Technical Poster Design

COMP 400

Poster Session Challenges

Challenges

- Audiences make decisions quickly
- Posters sometimes stand alone
- Audiences come and go as presenters talk

Solutions

- Poster must be **accessible**
  - Show overall organization
  - Show comparisons
- Poster must be **comprehensible**
- Speaker must **attract** others, **adapt** to situation

Similar to a Technical Paper

Tell an interesting story

What's your "news"?

- What problem are you solving?
- What are your results/conclusions?
- What sets your work apart?
  - E.g., new algorithm or theoretical approach
- Why does your work matter?
- How can your work be applied?

Similar to a Technical Paper

Structure your story

• Say what you're going to say	Abstract
• Say it	Body & Results
• Say what you said	Conclusion

Different From a Technical Paper

Space is at a premium

Interactivity is key

- Be concise – word choice, sentence fragments
- Be precise – word choice
- Pictures are often more effective than words
- Omit unnecessary details
  - **MUST KNOW** Use as your main focus
  - **Good to know** Add some
  - Nice to know Leave details for oral presentation

Textual Presentation Guides the Reader

- Scale expresses relative importance
- Indenting shows subordination
  - As in this example
- Color adds emphasis or coherence
- *Meaningless* **font** and color ***changes*** are ***distracting***
- Avoid low-contrast colors

- White space directs gaze

Font Style and Size

- Title – about 4-8 words
  - 90 – 120 pt
- Headings – about 3 words
  - 36 – 48 pt
- Text
  - 30 – 36 pt

Sans serif fonts best in large scale – posters

Serif fonts best on small scale – papers

Controlled Morphing Using Mass Distributions

Tao Ju (jutao@rice.edu), Ron Goldman (rng@rice.edu)

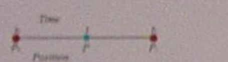
Morphing

Morphing transforms one target shape into another through transitions represented by averaging the target shapes.

Averaging Schemes

Linear Averaging

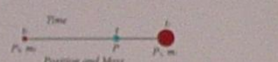
1. Taking the geometric center
2. Invariable speed of morphing



$P = (t_1 - t)P_0 + (t - t_0)P_1$

Weighted Averaging

1. Taking the center of masses
2. Controllable speed of morphing (Greater affinity for bigger mass)

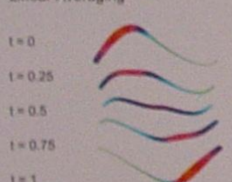


$P = ((t_1 - t)m_0P_0 + (t - t_0)m_1P_1)/((t_1 - t)m_0 + (t - t_0)m_1)$

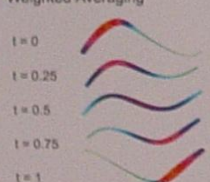
Rational B-spline Curves

A rational B-spline curve are defined by a series of control points with masses (weights). These masses are distributed along the curve so that each point on the curve has its own mass.

Linear Averaging



Weighted Averaging



Linear averaging produces wriggles in the middle of the morph, while weighted averaging sovles the problem by varying the morph speed along the curve with mass distribution.

Rational B-spline Surfaces

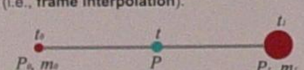
Rational B-spline surfaces also consist of points with masses. The following morphing sequence depicts the different between linear averaging and weighted averaging.

Linear Averaging: produces wriggles in the middle of the morph.

Weighted Averaging: generates smooth transition between targets.

Mass Assignment

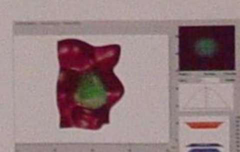
By varying mass distribution on the targets, we get different morphs. We can compute the appropriate masses so that the morph passes through a given point at a given time (i.e., **frame interpolation**).

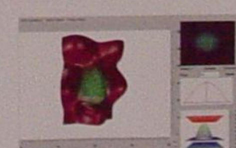


$m_0 = (P_1 - P)/(t_1 - t)$ $m_1 = (P - P_0)/(t - t_0)$

1. Robust interpolation. For any P and t, the morph always stays inbetween the two targets.
2. Smooth interpolation. The morphing function is infinitely differentiable at the interpolating point.

Editing Interface:



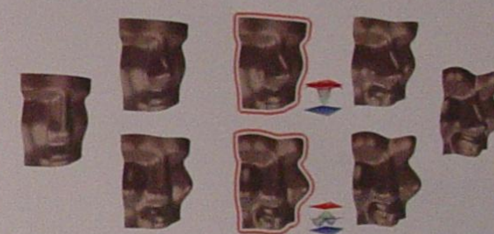


Surface Examples

Here is an example where two face models are morphed with different mass distributions.

Uniform mass distribution (linear averaging)

Non-uniform mass distribution (by frame interpolation)



Extensions

1. Morphing through multiple frames at given times. By computing mass distributions on each frame, a piece-wise morph can be constructed by weighted averaging in which the speed of the morph is continuous.

2. Morphing among multiple targets. Using barycentric coordinates, the morph can be controlled similarly by interpolating an intermediate frame.

Conclusion

Treat rational B-spline curves/surfaces as colletions of points with masses.

Use weighted averaging instead of linear averaging to take point masses into consideration.

Customize the morph by assigning different masses to different parts of the curves/surfaces.

feedtree Distributing Web news with peer-to-peer multicast

Dan Sandler, Alan Mislove, Ansley Post, Peter Druschel
Rice University

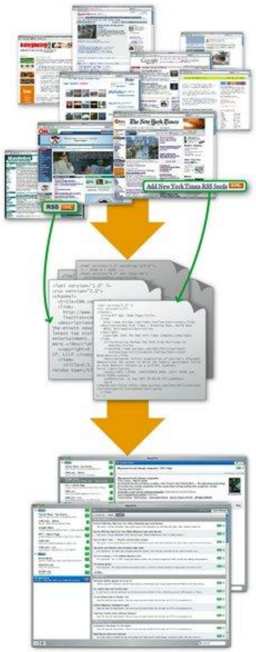
Background:

How RSS saved the Web.

The Web has experienced an explosion of MICRONEWS: highly focused chunks of content, published frequently and irregularly, scattered across scores of sites. Web surfers accustomed to clicking daily through one or two bookmarks are increasingly out of the loop.

RSS FEEDS have become a popular way to deal with this information flow. Alongside its usual HTML pages, a website may publish a summary of its most recent news stories in an XML-based format called RSS. (The availability of a site's feed is commonly advertised with an orange XML icon.)

USERS THEN "SUBSCRIBE" TO RSS FEEDS with special reader software, which periodically collects the latest items from the user's subscriptions and organizes them for convenient reading. At any time, a user can glance at her RSS reader to get a concise picture of the news she cares about. IT'S LIKE EMAIL FOR WEB NEWS, and it's proving very popular with users.



Problem:

RSS isn't scaling well.

Feed publishers have become concerned over the way in which RSS feed data is transferred over the network. RSS readers check for news by REPEATEDLY POLLING A NEWS FEED'S URL (typically once or twice per hour). Feeds can therefore consume more bandwidth than a typical Web resource. Some publishers have begun CURTAILING RSS SERVICE to cope.

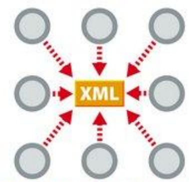
At the same time, end users want to see more timely news (that is, shorter delays between updates), so users have every incentive to *exacerbate* the stress on publishers by POLLING FEEDS EVEN MORE FREQUENTLY.

FeedTree:

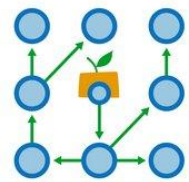
Cooperative micronews.

FeedTree addresses these problems by replacing the polling architecture of RSS with a PEER-TO-PEER (P2P) APPROACH. Users of FeedTree become "nodes" in PASTRY, A SELF-ORGANIZING P2P OVERLAY NETWORK developed at Rice. Rather than individually and redundantly polling a central server, nodes organize into MULTICAST TREES (one for each feed) to distribute new RSS data promptly and efficiently.

FeedTree-aware publishers INJECT NEW DATA IMMEDIATELY into the FeedTree network, eliminating the hour-long news delay of conventional RSS. Legacy feeds (those not multicast directly by the publisher) are polled by a subset of the nodes and shared with all subscribers.

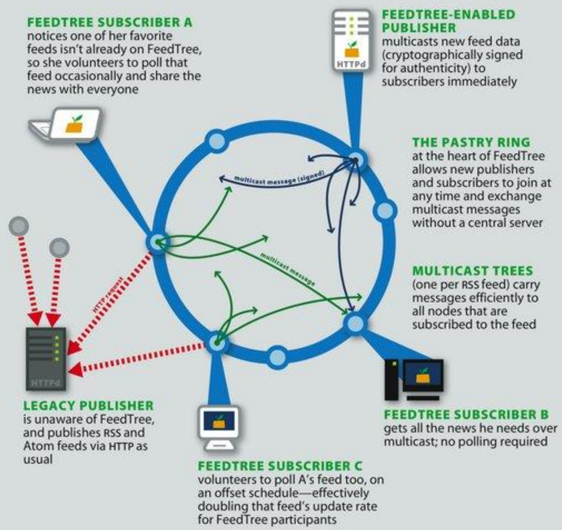


CONVENTIONAL DELIVERY
RSS readers poll the feed's Web server independently.



FEEDTREE DELIVERY
News "pushed" immediately by publisher. Nodes cooperate to forward news efficiently.

Participants in the FeedTree network.



Implementation:

Living in the real world.

We have built an HTTP PROXY that brings the benefits of FeedTree to any existing desktop RSS reader. The *ftproxy* application becomes a node in the FeedTree network and waits for the user's RSS reader to request a feed (by making an HTTP request for the feed's URL). In response to this request, *ftproxy* joins the FeedTree multicast tree for that feed, and begins listening for pushed updates. *ftproxy* will respond to future requests for the same URL by substituting the most up-to-date FeedTree updates for that feed.

SCREENSHOT: THE FEEDTREE PROXY
Web-based monitoring interface gives an overview of recent FeedTree events.

Feed authenticity.

Peer-to-peer multicast means that FeedTree users receive events from untrusted peers. Therefore publishers are encouraged to push CRYPTOGRAPHICALLY SIGNED FEED DATA using the FeedTree publishing tool. The publisher's public signing key is included in the conventional RSS feed for FeedTree nodes to download and use when verifying received data.

SCREENSHOT: SECURITY FOOTER
FeedTree appends authenticity information to each RSS entry's contents; the user can read this footer in any HTML-enabled RSS software.

Conclusion:

Better RSS service for everyone.

The FeedTree software is available today from FEEDTREE.NET. Users running *ftproxy* will see BETTER SERVICE than conventional RSS polling can provide. Publishers who install the *ftpublisher* tool will ensure TIMELY, AUTHENTIC UPDATES to FeedTree users.

FeedTree also represents a REAL-WORLD APPLICATION OF PEER-TO-PEER RESEARCH, and presents an excellent opportunity to study and improve the performance of these algorithms on real users' desktops and under real workloads.

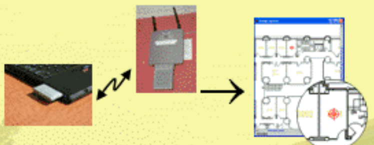
Available now at feedtree.net.

Practical Robust Localization over Large-Scale 802.11 Wireless Networks

Andreas Haeberlen Eliot Flannery Andrew M. Ladd Algis Rudys Dan S. Wallach Lydia E. Kavraki

Contact: Andreas Haeberlen · DH3001 · 713-348-3726 · ahae@cs.rice.edu

1 What does it do?



Our technique uses **Wireless Ethernet** to determine the **location** of a mobile device (PDA, Notebook...) in a building

2 Why use it?

- **Navigation:** Visitor/tourist guides
- **Advertising:** Location-aware ads
- **Robotics:** Helps a robot navigate
- **Security:** Finds 'wireless' hackers
- **Asset tracking:** Warehouses etc.

GPS does not work indoors!
Wireless Ethernet is widely available!

3 How good is it?

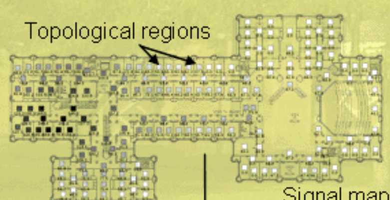
- **Accurate:** Finds the correct room in more than 95% of all attempts!
- **Good failure modes:** Incorrect results are almost always in adjacent rooms
- **Robust:** Works with different hardware and in changing environments
- **Fast:** Result available in seconds; can even track moving users!

4 What's new?

- **Much lower training time** than previous techniques (hours, not days!)
- **Calibration technique** to compensate for hardware/environment changes
- **Better robustness** due to Gaussian signal model
- **Topological localization** combined with Markov localization

5 How does localization work?

Training: Collect signal strength measurements in the entire building. This needs to be done only once!



Location estimate $\vec{\pi}_i$ — $P(o_j \mid s_i)$ — Observed signal strength $\vec{o}$

$$\vec{\pi}_{i+1} = \frac{P(o_j \mid s_i) \cdot \vec{\pi}_i}{\eta}$$

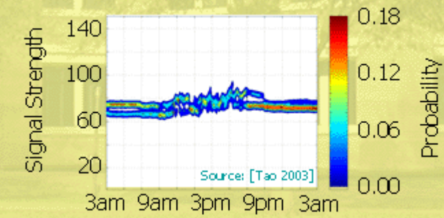
Bayes' formula

New location estimate $\vec{\pi}_{i+1}$

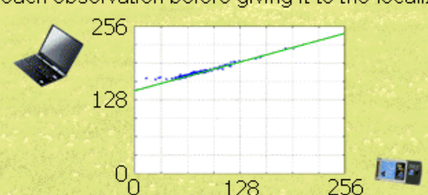
Localization: Device measures signal strength of all base stations in range and uses Markov localization to update its location estimate

6 How does calibration work?

Problem: Reported signal strength values are different for different hardware, and can change over time:



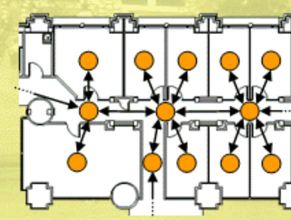
Solution: Approximate the mapping from 'old' values to 'new' values by a linear function; apply inverse function to each observation before giving it to the localizer



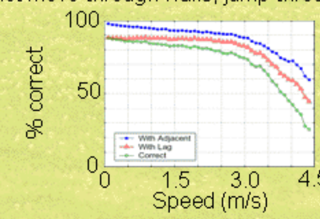
Parameters can be estimated automatically, or by collecting a few measurements at a known location

7 How does tracking work?

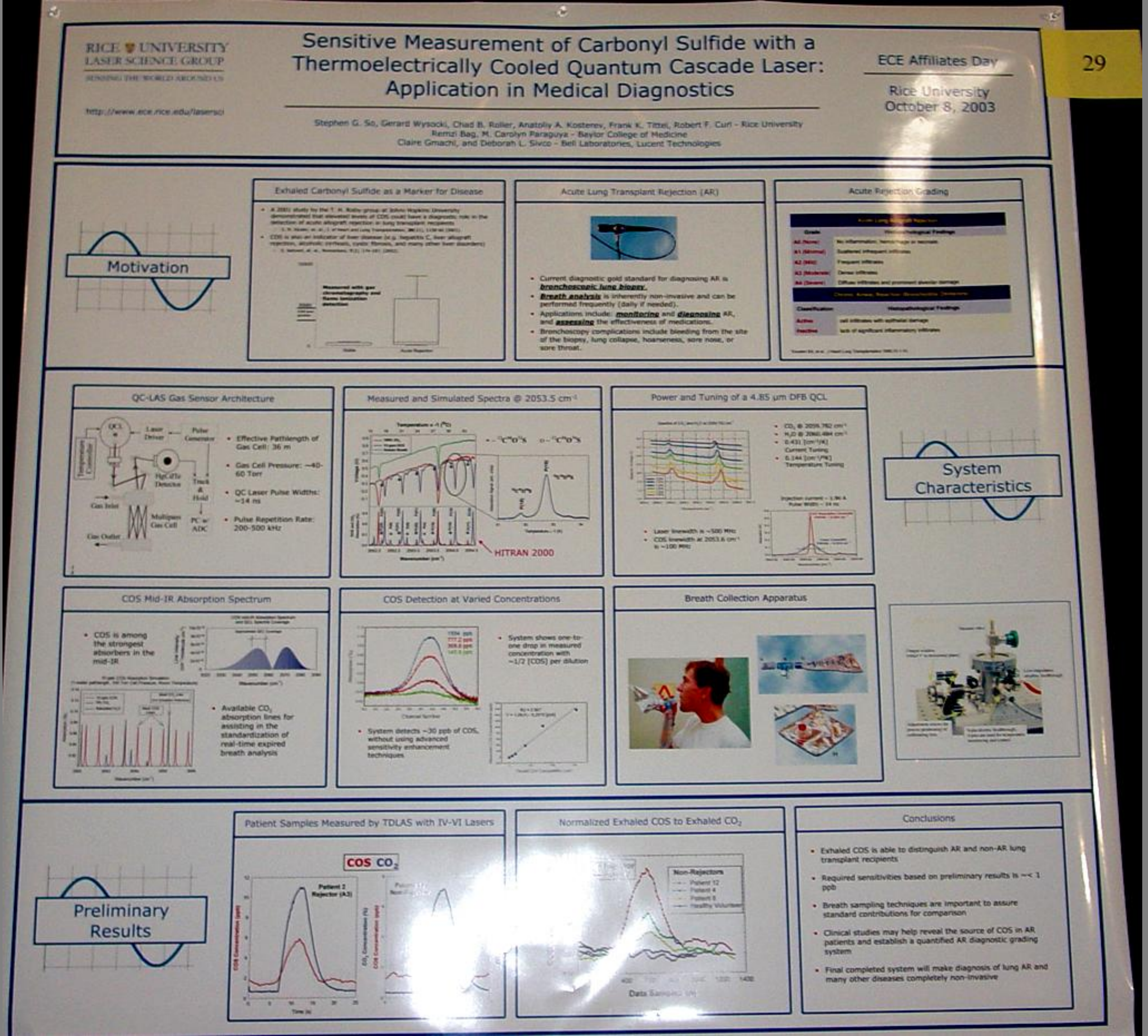
Use Markov chain to model user movement, and update location estimate after each iteration



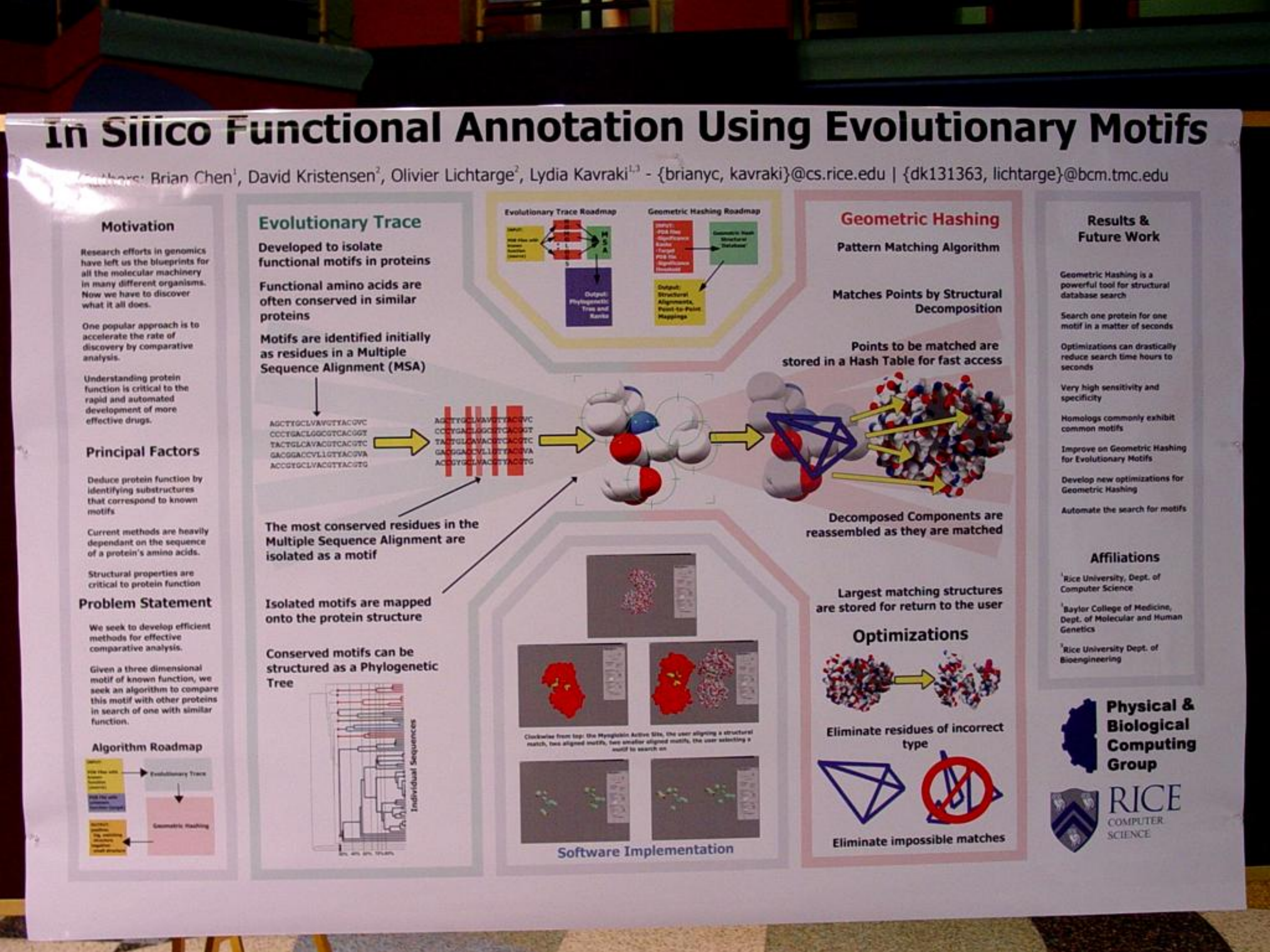
Markov chain encodes knowledge about topology: Cannot move through walls, jump through ceilings, ...

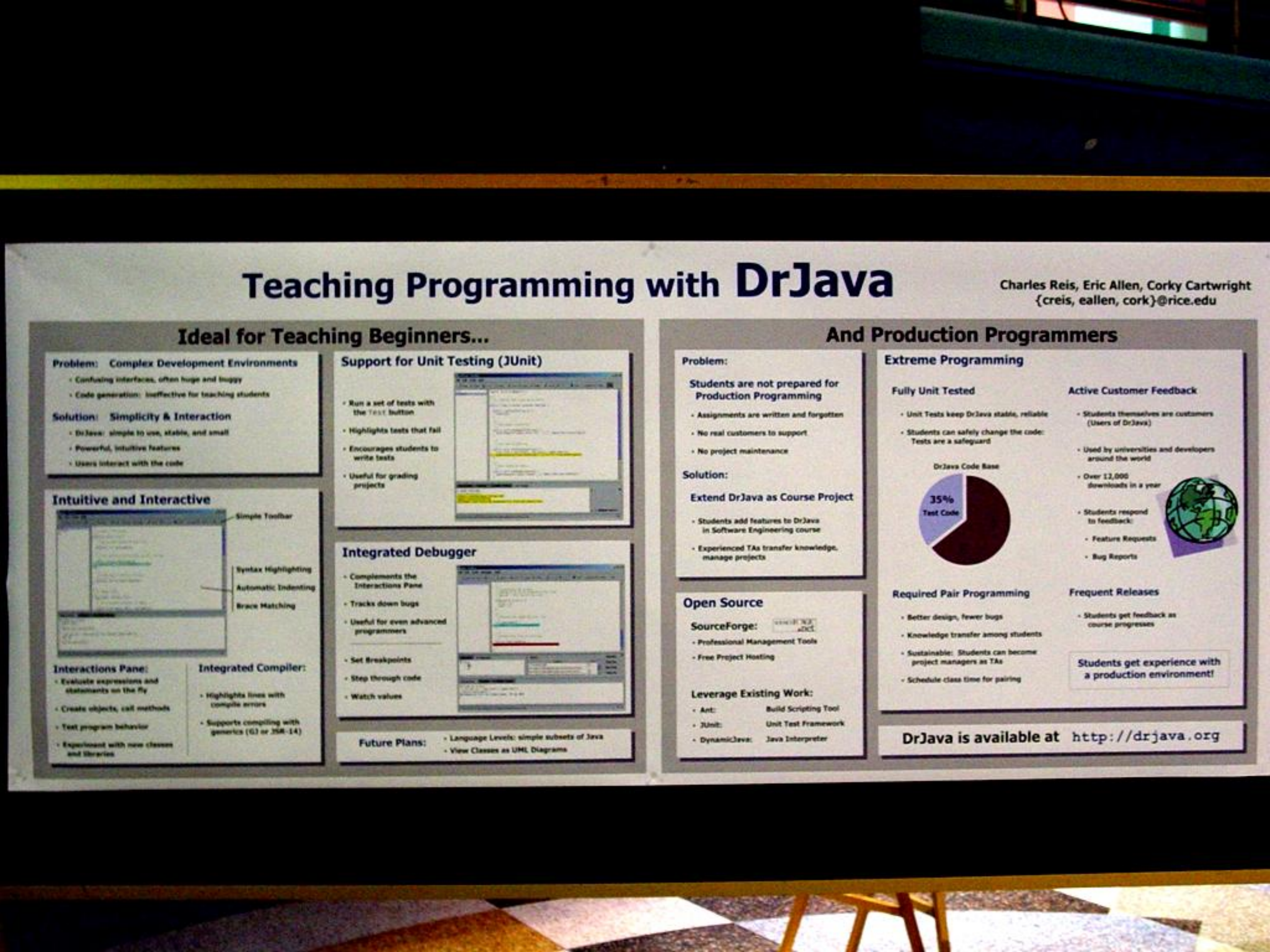


Result: Excellent accuracy up to speeds of 3-4 m/s, with one location update every 1.6 seconds



RICE UNIVERSITY
LASER SCIENCE GROUP
http://www.ece.rice.edu/lasersci
Sensitive Measurement of Carbonyl Sulfide with a Thermoelectrically Cooled Quantum Cascade Laser: Application in Medical Diagnostics
Stephen G. So, Gerard Wysocki, Chad B. Roller, Anatoliy A. Kosterev, Frank K. Tittel, Robert F. Curl - Rice University
Remzi Bag, M. Carolyn Paraguya - Baylor College of Medicine
Claire Gmachl, and Deborah L. Sivco - Bell Laboratories, Lucent Technologies
ECE Affiliates Day
Rice University
October 8, 2003
29
Motivation
Exhaled Carbonyl Sulfide as a Marker for Disease
Acute Lung Transplant Rejection (AR)
Current diagnostic gold standard for diagnosing AR is bronchoscopic lung biopsy.
Breath analysis is inherently non-invasive and can be performed frequently (daily if needed).
Applications include: monitoring and diagnosing AR, and assessing the effectiveness of medications.
Bronchoscopy complications include bleeding from the site of the biopsy, lung collapse, hoarseness, sore nose, or sore throat.
Acute Rejection Grading
QC-LAS Gas Sensor Architecture
Effective Pathlength of Gas Cell: 36 m
Gas Cell Pressure: ~40-60 Torr
QC Laser Pulse Widths: ~14 ns
Pulse Repetition Rate: 200-500 kHz
Measured and Simulated Spectra @ 2053.5 cm⁻¹
HITRAN 2000
Power and Tuning of a 4.85 µm DFB QCL
System Characteristics
COS Mid-IR Absorption Spectrum
COS is among the strongest absorbers in the mid-IR
Available CO₂ absorption lines for assisting in the standardization of real-time expired breath analysis
COS Detection at Varied Concentrations
System shows one-to-one drop in measured concentration with ~1/2 [COS] per dilution
System detects ~30 ppb of COS, without using advanced sensitivity enhancement techniques
Breath Collection Apparatus
Preliminary Results
Patient Samples Measured by TDLAS with IV-VI Lasers
COS CO₂
Normalized Exhaled COS to Exhaled CO₂
Conclusions
Exhaled COS is able to distinguish AR and non-AR lung transplant recipients
Required sensitivities based on preliminary results is ~< 1 ppb
Breath sampling techniques are important to assure standard contributions for comparison
Clinical studies may help reveal the source of COS in AR patients and establish a quantified AR diagnostic grading system
Final completed system will make diagnosis of lung AR and many other diseases completely non-invasive


In Silico Functional Annotation Using Evolutionary Motifs
Authors: Brian Chen[1], David Kristensen[2], Olivier Lichtarge[2], Lydia Kavraki[1,3] - {brianyc, kavraki}@cs.rice.edu | {dk131363, lichtarge}@bcm.tmc.edu
Motivation
Research efforts in genomics have left us the blueprints for all the molecular machinery in many different organisms. Now we have to discover what it all does.
One popular approach is to accelerate the rate of discovery by comparative analysis.
Understanding protein function is critical to the rapid and automated development of more effective drugs.
Principal Factors
Deduce protein function by identifying substructures that correspond to known motifs
Current methods are heavily dependant on the sequence of a protein's amino acids.
Structural properties are critical to protein function
Problem Statement
We seek to develop efficient methods for effective comparative analysis.
Given a three dimensional motif of known function, we seek an algorithm to compare this motif with other proteins in search of one with similar function.
Algorithm Roadmap
Evolutionary Trace
Geometric Hashing
Evolutionary Trace
Developed to isolate functional motifs in proteins
Functional amino acids are often conserved in similar proteins
Motifs are identified initially as residues in a Multiple Sequence Alignment (MSA)
The most conserved residues in the Multiple Sequence Alignment are isolated as a motif
Isolated motifs are mapped onto the protein structure
Conserved motifs can be structured as a Phylogenetic Tree
Individual Sequences
Evolutionary Trace Roadmap
MSA
Output: Phylogenetic Tree and Ranks
Geometric Hashing Roadmap
Geometric Hash Structural Database
Output: Structural Alignments, Point-to-Point Mappings
Clockwise from top: the Myoglobin Active Site, the user aligning a structural match, two aligned motifs, two smaller aligned motifs, the user selecting a motif to search on
Software Implementation
Geometric Hashing
Pattern Matching Algorithm
Matches Points by Structural Decomposition
Points to be matched are stored in a Hash Table for fast access
Decomposed Components are reassembled as they are matched
Largest matching structures are stored for return to the user
Optimizations
Eliminate residues of incorrect type
Eliminate impossible matches
Results & Future Work
Geometric Hashing is a powerful tool for structural database search
Search one protein for one motif in a matter of seconds
Optimizations can drastically reduce search time hours to seconds
Very high sensitivity and specificity
Homologs commonly exhibit common motifs
Improve on Geometric Hashing for Evolutionary Motifs
Develop new optimizations for Geometric Hashing
Automate the search for motifs
Affiliations
[1]Rice University, Dept. of Computer Science
[2]Baylor College of Medicine, Dept. of Molecular and Human Genetics
[3]Rice University Dept. of Bioengineering
Physical & Biological Computing Group
RICE COMPUTER SCIENCE



Teaching Programming with DrJava
Charles Reis, Eric Allen, Corky Cartwright
{creis, eallen, cork}@rice.edu
Ideal for Teaching Beginners...
Problem: Complex Development Environments
- Confusing interfaces, often huge and buggy
- Code generation: ineffective for teaching students
Solution: Simplicity & Interaction
- DrJava: simple to use, stable, and small
- Powerful, intuitive features
- Users interact with the code
Intuitive and Interactive
Simple Toolbar
Syntax Highlighting
Automatic Indenting
Brace Matching
Interactions Pane:
- Evaluate expressions and statements on the fly
- Create objects, call methods
- Test program behavior
- Experiment with new classes and libraries
Integrated Compiler:
- Highlights lines with compile errors
- Supports compiling with generics (GJ or JSR-14)
Support for Unit Testing (JUnit)
- Run a set of tests with the Test button
- Highlights tests that fail
- Encourages students to write tests
- Useful for grading projects
Integrated Debugger
- Complements the Interactions Pane
- Tracks down bugs
- Useful for even advanced programmers
- Set Breakpoints
- Step through code
- Watch values
Future Plans:
- Language Levels: simple subsets of Java
- View Classes as UML Diagrams
And Production Programmers
Problem:
Students are not prepared for Production Programming
- Assignments are written and forgotten
- No real customers to support
- No project maintenance
Solution:
Extend DrJava as Course Project
- Students add features to DrJava in Software Engineering course
- Experienced TAs transfer knowledge, manage projects
Open Source
SourceForge:
- Professional Management Tools
- Free Project Hosting
Leverage Existing Work:
- Ant: Build Scripting Tool
- JUnit: Unit Test Framework
- DynamicJava: Java Interpreter
Extreme Programming
Fully Unit Tested
- Unit Tests keep DrJava stable, reliable
- Students can safely change the code: Tests are a safeguard
DrJava Code Base
35% Test Code
Active Customer Feedback
- Students themselves are customers (Users of DrJava)
- Used by universities and developers around the world
- Over 12,000 downloads in a year
- Students respond to feedback:
- Feature Requests
- Bug Reports
Required Pair Programming
- Better design, fewer bugs
- Knowledge transfer among students
- Sustainable: Students can become project managers as TAs
- Schedule class time for pairing
Frequent Releases
- Students get feedback as course progresses
Students get experience with a production environment!
DrJava is available at http://drjava.org

Elections in the Auditorium: Networking voting machines for auditability

Daniel Sandler, Kyle Derr, Ted Torous, Dan S. Wallach

RISKS & RESEARCH

The story so far

Direct recording electronic (DRE) machines

Opportunities:
- Accessibility/usability
- Rapid tallying
- Procedural compliance

Risks:
- Software/hardware failures
- System insecurity & bad design
- Procedural mistakes

Experience: Webb County (Laredo)

March 7, 2006: Primary election

First local use of ES&S DRE machines

Approx. 50,000 votes cast

Margin of victory in Flores v. Lopez: **about 100** (0.2%)

We were asked to examine the voting machines

Plenty of evidence of procedural problems

Problem #1: Test votes

Election was on 3/7

93 votes apparently cast on other days

Of the voting machines involved:

- **4 machines:** clock probably set wrong
- **26 machines:** test votes counted in final tally

Problem 1, illustrated. Test votes. (Actual event log from Laredo primary)

Votronic	PEB#	Type	Date	Time	Event
5145172	161061	SUP	03/06/2006	15:04:09	01 Terminal clear and test
	161126	SUP	03/06/2006	15:19:34	09 Terminal open
	160973	SUP	03/06/2006	15:26:59	20 Normal ballot cast
			03/06/2006	15:30:39	20 Normal ballot cast
	161126	SUP	03/06/2006	15:38:37	27 Override
			03/06/2006	15:38:37	10 Terminal close

Problem #2: Lost votes

Most machines were cleared on 3/6

10 machines: cleared on 3/7

Poll workers were not supposed to do this!

Were votes lost?

Problem 2, illustrated. Machine cleared at 3:30 PM. Were votes lost?

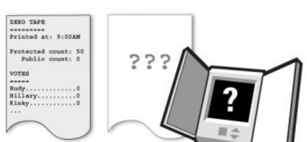
Votronic	PEB#	Type	Date	Time	Event
5140052	161061	SUP	03/07/2006	15:29:03	01 Terminal clear and test
	160980	SUP	03/07/2006	15:31:15	09 Terminal open
			03/07/2006	15:34:47	13 Print zero tape
			03/07/2006	15:36:36	13 Print zero tape
	160999	SUP	03/07/2006	15:56:50	20 Normal ballot cast
			03/07/2006	16:47:12	20 Normal ballot cast
			03/07/2006	18:07:29	20 Normal ballot cast
			03/07/2006	18:17:03	20 Normal ballot cast
			03/07/2006	18:37:24	22 Super ballot cancel
			03/07/2006	18:41:18	20 Normal ballot cast
			03/07/2006	18:46:23	20 Normal ballot cast
	160980	SUP	03/07/2006	19:07:14	10 Terminal close

Problem #3: Insufficient audit data

Many machines cleared after the election,
leaving the flash cards as only evidence

"Zero tapes" lost

"Cancelled ballot" logs not kept



Conclusion: Probably honest mistakes and poor procedure
How do we know for sure? Can we do better?

Build a better voting machine

1. *Make it harder to make mistakes on election day*
2. *Make it easier to audit the results after the election is over*

Big idea: Store everything everywhere, securely

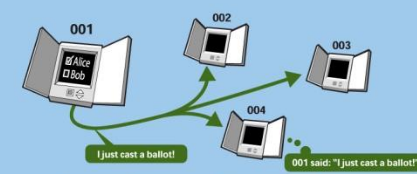
If each machine is trusted to keep its own event log and ballots, audits are meaningless

Peer-to-peer lessons: massively redundant storage; make voting machines interchangeable, disposable

Moore's Law means never having to throw anything away

"The Auditorium"

A **broadcast network** in which **all messages are signed** and every node **logs every message,** using **timeline entanglement** to provide auditable, tamper-evident records



Everyone hears everything in the Auditorium.

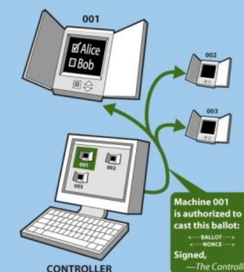
Timeline entanglement

[Maniatis and Baker '02]

Secure timelines: New messages contain crypto hashes of earlier messages (analogy: like taking a photo of today's newspaper)

Entanglement: Timelines from different nodes intersect, resulting in a tamper-evident global history of events

Usability improvements for poll workers



The network gives us the opportunity to deploy an "election controller" machine in each polling place:

- Distributes ballots to machines as necessary for each voter
- Stores & tallies encrypted ballots
- Monitors all machines
- Helps enforce correct procedures and prevent errors
- If it fails, replace it with a spare
- Joint work with Mike Byrne, Rice Computer-Human Interaction Lab (CHIL)

Result: Better auditability

When the polls close, we now have a complete picture of election day from many angles, thanks to each voting machine's Auditorium logs. Any discrepancies indicate potential irregularities worth investigating further.

This system is under development by the Rice Computer Security Lab as part of the VoteBox project.

COMPUTER SECURITY LAB · RICE UNIVERSITY

RICE

Details Matter!

- Use consistent formatting
- Check grammar & spelling
- Include contact info
- Use a correct bibliography
- Give credit to others

POST: A Secure, Resilient Cooperative Messaging System

http://freepastry.rice.edu/post

RICE

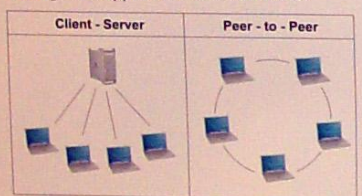
Motivation

Problem

- Current peer-to-peer (p2p) systems only used for illegal file sharing
  - Gnutella, KaZaA, Napster have provided the notion that p2p is not good for anything legal
- Are proposed p2p overlays mature enough to support collaborative applications?
  - High requirements of security
  - Existing p2p applications are simple
- Opportunity to improve existing collaborative applications (email, instant messaging)
  - Added robustness and resilience
  - Reduced cost
  - Increase security

General Solution

- Provide a generic, serverless collaborative platform, POST, based on p2p technologies
  - Create a middleware layer which enables the writing of collaborative applications
  - Use a p2p overlay, such as Pastry, for both data storage and application-level multicast



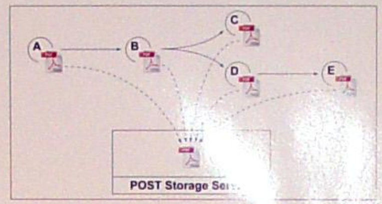
- Target applications inherit desirable properties from POST
  - Increased robustness and resilience from distributed nature of p2p
  - Reduced cost due to no dedicated servers
  - Better security from authenticated messages and default encryption of data

Architecture

- POST provides three primitives to applications written on top of it

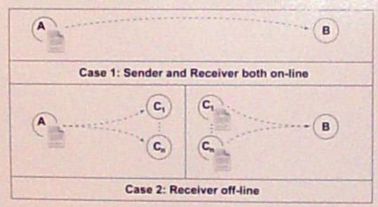
Single-Copy Data Storage

- Data is stored securely with multiple copies coalesced into one
  - In a collaborative system, sharing is common



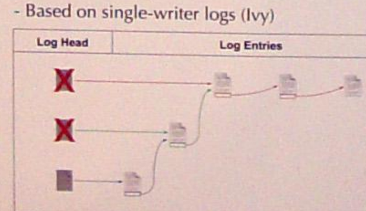
User Notification

- POST allows users to send application-specific notifications to others
  - Works regardless of recipient on or offline



User Specific Metadata

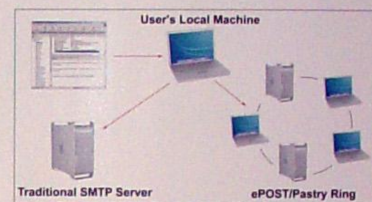
- POST provides user-specific metadata for each application it supports
  - Based on single-writer logs (Ivy)



Projected Applications

Email

- Email application ePOST compatible with existing clients and protocols
  - Users run local proxy
  - Messages broken into MIME components, each stored in Data Storage
  - Delivery using Notification service
  - Email folders represented using Metadata service



Other

- Instant messaging application imPOST
  - Uses Notification service for delivery and Metadata service for buddy lists
- Shared Calendaring application calPOST
  - Metadata service used to store appointments

Status

- POST implemented on top of FreePastry, PAST, and Scribe
- ePOST completely implemented with local IMAP and SMTP server
  - Efficiency and feasibility currently being studied within our group
  - Gaining experience with deployed p2p system
- imPOST completely implemented
  - Not yet integrated with existing IM protocols and clients

Practicalities

Editing:

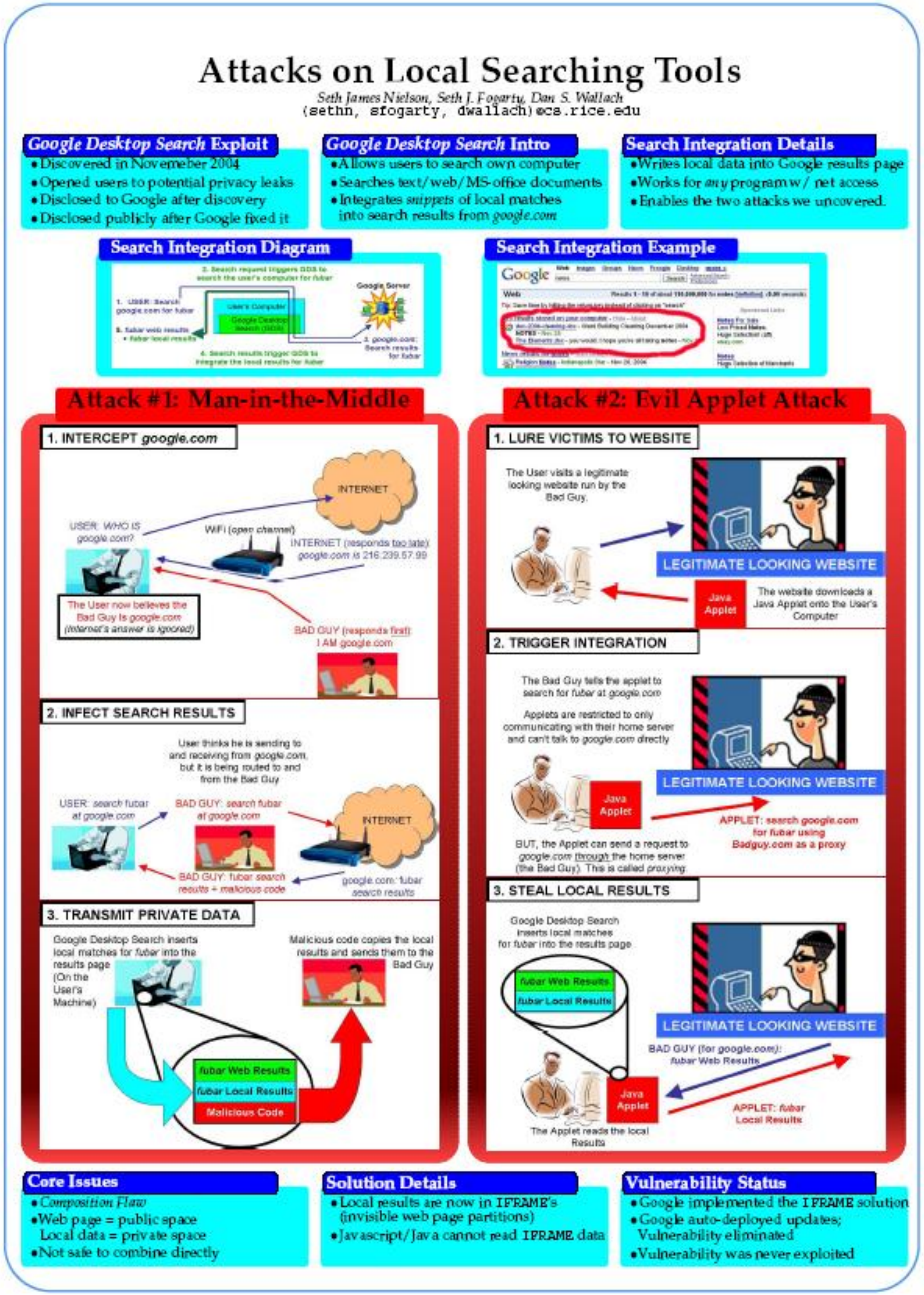
- LaTeX, PowerPoint, …
- Many templates online, or use a friend’s

Printing:

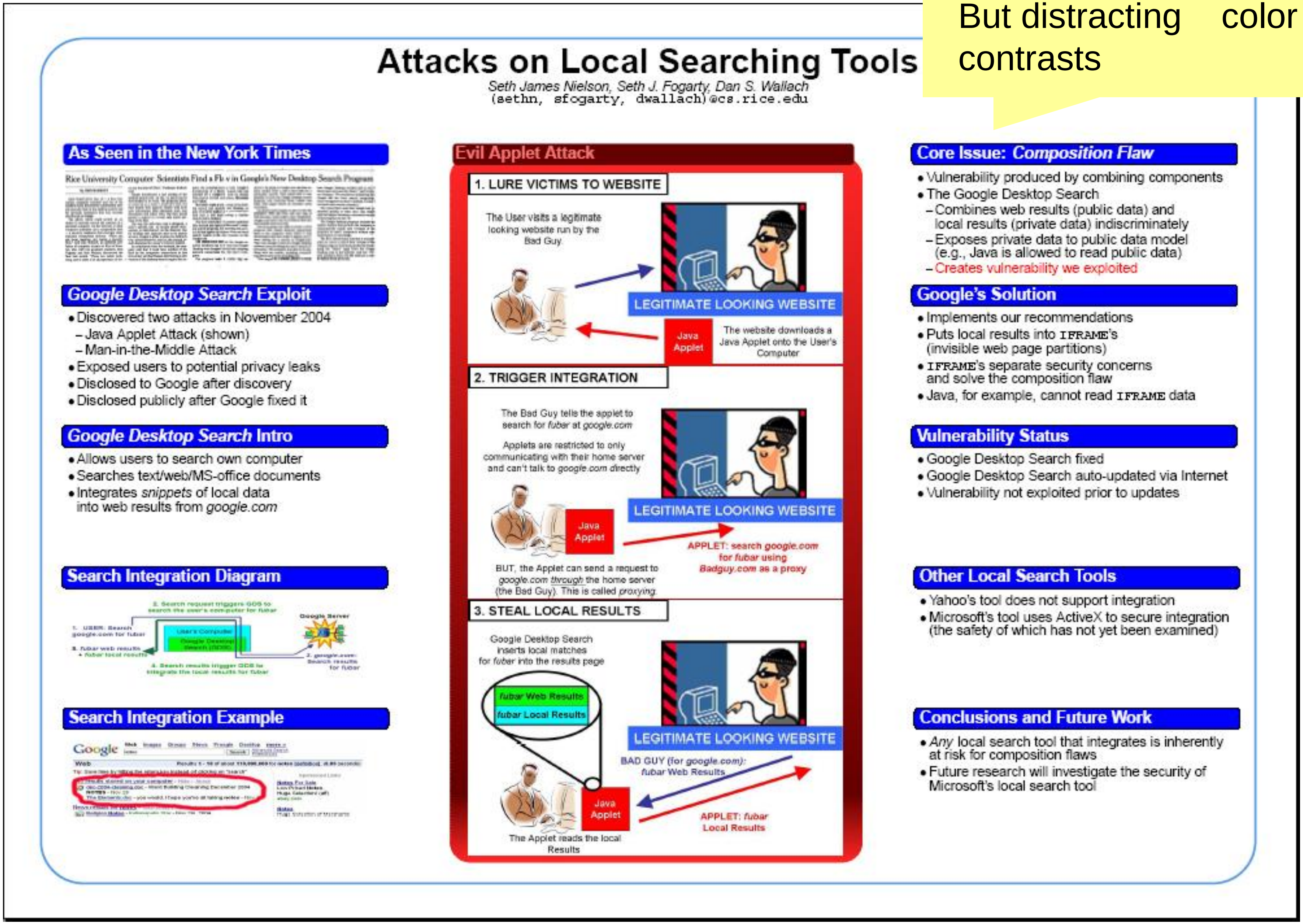
- Plotters in Earth Science: http://terra.rice.edu/videos/
- Plotter in library: http://library.rice.edu/services/dmc/resources/peripherals/printers/hp-5500ps-guidelines/
- Plotters in Mudd Building: https://docs.rice.edu/confluence/display/ITTUT/Plotters
- University Copy Center in the RMC

- Relatively expensive! (roughly $50 for smaller poster with white background)
- One plot/student paid for course – use fund #A1-739000

- Takes time – plan ahead.


Attacks on Local Searching Tools
Seth James Nielson, Seth J. Fogarty, Dan S. Wallach
(sethn, sfogarty, dwallach)@cs.rice.edu
Google Desktop Search Exploit
•Discovered in Novemeber 2004
•Opened users to potential privacy leaks
•Disclosed to Google after discovery
•Disclosed publicly after Google fixed it
Google Desktop Search Intro
•Allows users to search own computer
•Searches text/web/MS-office documents
•Integrates snippets of local matches into search results from google.com
Search Integration Details
•Writes local data into Google results page
•Works for any program w/ net access
•Enables the two attacks we uncovered.
Search Integration Diagram
Search Integration Example
Attack #1: Man-in-the-Middle
1. INTERCEPT google.com
INTERNET
USER: WHO IS google.com?
WiFi (open channel)
INTERNET (responds too late): google.com is 216.239.57.99
The User now believes the Bad Guy is google.com (Internet's answer is ignored)
BAD GUY (responds first): I AM google.com
2. INFECT SEARCH RESULTS
User thinks he is sending to and receiving from google.com, but it is being routed to and from the Bad Guy
USER: search fubar at google.com
BAD GUY: search fubar at google.com
INTERNET
BAD GUY: fubar search results + malicious code
google.com: fubar search results
3. TRANSMIT PRIVATE DATA
Google Desktop Search inserts local matches for fubar into the results page (On the User's Machine)
Malicious code copies the local results and sends them to the Bad Guy
fubar Web Results
fubar Local Results
Malicious Code
Attack #2: Evil Applet Attack
1. LURE VICTIMS TO WEBSITE
The User visits a legitimate looking website run by the Bad Guy.
LEGITIMATE LOOKING WEBSITE
Java Applet
The website downloads a Java Applet onto the User's Computer
2. TRIGGER INTEGRATION
The Bad Guy tells the applet to search for fubar at google.com
Applets are restricted to only communicating with their home server and can't talk to google.com directly
LEGITIMATE LOOKING WEBSITE
Java Applet
APPLET: search google.com for fubar using Badguy.com as a proxy
BUT, the Applet can send a request to google.com through the home server (the Bad Guy). This is called proxying.
3. STEAL LOCAL RESULTS
Google Desktop Search inserts local matches for fubar into the results page
fubar Web Results
fubar Local Results
LEGITIMATE LOOKING WEBSITE
BAD GUY (for google.com): fubar Web Results
Java Applet
APPLET: fubar Local Results
The Applet reads the local Results
Core Issues
•Composition Flaw
•Web page = public space
Local data = private space
•Not safe to combine directly
Solution Details
•Local results are now in IFRAME's (invisible web page partitions)
•Javascript/Java cannot read IFRAME data
Vulnerability Status
•Google implemented the IFRAME solution
•Google auto-deployed updates; Vulnerability eliminated
•Vulnerability was never exploited

Better organization
But distracting color contrasts



Attacks on Local Searching Tools

Seth James Nielson, Seth J. Fogarty, Dan S. Wallach
(sethn, sfogarty, dwallach)@cs.rice.edu

As Seen in the New York Times

Rice University Computer Scientists Find a Flaw in Google's New Desktop Search Program

Google Desktop Search Exploit

- Discovered two attacks in November 2004
  - Java Applet Attack (shown)
  - Man-in-the-Middle Attack
- Exposed users to potential privacy leaks
- Disclosed to Google after discovery
- Disclosed publicly after Google fixed it

Google Desktop Search Intro

- Allows users to search own computer
- Searches text/web/MS-office documents
- Integrates *snippets* of local data into web results from *google.com*

Search Integration Diagram

Search Integration Example

Evil Applet Attack

1. LURE VICTIMS TO WEBSITE

The User visits a legitimate looking website run by the Bad Guy.

LEGITIMATE LOOKING WEBSITE

Java Applet

The website downloads a Java Applet onto the User's Computer

2. TRIGGER INTEGRATION

The Bad Guy tells the applet to search for *fubar* at *google.com*

Applets are restricted to only communicating with their home server and can't talk to *google.com* directly

Java Applet

LEGITIMATE LOOKING WEBSITE

APPLET: search *google.com* for *fubar* using *Badguy.com* as a proxy

BUT, the Applet can send a request to *google.com* *through* the home server (the Bad Guy). This is called *proxying*.

3. STEAL LOCAL RESULTS

Google Desktop Search inserts local matches for *fubar* into the results page

fubar Web Results

fubar Local Results

LEGITIMATE LOOKING WEBSITE

BAD GUY (for *google.com*): *fubar* Web Results

Java Applet

APPLET: *fubar* Local Results

The Applet reads the local Results

Core Issue: *Composition Flaw*

- Vulnerability produced by combining components
- The Google Desktop Search
  - Combines web results (public data) and local results (private data) indiscriminately
  - Exposes private data to public data model (e.g., Java is allowed to read public data)
  - Creates vulnerability we exploited

Google's Solution

- Implements our recommendations
- Puts local results into `IFRAME`'s (invisible web page partitions)
- `IFRAME`'s separate security concerns and solve the composition flaw
- Java, for example, cannot read `IFRAME` data

Vulnerability Status

- Google Desktop Search fixed
- Google Desktop Search auto-updated via Internet
- Vulnerability not exploited prior to updates

Other Local Search Tools

- Yahoo's tool does not support integration
- Microsoft's tool uses ActiveX to secure integration (the safety of which has not yet been examined)

Conclusions and Future Work

- *Any* local search tool that integrates is inherently at risk for composition flaws
- Future research will investigate the security of Microsoft's local search tool