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?
Structure your story

- Say what you’re going to say
- Say it
- Say what you said

Abstract
Body & Results
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
- Weighted Averaging: 1. Taking the center of masses 2. Controllable speed of morphing (Greater affinity for bigger mass)

Rational B-spline Surfaces
Rational B-spline surfaces also consist of points with masses. The following morphing sequence depicts the difference 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).

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 collections 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.
Background:
How RSS saved the Web.
The Web has experienced an explosion of microwe, 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 rws feeds, 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 rss 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 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 pasty, a self-organizing p2p overlay network developed at Rice. Rather than individually and redundantly polling a central server, nodes organize into multicasts tree (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.

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 proxy 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, the proxy will send the FeedTree multicast tree for that feed, and begins listening for pushed updates. The proxy will respond to future requests for the same url by substituting the most up-to-date FeedTree updates for that feed.

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.

Conclusion:
Better RSS service for everyone.
The FeedTree software is available today from feedtree.net. Users running a feed proxy will see better service than conventional rws polling can provide. Publishers who install the fpublisher 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!

Topological regions

Location estimate
\[ \hat{\theta}_t \]

Signal map

Observed signal strength \( \sigma \)

\[ P(\sigma | \hat{\theta}_t) \]

Bayes' formula

\[ \hat{\theta}_{t+1} = P(\sigma | \hat{\theta}_t) \hat{\theta}_t \]

New location estimate

\[ \hat{\theta}_{t+1} \]

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.

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, ...

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

7. How does tracking work?
Result: Excellent accuracy up to speeds of 3-4 m/s, with one location update every 1.6 seconds.
Sensitive Measurement of Carbonyl Sulfide with a Thermoelectrically Cooled Quantum Cascade Laser: Application in Medical Diagnostics

Motivation

Exhaled Carbonyl Sulfide as a Marker for Disease:
- A study by [Author] et al. at [Institution] showed a significant correlation between increased levels of Carbonyl Sulfide in exhaled breath and [specific disease].
- Carbonyl Sulfide is known to be a byproduct of certain metabolic pathways, which can be indicative of disease.

Acute Lung Transplant Rejection (AT):
- Current diagnostic gold standard for diagnosing AT is bronchoscopy.
- Transplant analysis is extremely non-invasive and can be performed at lower costs (if needed).
- Applications include: monitoring, diagnosing, and assessing the effectiveness of immunosuppressants.
- Bronchoalveolar lavage (BAL) fluid analysis provides a non-invasive alternative to bronchoscopy.

Acute Rejection Grading:
- Grading of acute rejection based on BAL fluid analysis.
- BAL fluid analysis can be used to monitor the effectiveness of immunosuppressants.

QC-LAS Gas Sensor Architecture:
- Overview of the quantum cascade laser sensor architecture.
- Effective pathlength of gas cell: 36 m.
- Gas cell pressure: 400 Torr.
- QC laser power: 0.1 W.
- Laser frequency: 3.45 THz.

System Characteristics:

COS Mid-IR Absorption Spectrum:
- COS is among the most intense absorbers in the mid-IR.
- Available COS absorption lines for use in the standardization of real-time expired breath analysis.

COS Detection at Varied Concentrations:
- System detects 0.05% of COS, without using advanced sensitivity enhancement techniques.
- System shown to detect 0.001% of COS, without using advanced sensitivity enhancement techniques.

Breath Collection Apparatus:
- Overview of the apparatus used for breath collection.

COS Detected by TD-LAS with II-VI Lasers:
- Patient samples measured by TD-LAS with II-VI lasers.
- 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 are ~ 1 ppb.
- Breath sampling techniques are important to assess 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, David Kristensen, Olivier Lichtarge, Lydia Kavraki - {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 dependent 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 motifs in search of one with similar function.

Algorithm Roadmap
- Phylogenetic Tree
- Software Implementation

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.

Evolutionary Trace Roadmap
- Output: Phylogenetic Tree and Motifs

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
- Rice University, Dept. of Computer Science
- Rice University Dept. of Bioengineering
- Baylor College of Medicine, Dept. of Molecular and Human Genetics

Physical & Biological Computing Group
- RICE COMPUTER SCIENCE
Teaching Programming with DrJava

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

Support for Unit Testing (JUnit)
- Run a set of tests with the button
- Highlights tests that fail
- Encourages students to write tests
- Useful for grading projects

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 (G1 or JDK-14)

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

DrJava is available at http://drjava.org

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

Extreme Programming
- Fully Unit Tested
  - Unit Tests keep DrJava stable, reliable
  - Students can safely change the code: Tests are a safeguard

Active Customer Feedback
- Students themselves are customers (users of DrJava)
- Used by universities and developers around the world
- Over 13,000 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

Frequent Releases
- Students get feedback as course progresses

Leverage Existing Work:
- Ant: Build Scripting Tool
- JUnit: Unit Test Framework
- DynamicJava: Java Interpreter
- DrJava Code Base

Charles Reis, Eric Allen, Corky Cartwright
{creis, eallen, cork}@rice.edu
Elections in the Auditorium: Networking voting machines for auditability
Daniel Sandler, Kyle Derr, Ted Torous, Dan S. Wallach

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 #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 #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

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.
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

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

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

Status

- ePOST 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
- iPOST 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

Google Desktop Search Exploit
- Discovered in November 2004
- Disclosed only to potential victims
- Disclosed publicly after Google fixed it

Google Desktop Search Intro
- A tool users can use to search their own computer
- Searches HTML, PDF, MS Office documents
- Integrates search features into search results from google.com

Search Integration Details
- Writes tool data into Google results page
- Works for any program, not just Google
- Enables the two attacks we uncovered

Search Integration Example

Attack #1: Man-in-the-Middle
1. INTERCEPT google.com
   - The user visits a legitimate website on their computer
   - The user now believes the site is google.com
   - A legitimate looking website behind the mirror

2. INJECT SEARCH RESULTS
   - The user believes they are searching for data from google.com, but it is being replaced by the Bad Guy
   - Malicious code loads into the user's computer

3. TRANSMIT PRIVATE DATA
   - Google Desktop Search inserts local matches for user into the results page
   - Malicious code copies the local results and sends them to the Bad Guy

Core Issues
- Compromised user
- Web page = private space
- JavaScript = public space
- Not safe to contain directly

Solution Details
- Local results are now in an iframe's private web page partition
- JavaScript/Java cannot read iframe data

Vulnerability Status
- Google patched the 1752484 hole
- Google now displays updated searches
- Vulnerability eliminated
- Vulnerability was never exploited

Attack #2: Evil Applet Attack
1. LURE VICTIM TO WEBSITE
   - The user visits a legitimate website on their computer
   - The website downloads an evil applet onto the user's computer

2. TRIGGER INTEGRATION
   - The Bad Guy tells the applet to search the user's Google cache
   - Javascript applets are restricted only to communicating with their server and cannot talk to public services

3. STEAL LOCAL RESULTS
   - Google Desktop Search inserts local matches for the results page
   - Malicious code copies the local results and sends them to the Bad Guy

Core Issues
- Compromised user
- Web page = private space
- JavaScript = public space
- Not safe to contain directly

Solution Details
- Local results are now in an iframe's private web page partition
- JavaScript/Java cannot read iframe data

Vulnerability Status
- Google patched the 1752484 hole
- Google now displays updated searches
- Vulnerability eliminated
- Vulnerability was never exploited
Better organization
But distracting color contrasts