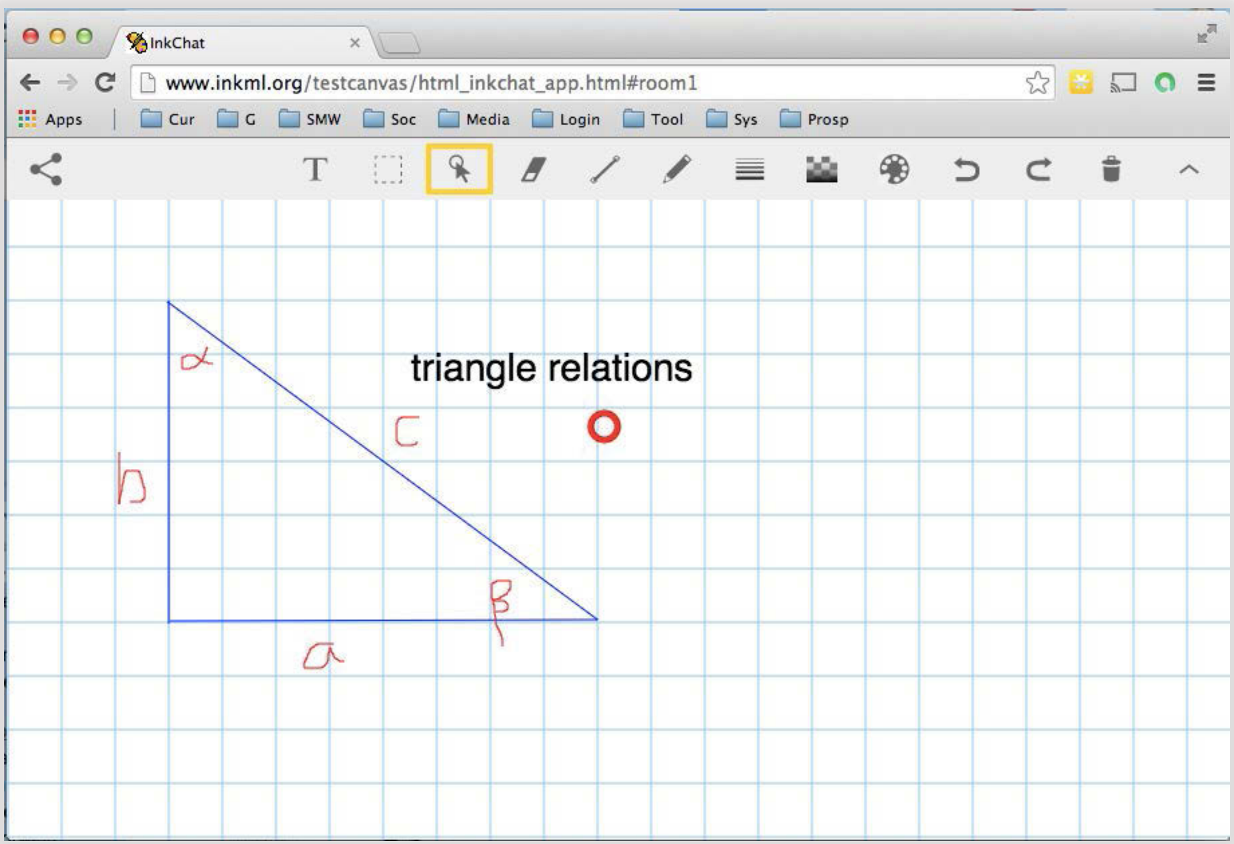
We consider an extension of multisets, in which elements can occur multiple times, to *hybrid sets*, where the multiplicity of an element in a hybrid set can range over all of $\mathbb{Z}$, instead of just $\mathbb{N}_0$. Thus a hybrid set, over an underlying set U, is a mapping $U \to \mathbb{Z}$, i.e., it is an element of $\mathbb{Z}^U$. We use the following definition, adapted from [4]:

Definition 7. *Given a universe U, any function $H : U \to \mathbb{Z}$ is called a* hybrid set.

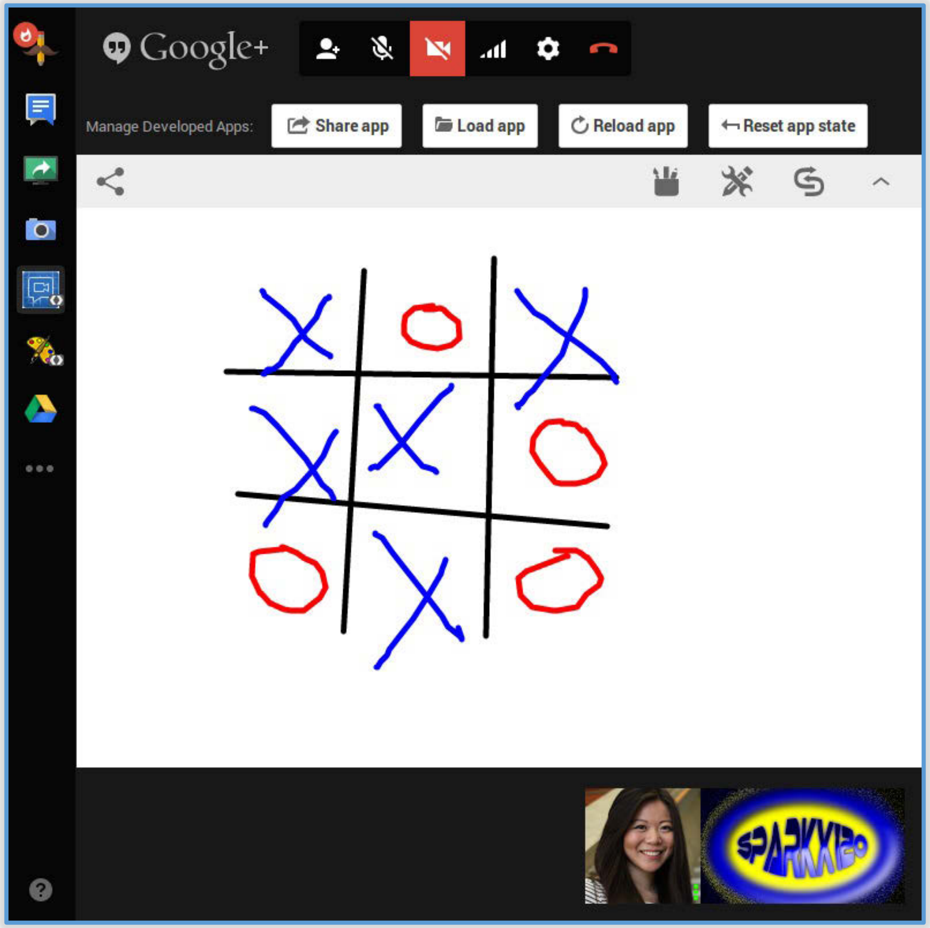
Notation 1 *Since the characteristic function of a set makes any set a hybrid set, we will implicitly lift all sets to hybrid sets whenever necessary.*

We can immediately define some useful vocabulary for working with hybrid sets.

Collaboration:
Document Annotation



Collaboration:
Google Hangout Embedding



$$\int_{\partial c} \omega = \int_c d\omega$$

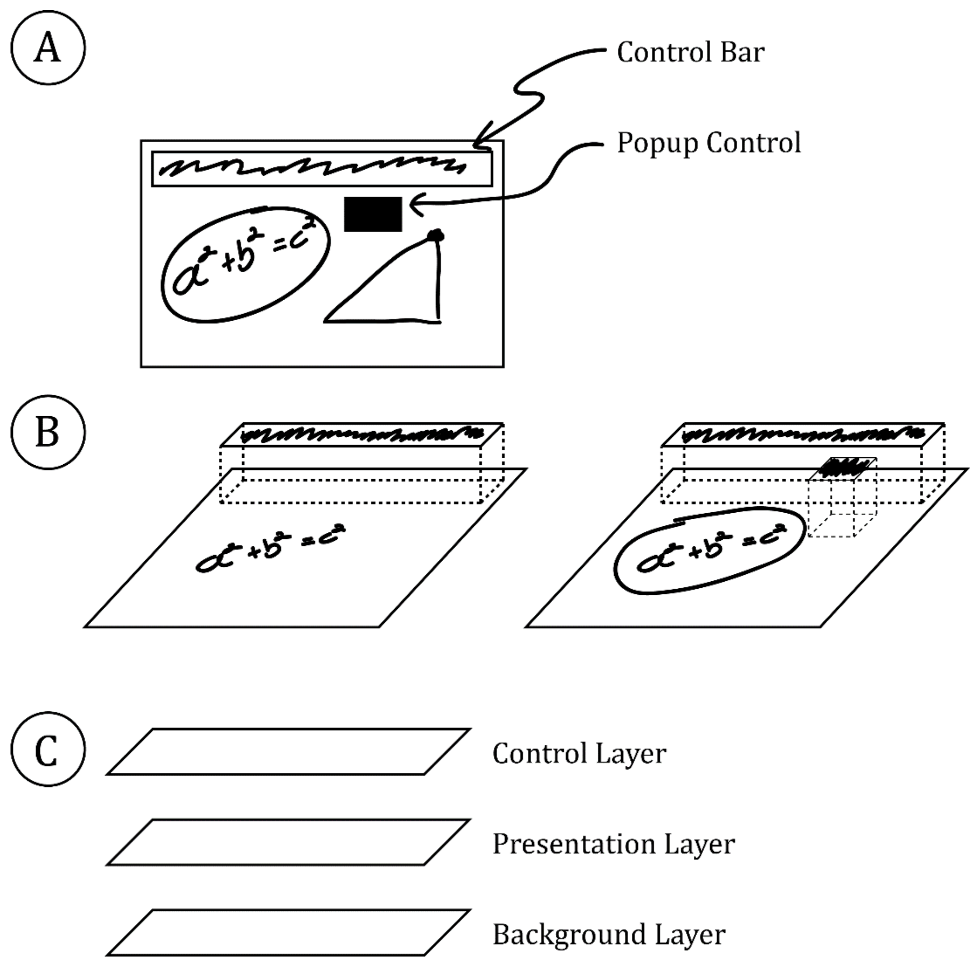
Cloud Integration

- Save or load files to cloud storage
  - DropBox
  - Google Keep
  - Others possible

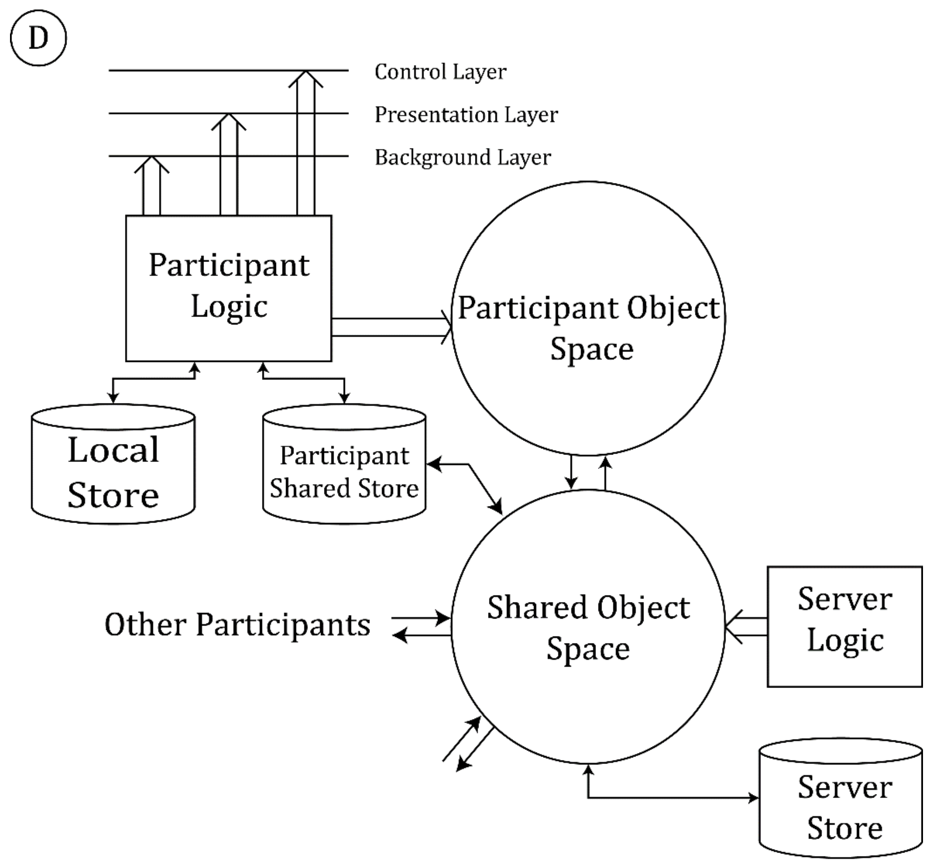
- Previous work to store user profiles
  - Save cloud of ground-truth labelled symbols (corrected/accepted)

- Future work to store user-defined brushes

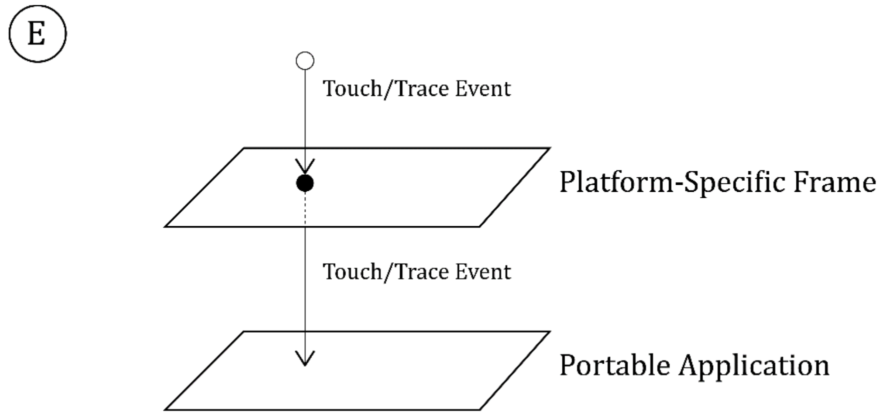
Architectural Direction

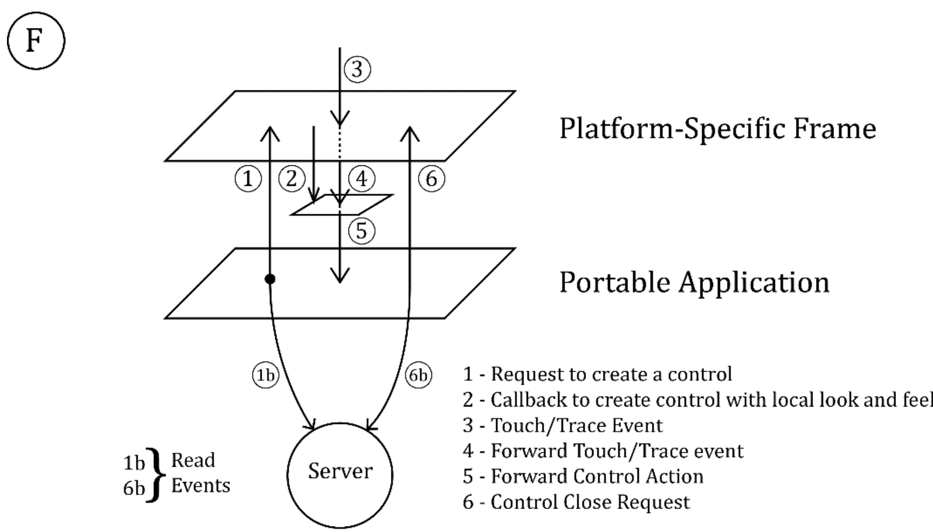


Architectural Direction

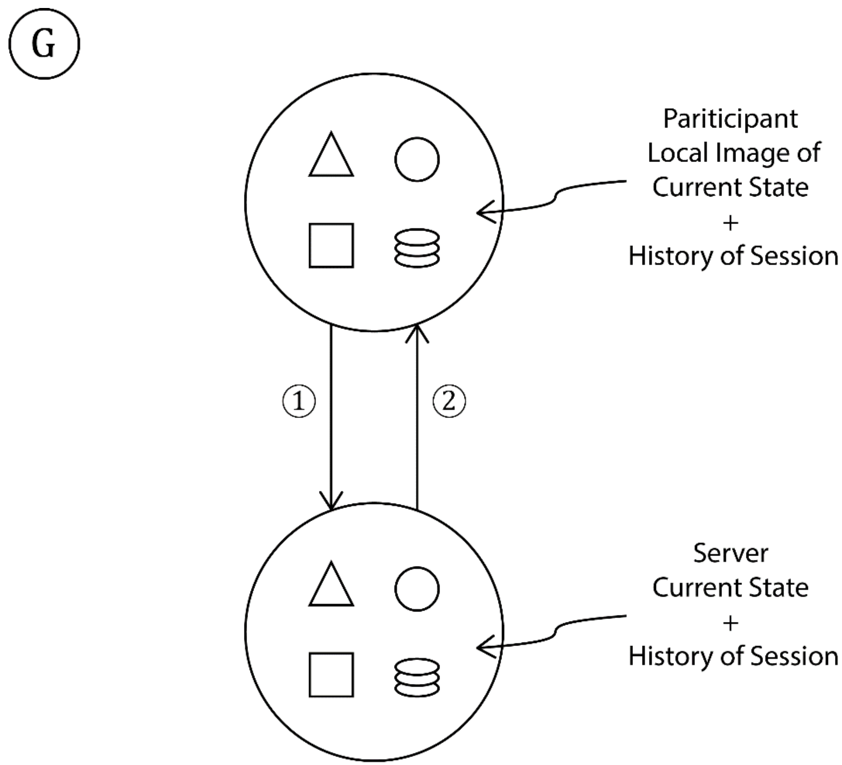


Architectural Direction





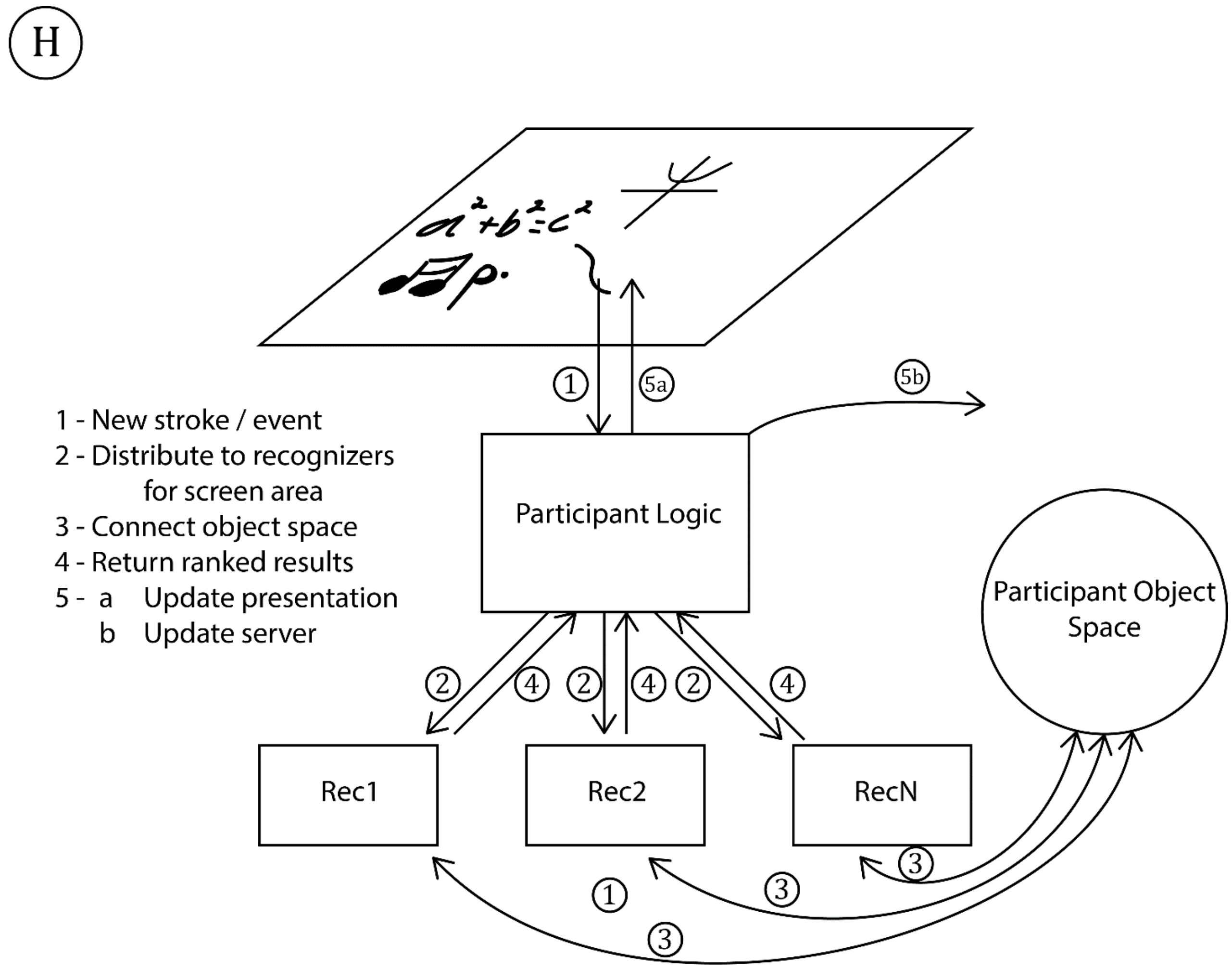
Architectural Direction



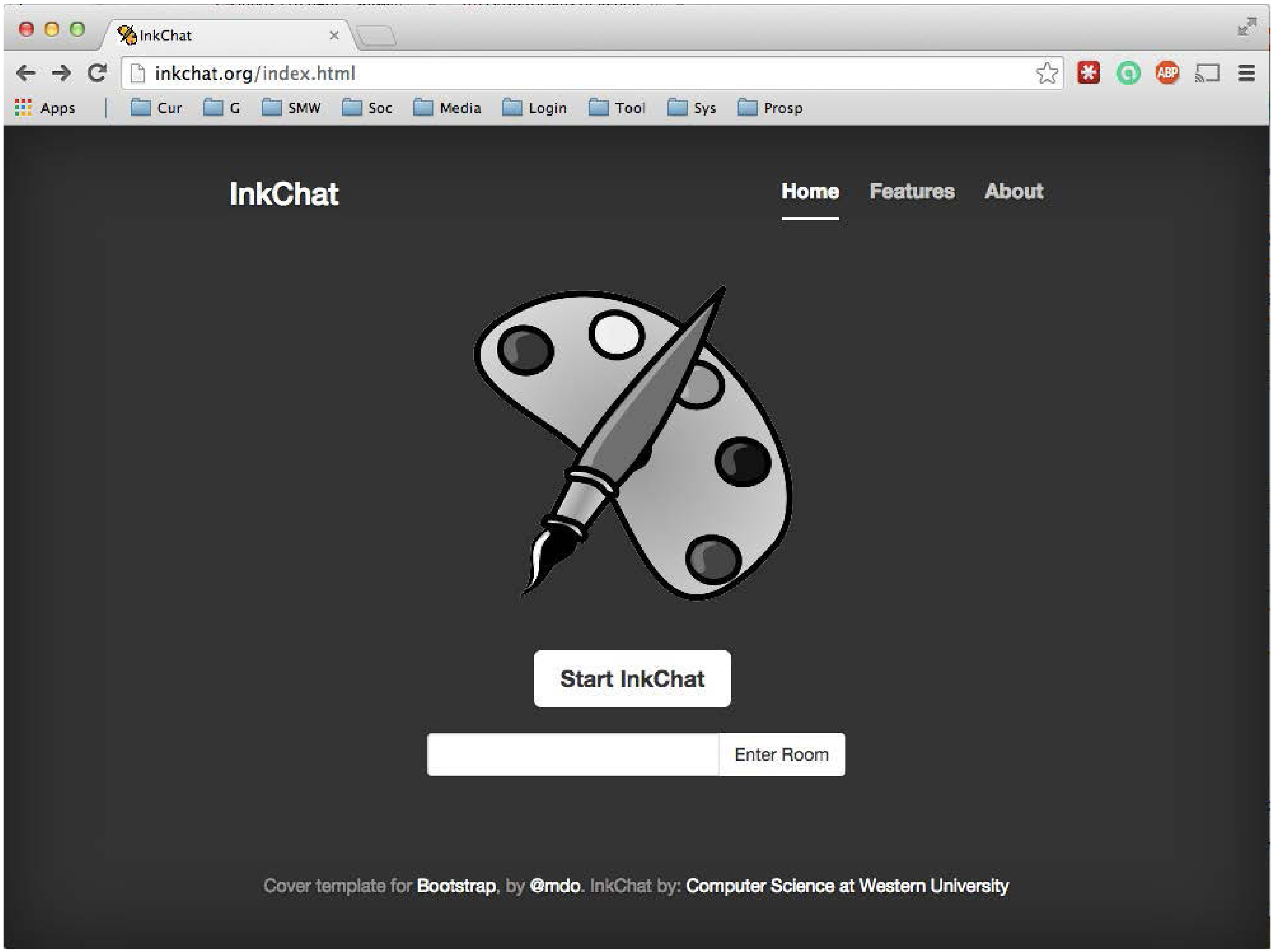
1 - Notification of Participant event (object creation / deletion / movement history navigation / page change / etc)

2 - Notification of Server Event (Passed on from another client or result of server configuration)

Architectural Direction



Application Web Site



Conclusions

- Technical collaboration requires tools not found in the business setting.
- Drawing, mathematics and scientific documents are in the work flow.
- The treachery of images.
- Needed:
  - Math handwriting recognition.
  - Easy geometry and diagrams.
  - Document mark up.
  - APIs to scientific software.
- Even a little goes a long way….
- … there is a lot of opportunity for future development.

Thanks

Bruce Char
Joseph Choi
Michael Friesen
Oleg Golubitsky
Rui Hu
Vadim Mazalov
Shirley Miao

Jeliasko Polihronov
Maya Ramamurthy
Elena Smirnova
Clare So
Stephen Solis
Coby Viner
James Wake

Maplesoft
Microsoft
MITACS
NSERC