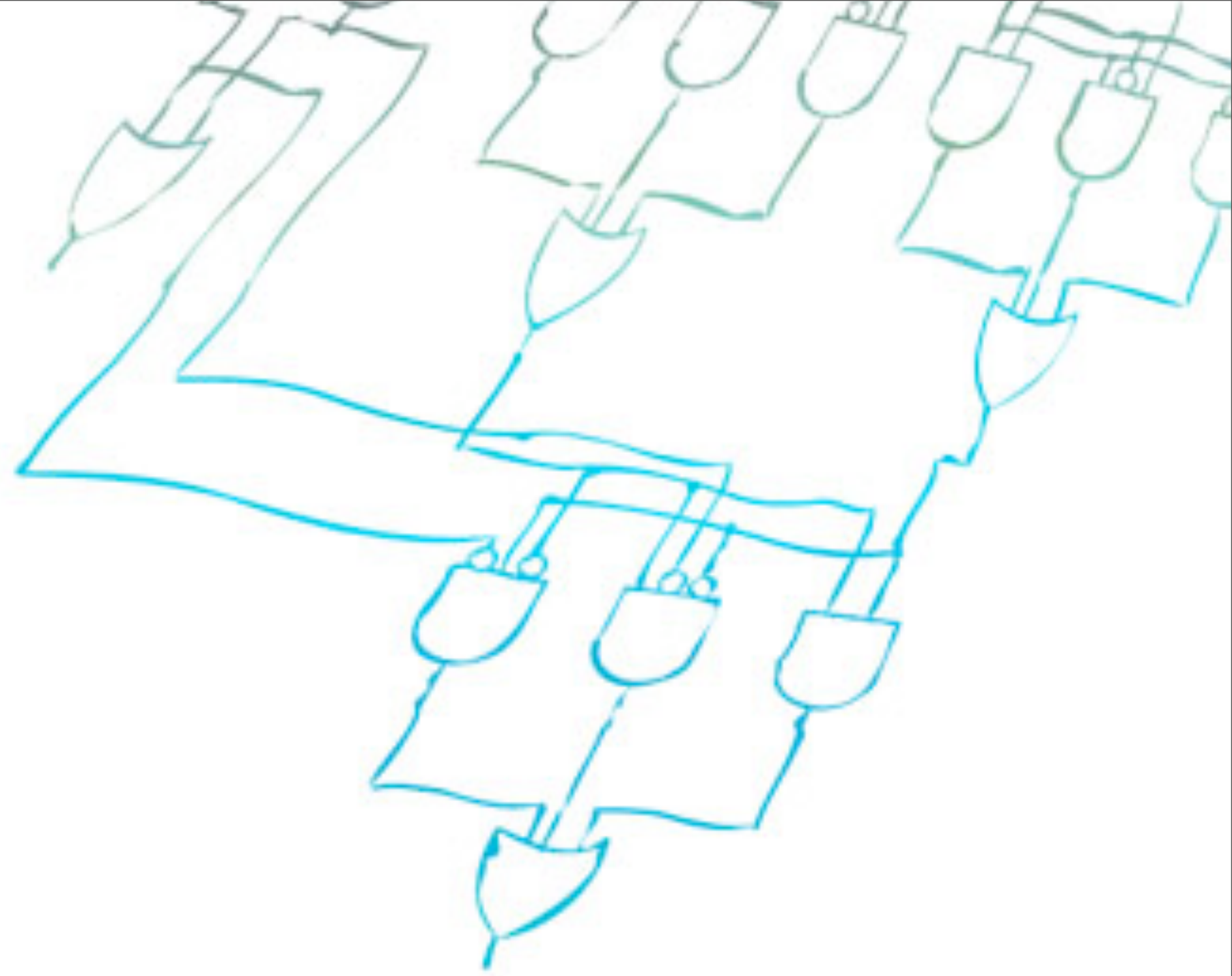
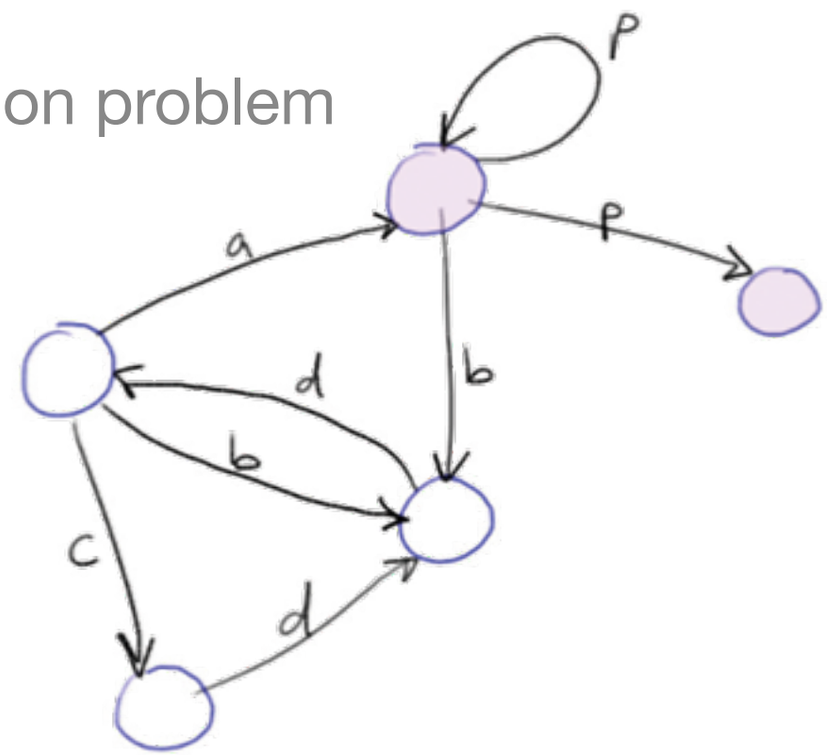


Whiteboards that Compute: Performance Challenges Ahead

Ryan Dixon & Timothy Sherwood
UC Santa Barbara

Whiteboard Computing

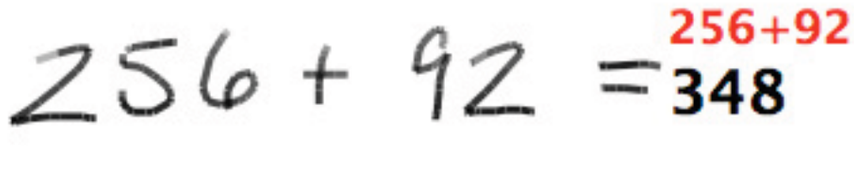
- A board that recognizes handwriting
- A board that recognizes mathematical equations
- A board that recognizes *<insert your favorite item here>*
- It’s more than just a recognition problem

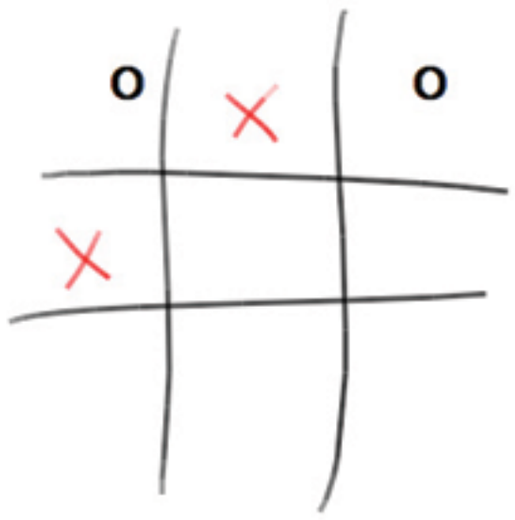


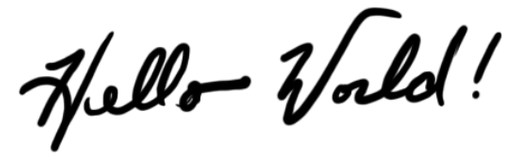
Whiteboard Computing *defined*

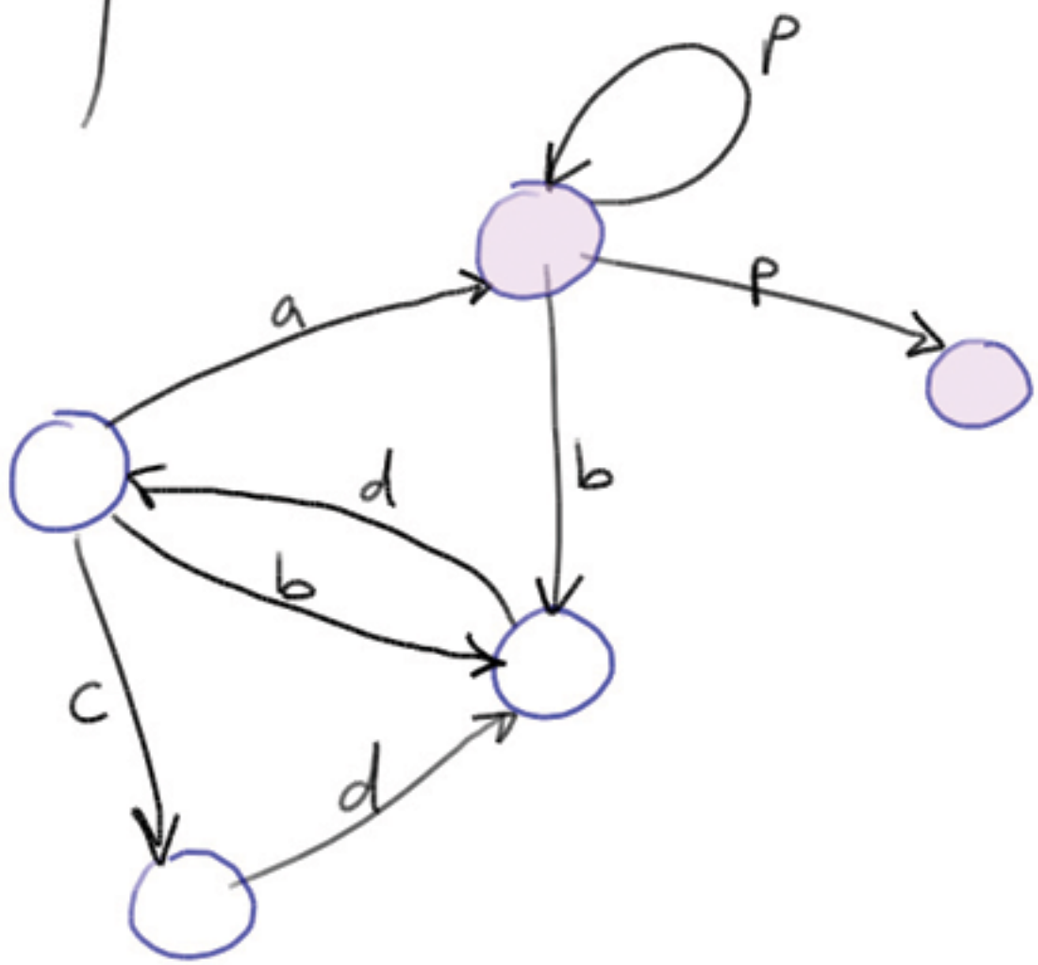
- A board that is capable of augmenting a user's drawings to enable a more efficient means of solving a given problem than would be possible using a traditional board and marker.

- A generalization of the spreadsheet.







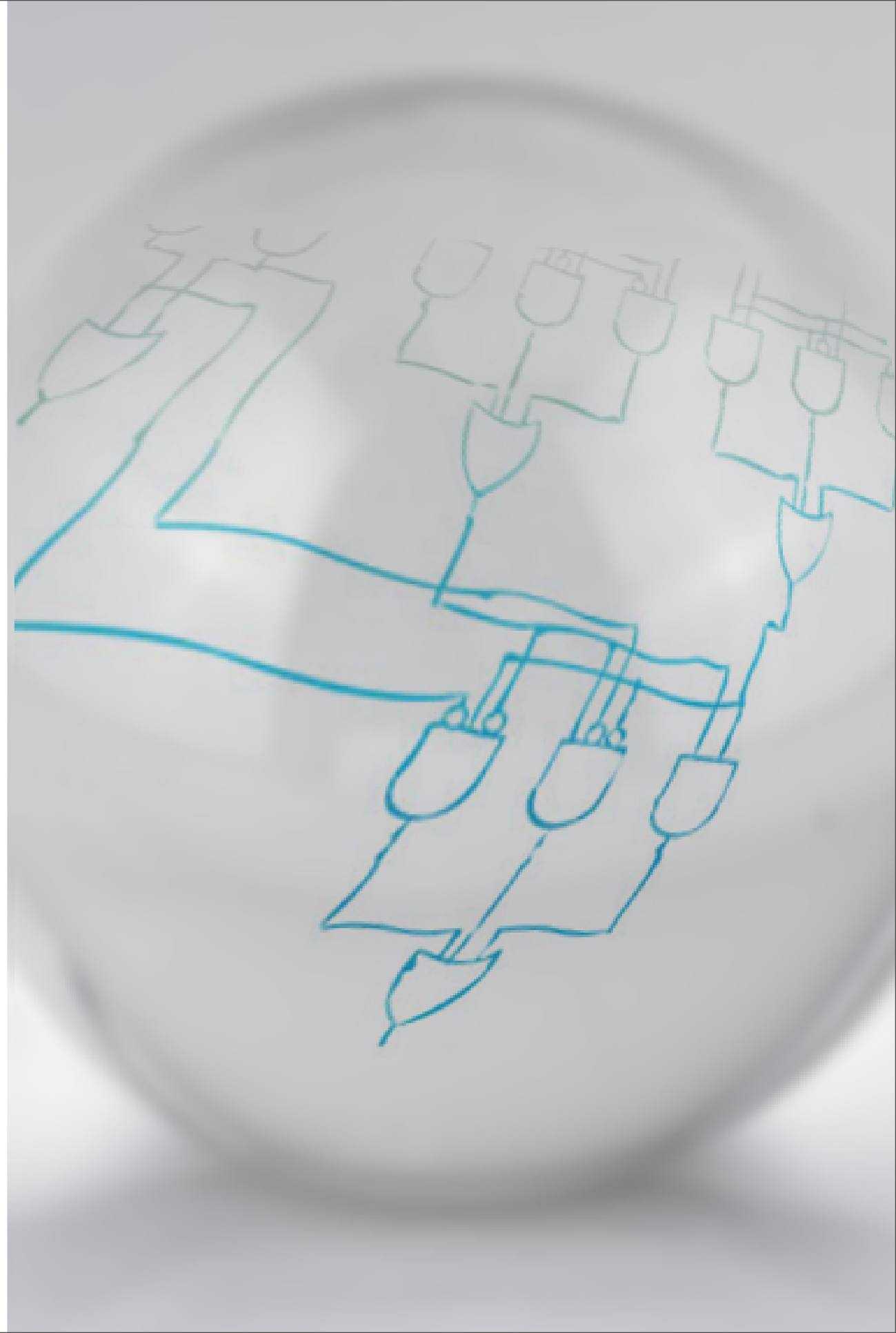


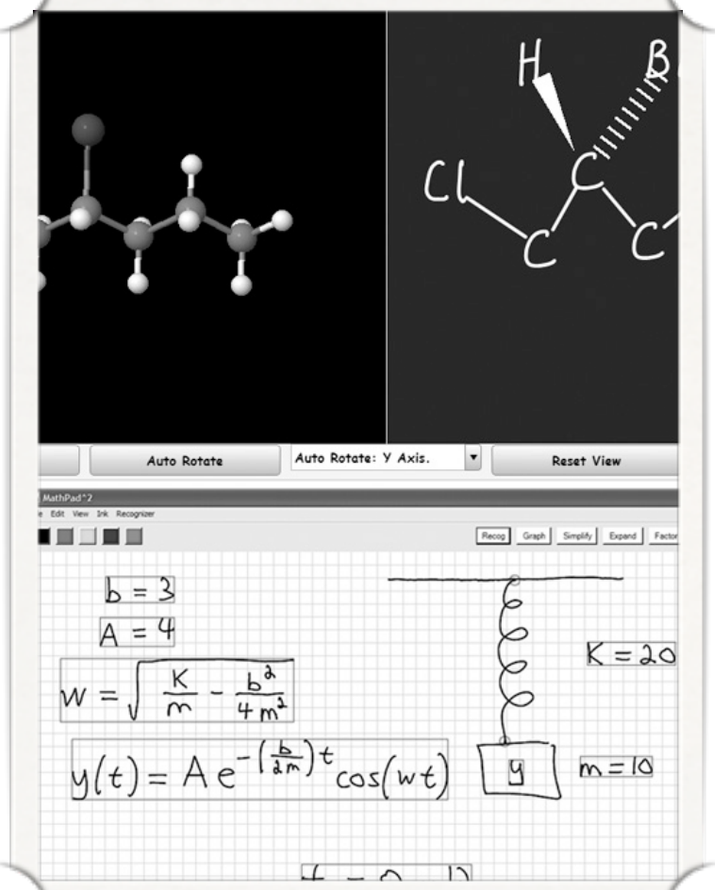
$$W = n\sqrt{\frac{K}{m} - \frac{b^2}{4m^2}}$$

app

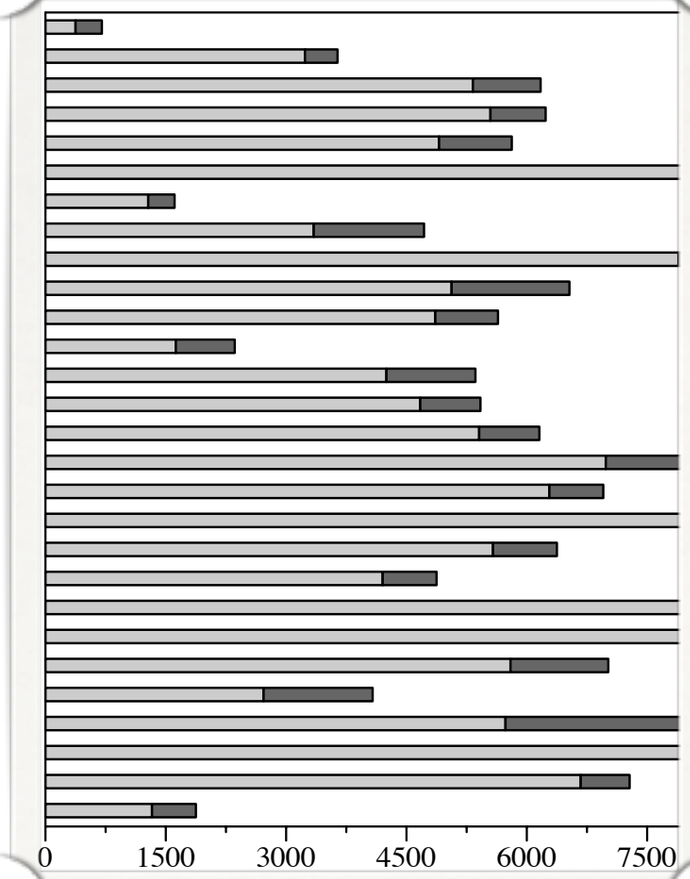
Emerging Workloads

- Shift to Multi-Core Architectures
- Architects look 5-15 years ahead
- We are looking ahead for future mass market workloads with potential to motivate the next generation design

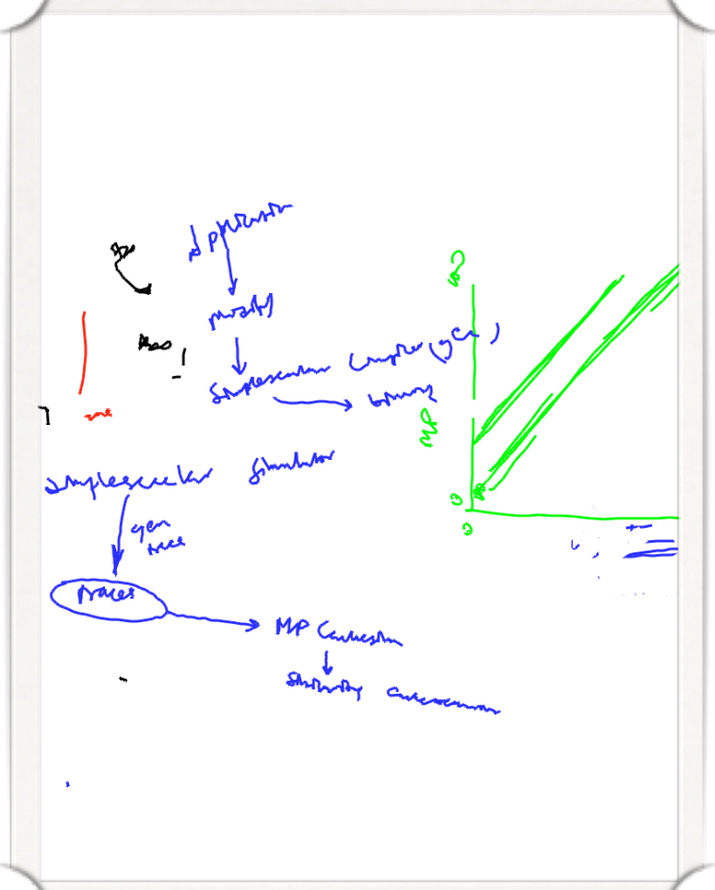




Sketch
Systems

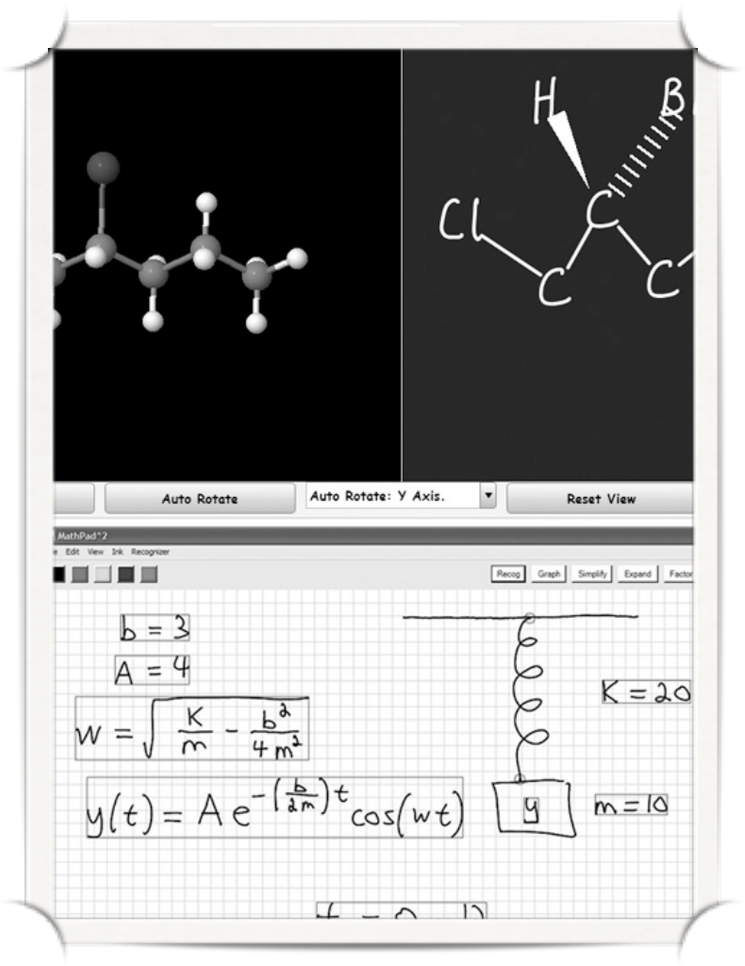


Application
Performance



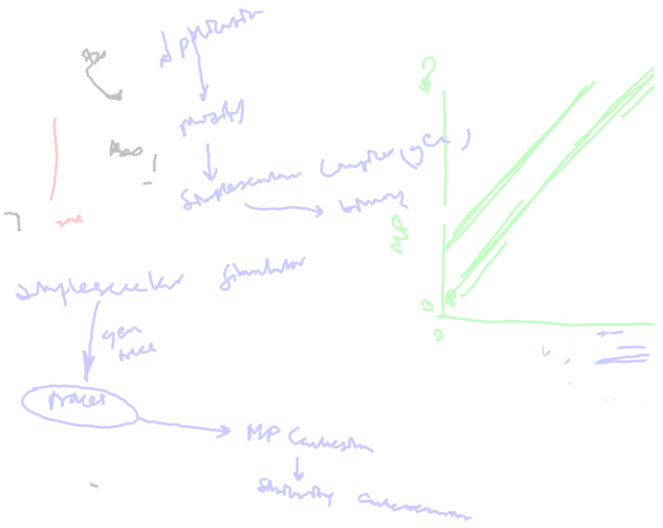
User
Performance

Three Topics for Today



Sketch Systems

Application Performance

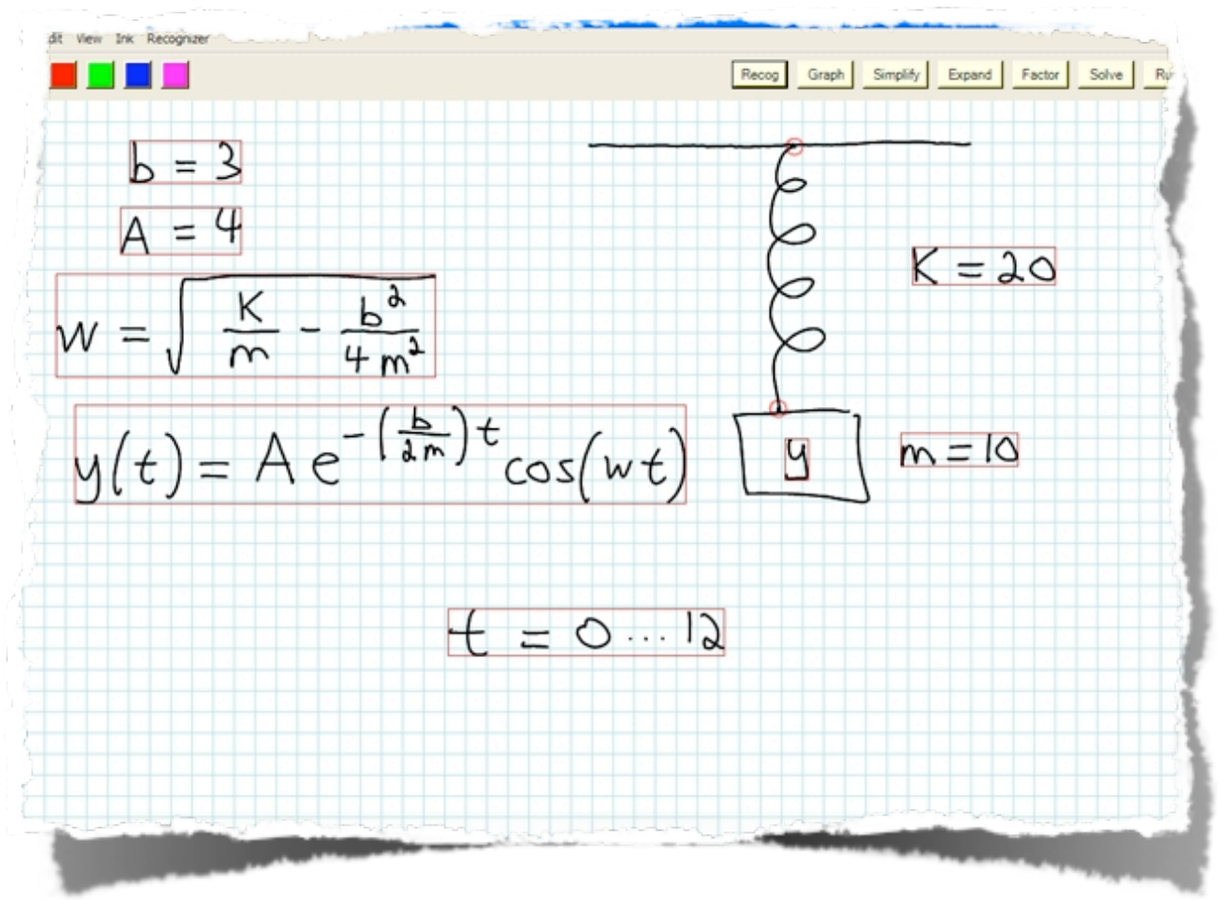


User Performance

Three Topics for Today

Free-form Recognition

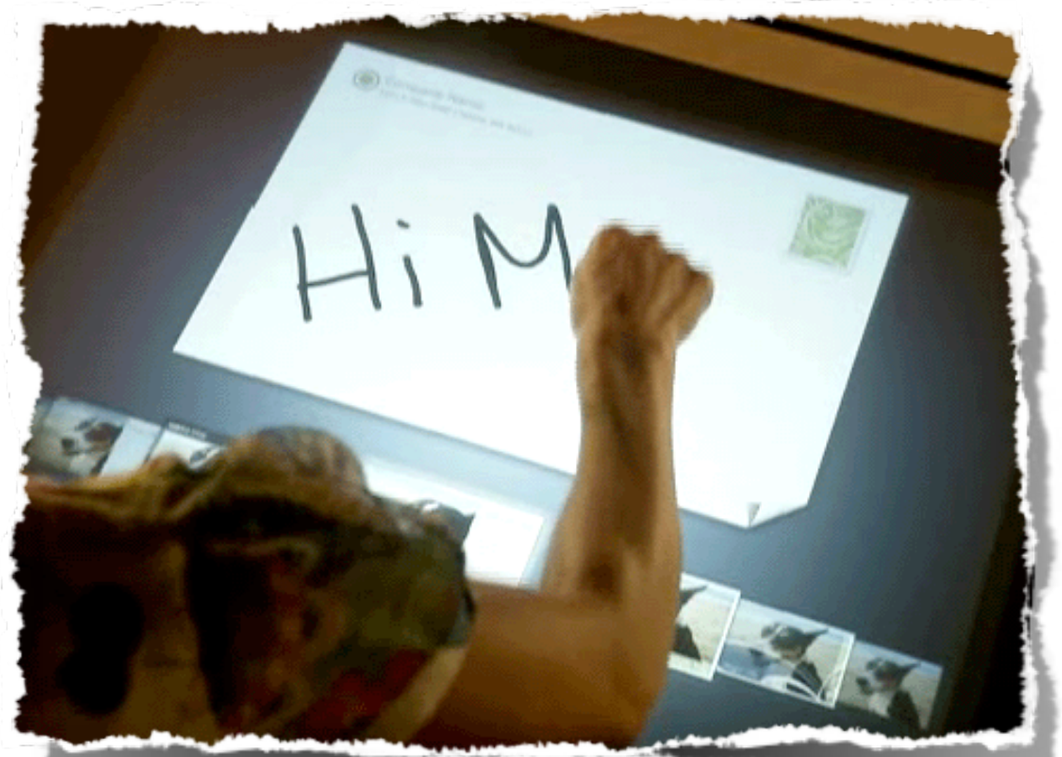
- **ASSIST** *Mechanical Systems*[1]
- **ChemPad** *Molecules*[2]
- **MathPad²** *Equations*[3]
- **SimuSketch** *Matlab Simulink*[4]



1. *Christine Alvarado. A Natural Sketching Environment: Bringing the Computer Into Early Stages of Mechanical Design. 2000.*
2. *D. Tenneson and S. Becker. Chempad: Generating 3D Molecules From 2D Sketches. 2005.*
3. *Joseph Laviola and Robert Zeleznik. Mathpad²: A System for the Creation and Exploration of Mathematical Sketches. 2004.*
4. *Levent Kara and Thomas Stahovich. Hierarchical Parsing and Recognition of Hand-Sketched Diagrams. 2004.*

Commercial Products

LiveBoard



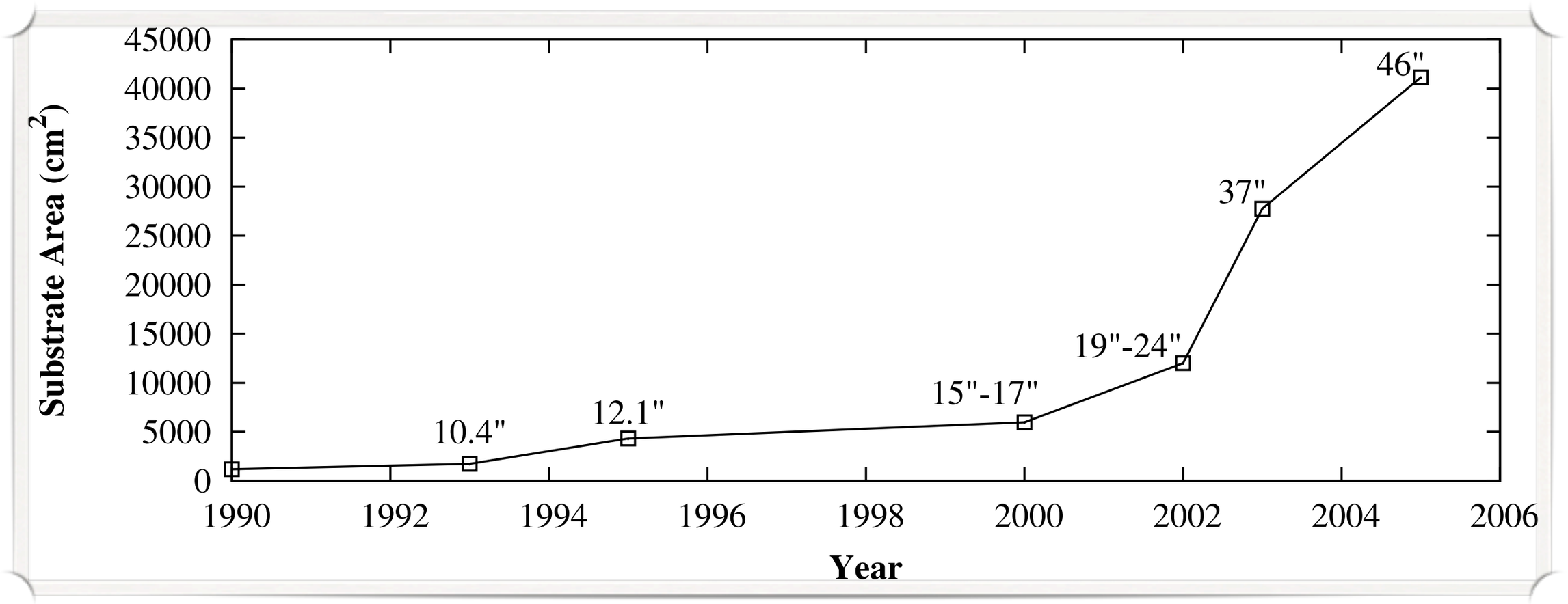
Microsoft Surface

eBeam

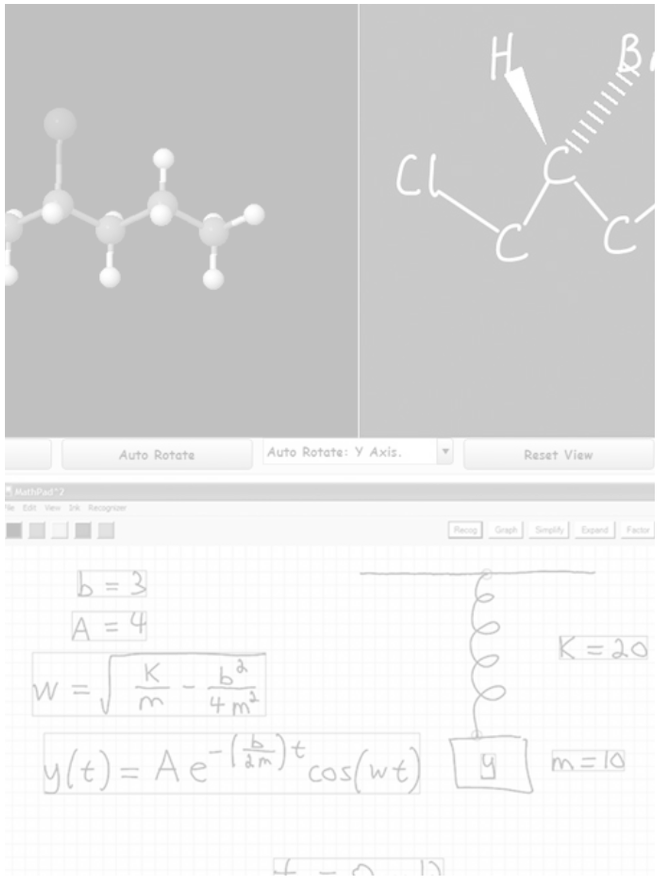
Tablet PC

Why Now?

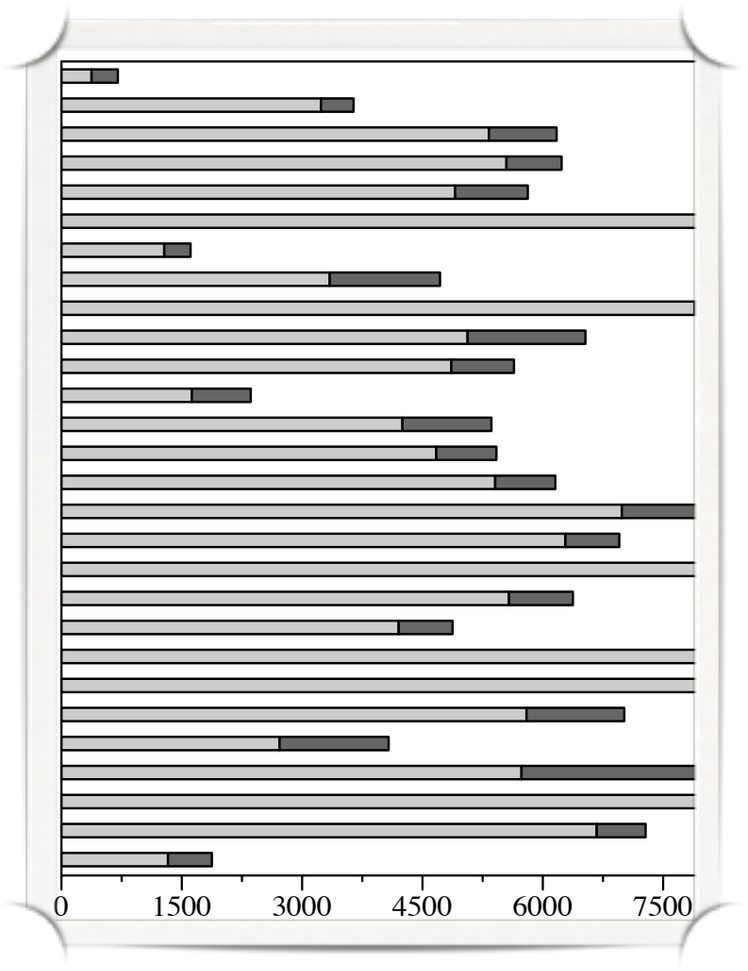
- Larger Surfaces Are Affordable[1]
- Surface Computing
- Input Technology Available
- One-off Recognition Projects



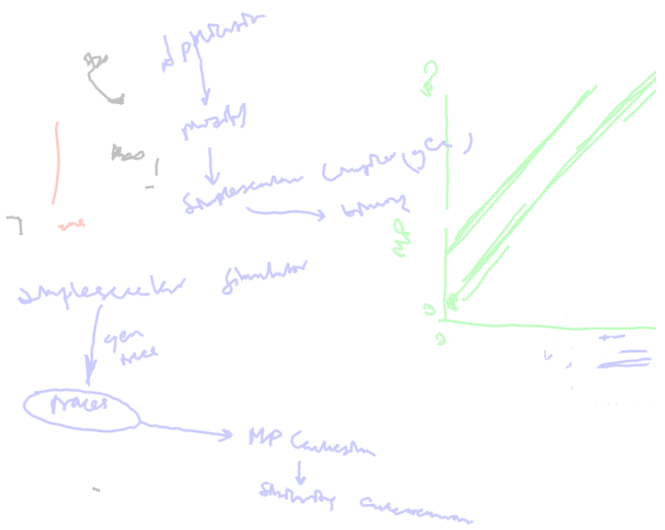
1. AKT Large Area PECVD Capability. 2005.



Sketch
Systems

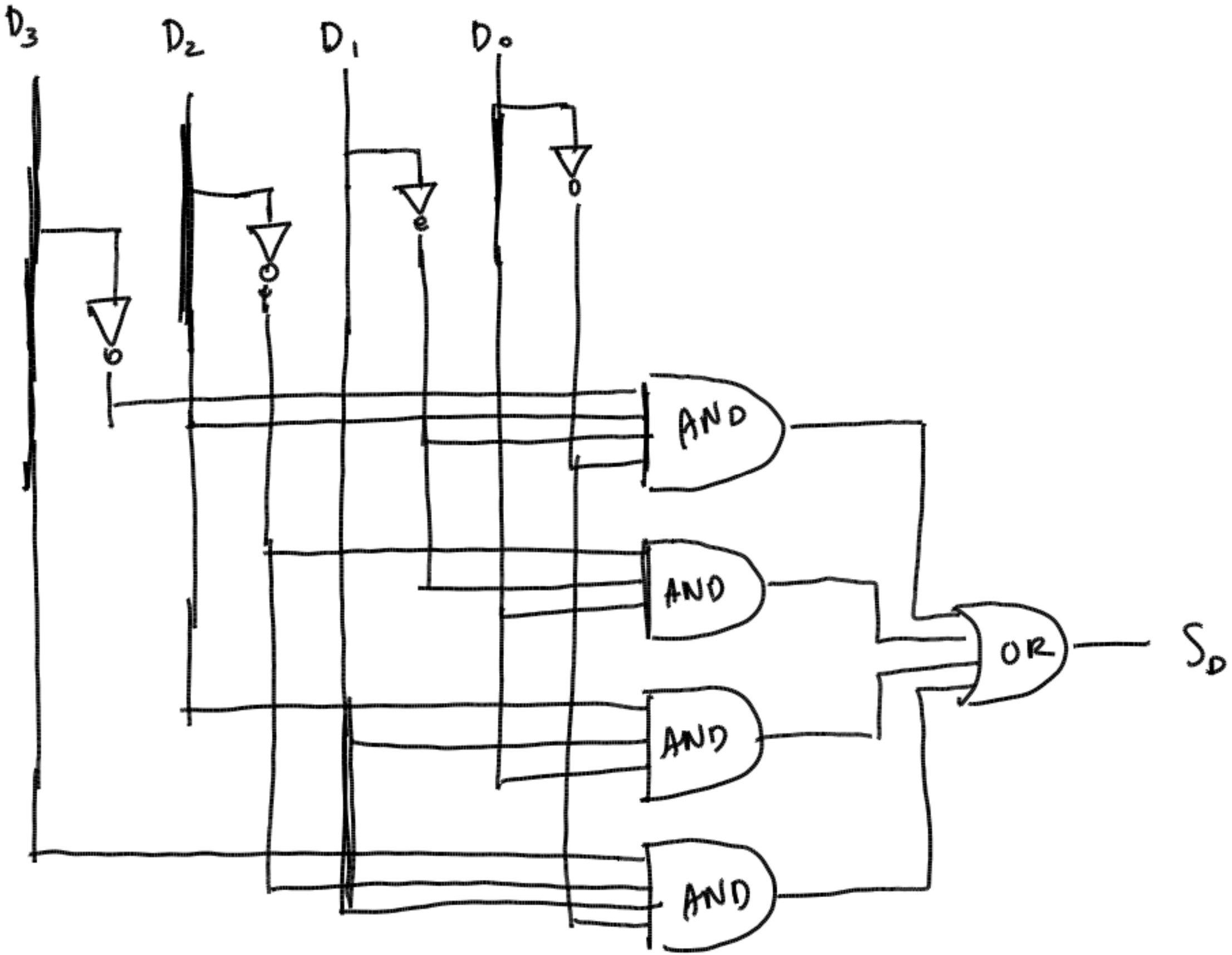


Application
Performance



User
Performance

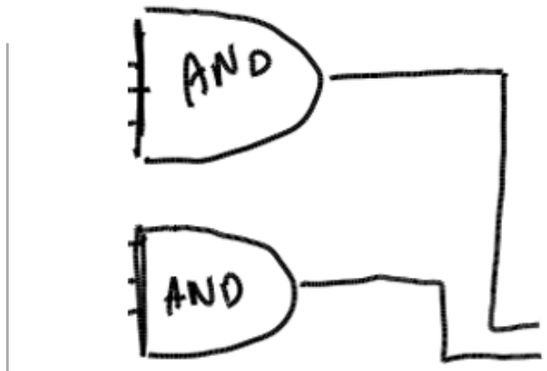
Three Topics for Today

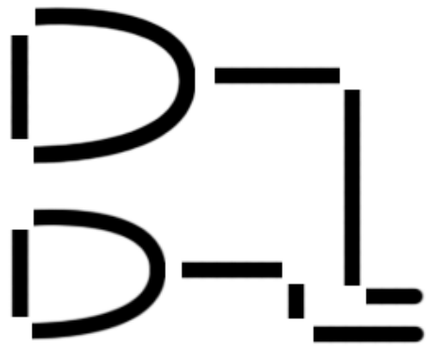


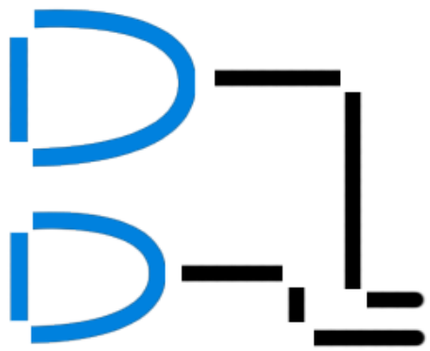
Circuit Recognition and TestRig

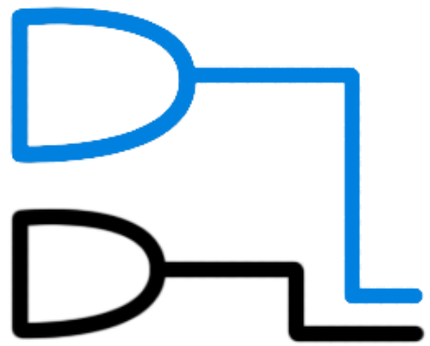
Recognition Pipeline

- **Segment**
  - Find atomic strokes composing the sketch
  - Sezgin[1] Corner Detection Algorithm
- **Label**
  - Classify each atomic stroke
  - Conditional Random Fields[2] | Loopy BP
- **Group**
  - Group strokes by semantics
  - Support Vector Machines[3]

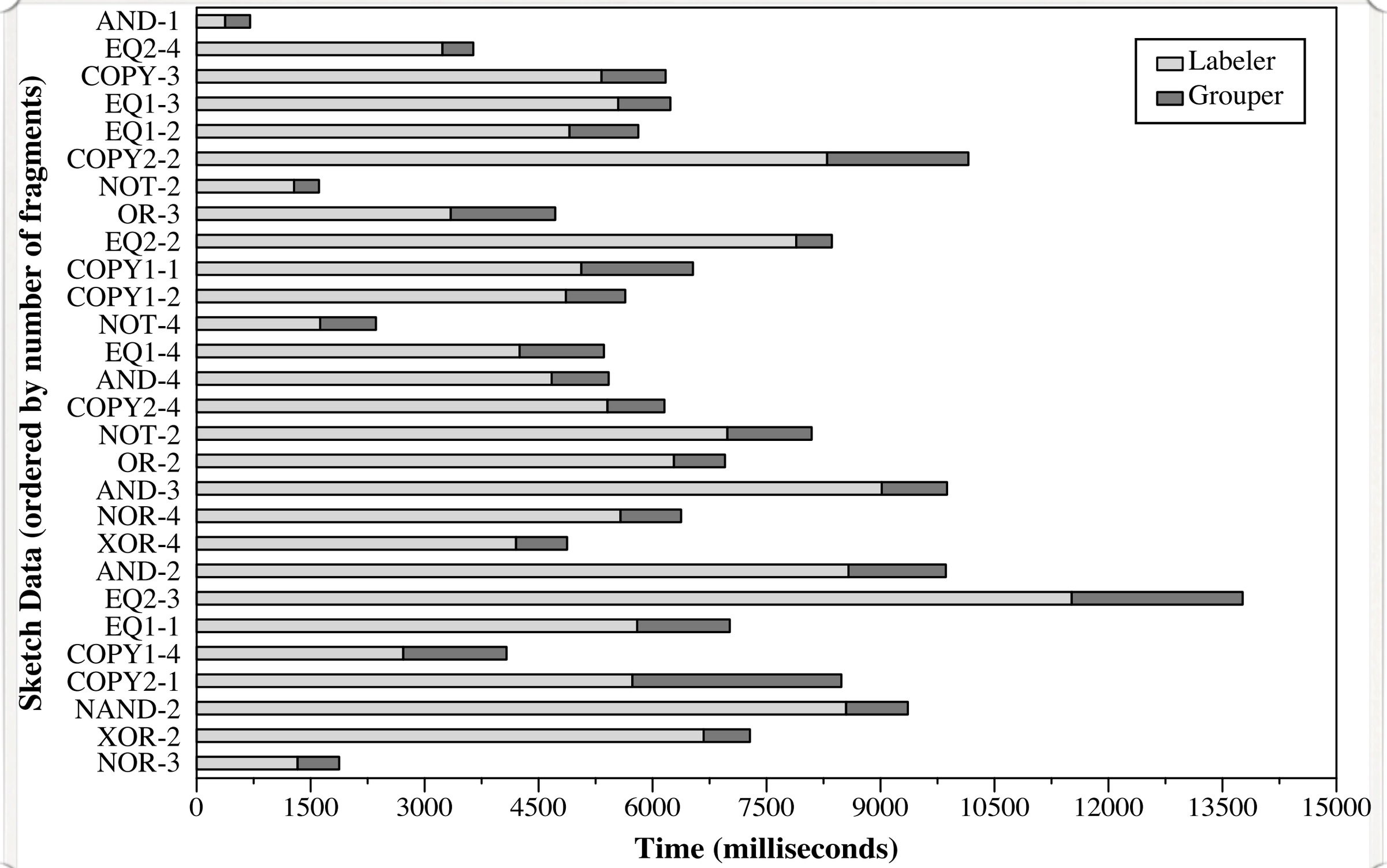






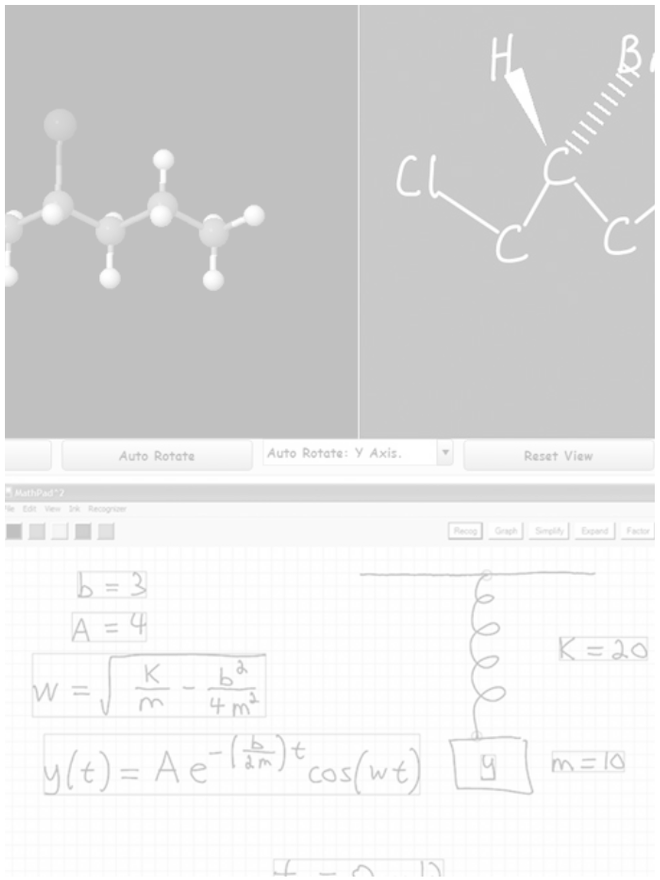


1. *Tevfik Metin Sezgin and Randall Davis. Scale-space Based Feature Point Detection for Digital Ink. 2004.*
2. *M.Szummer and Y. Qi. Contextual recognition of hand-drawn diagrams with conditional random fields. 2004.*
3. *Christine Alvarado. Sketch recognition for digital circuit design in the classroom. 2007*



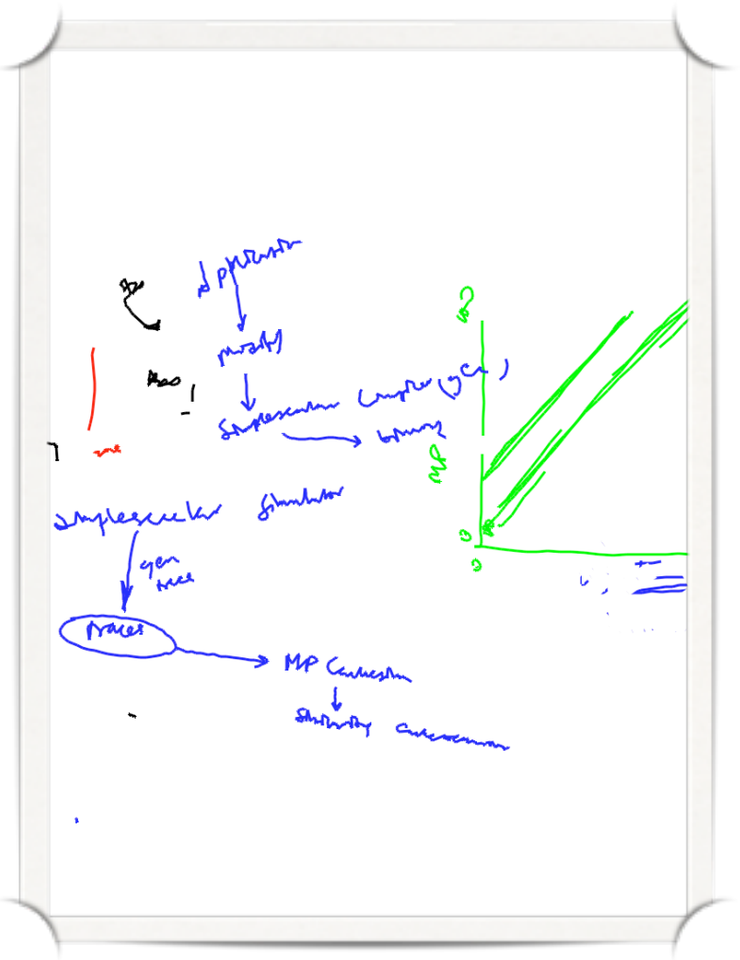
TestRig Performance

the bottlenecks



Sketch
Systems

Application
Performance

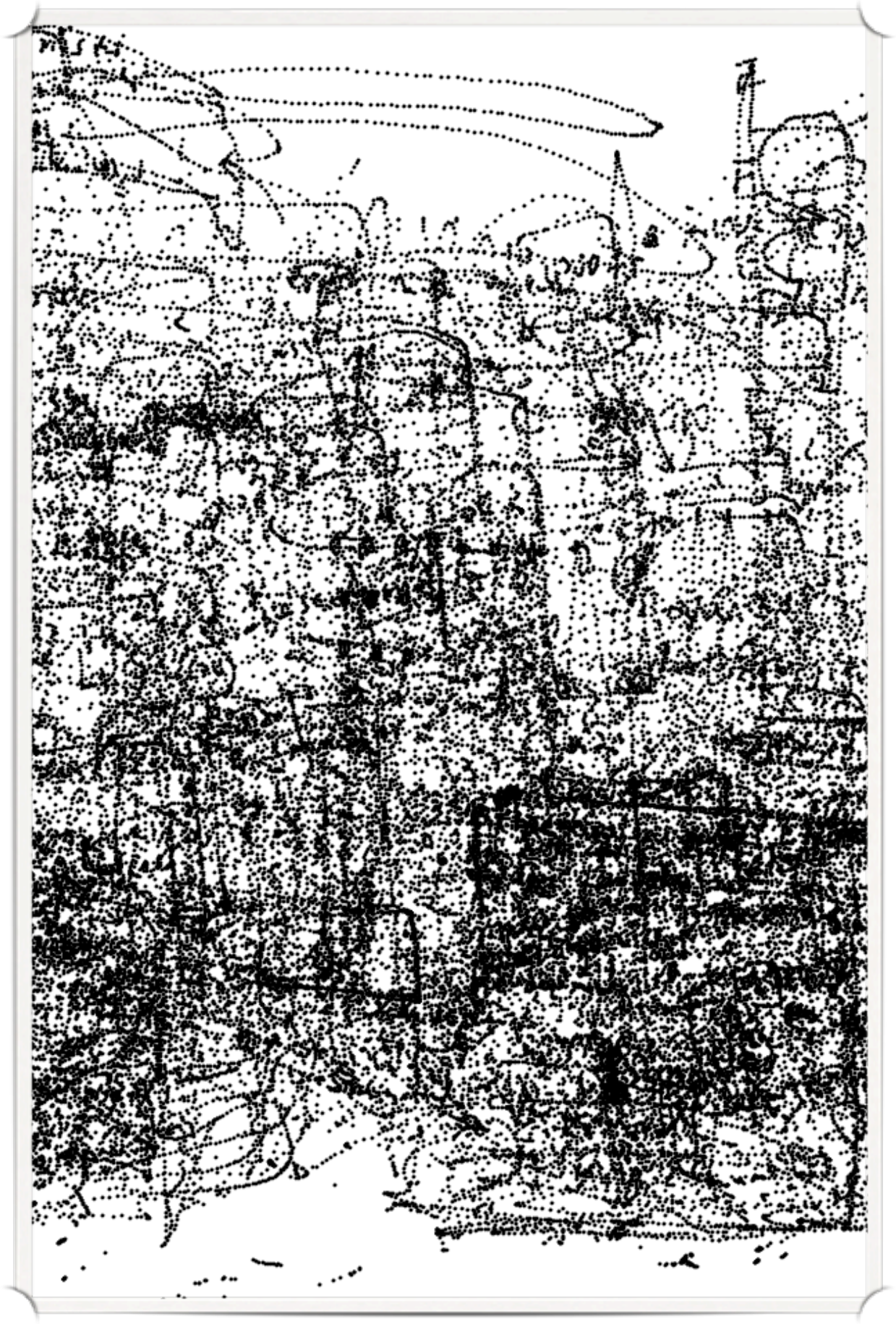


User
Performance

Three Topics for Today

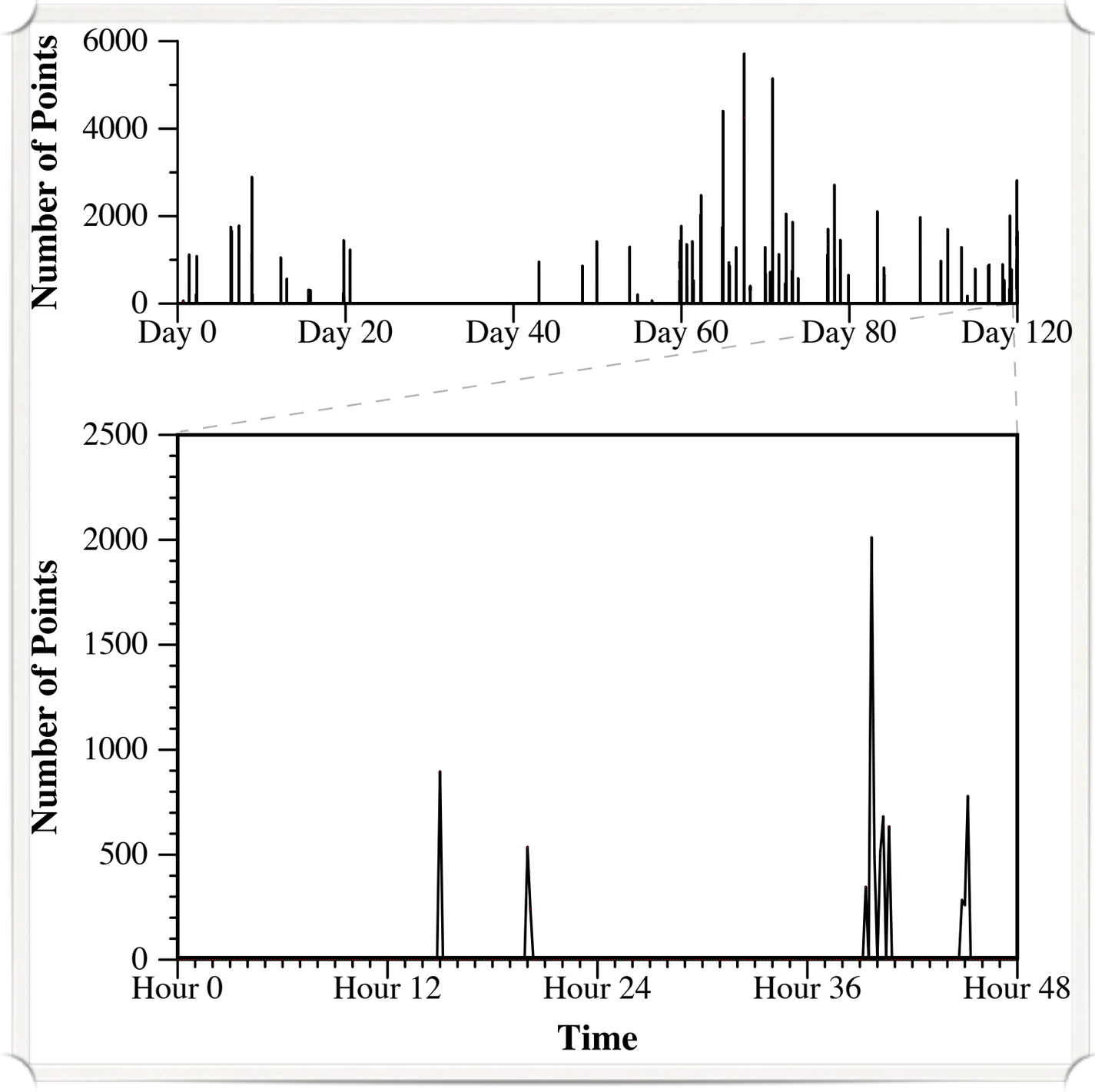
User Characterization

- Goal: understand our lab's board activity
- eBeam capture device
- Four months of data collection
- 300,000 points generated
- Timing, color, erasure, and stroke segmentation data



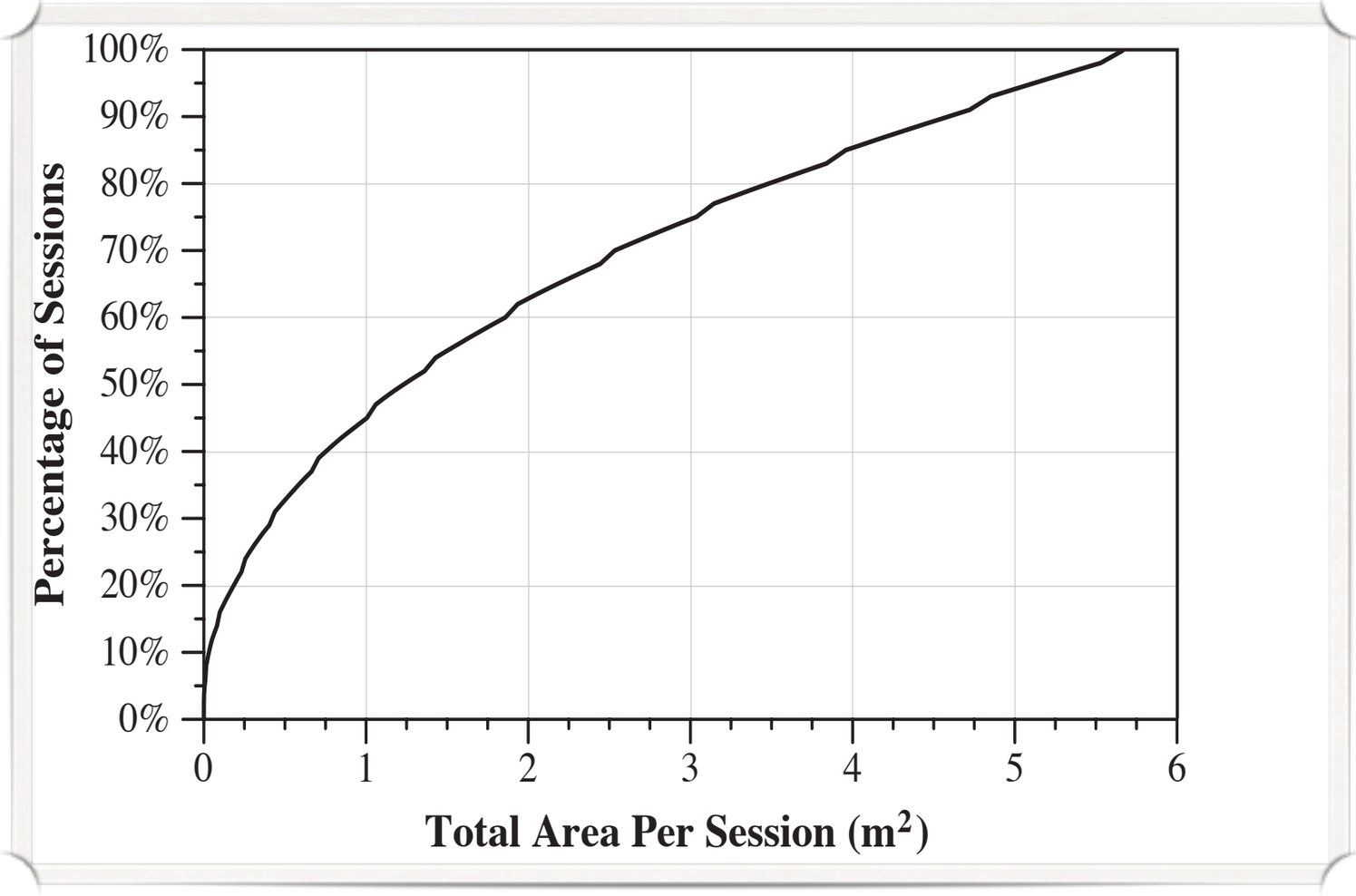
Stroke Data

- Board traffic is bursty
- Sessions clearly defined by ranges of activity and inactivity
- Approximately half of all sessions contained fewer than 1,500 strokes



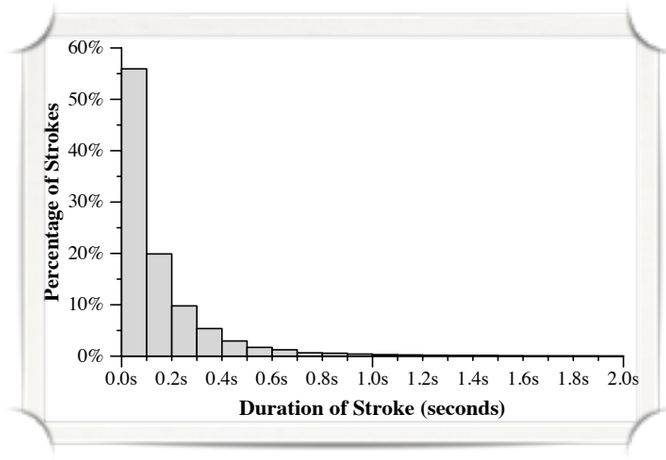
Session Size

- Sessions based on timing data can be used for many measurements
- Roughly half of the sessions required only 1 m^2 of the whiteboard area.
- In rare cases, the entire board was required during a single session



Other data covered in the paper…

- Points Generated per Day
- Number of Strokes per Session
- Percentage of Erasure Strokes
- Duration of Strokes
- Inter-Stroke Timing
- Number of Corners per Stroke



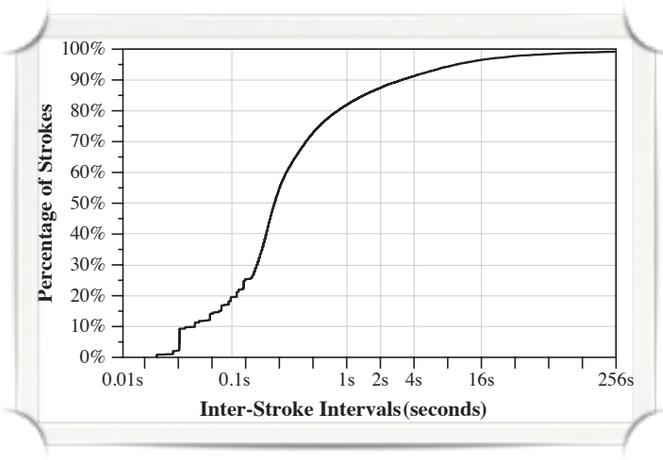
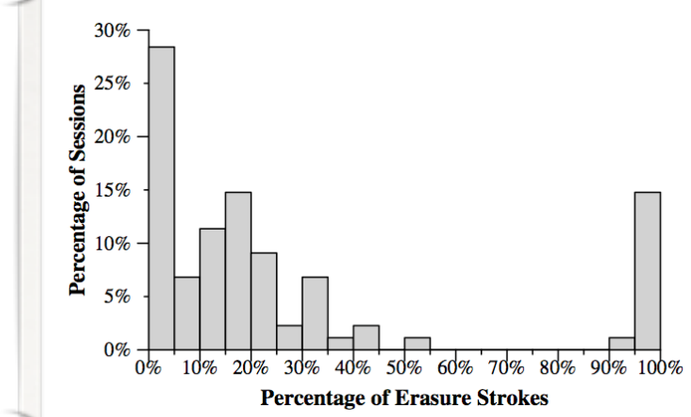


TABLE I
NUMBER OF CORNERS PRESENT IN FRAGMENTED DRAWING STROKES
STROKE FRAGMENTATION AND CORNER DETECTION PERFORMED USING SPEED CALCULATIONS AS PRESENTED IN [18]

# Corners	0	1	2+
# Segments (%)	15291 (93.9%)	579 (3.6%)	407 (2.5%)

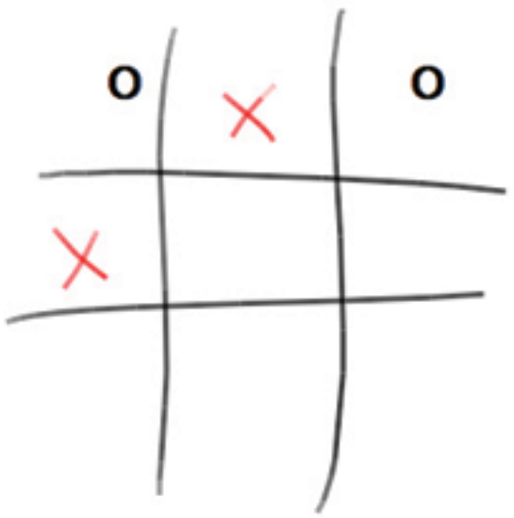


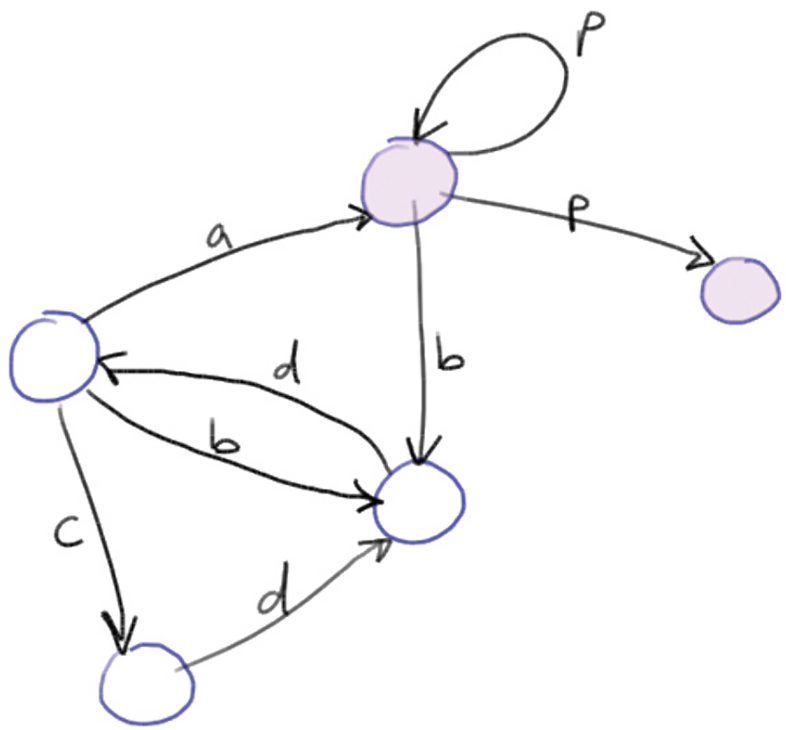
Conclusion

- This is preliminary work! No existing applications exhibit the full workload were are interested in analyzing.
- It is important that we continue to investigate future workloads that have the potential to drive our industry forward.
- Technology is ripe for solving the whiteboard computing problem.
  - There are many one-off projects that handle free-form sketching.
  - Surface computing and similar projects are adding momentum
- A workload analysis of one state-of-the-art sketch recognition system
- A characterization of one typical whiteboard usage scenario

Future: Application Framework

- Many, small concurrent applications
- Communication via stroke metadata
- Applications build on top of existing sketch applications





app

256 + 92 = 348 (256+92)

Q | A