conduit!

Volume 10, Number 2 | Department of Computer Science | Fall, 2001
Brown University

ADAPTING HUMAN PERFORMANCE FOR HUMANOID ROBOTS

Nancy Pollard and DB

How can we make humanoid robots expressive, useful, or entertaining? Is there a market for humanoid robots? How can we make them safe to be around?

I had the opportunity to work in Japan for a few weeks this summer, and humanoid robots were a recurring theme of the trip. There is a large concentration of humanoid research in Japan, including significant corporate efforts by Honda and Sony. Honda has been working on humanoid robots for 15 years now. Their interest in humanoids is quite unusual when you think about it, given their primary business of cars! You have to admire their ability to think long-term. Honda is now expressing interest in making its humanoid research program begin to pay for itself, however, and their child-sized Asimo robot will be available for lease.

As an aside, here is an interesting fact about Asimo. If you see it walk, it swings its arms vigorously—more so than the other Honda robots. When I first saw this, I wondered whether this was an artistic decision or whether it contributed to the dynamics of the walking motion. According to one of the long-time researchers on this project, the dynamics of the arm swing actually are important, and swinging the arms in this way saves them about 16% of the total energy required for the walk. This is good news for those of us who believe that concepts such as energy minimization are important for understanding human (and humanoid) motion.

Sony has entered this area of humanoid research more recently. Their 50cm high Sony Dream Robot (SDR) has been positioned as the successor to the Aibo dog. With their focus on entertainment, Sony may be the first actually to turn a profit on humanoids. We saw the SDR kick a ball in a videotaped demo, so it is clear that Sony is thinking about Robocup (*www.robocup.org*) soccer tournaments with the SDR as a competitor.

> *"we heard about and saw humanoid robots that could fall down and get up again"*

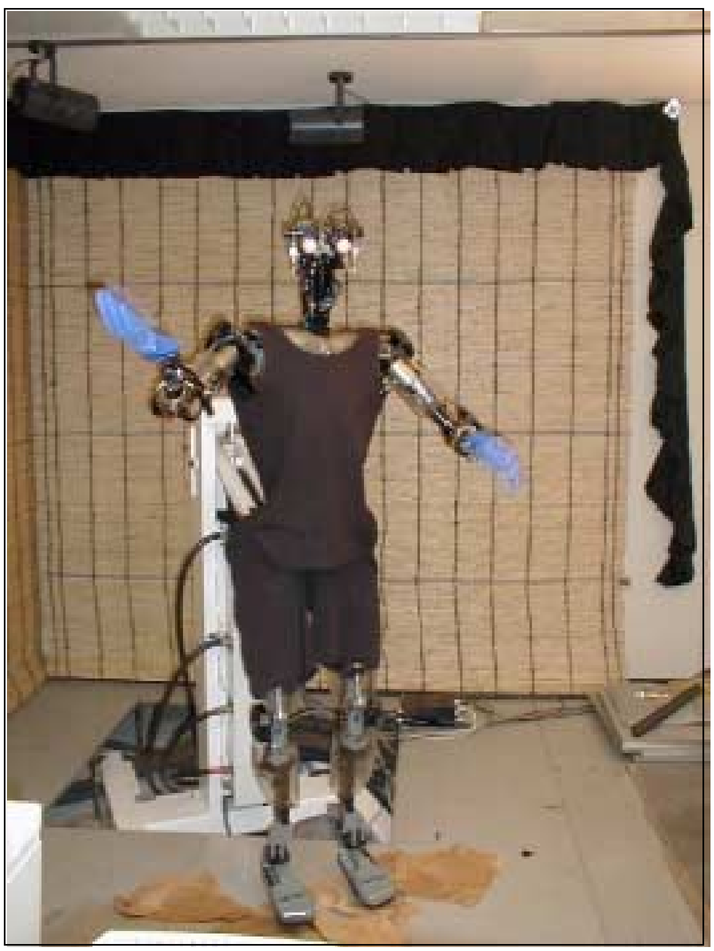
There is also a great deal of humanoid research at Japanese labs and universities. At a workshop with the theme "New Frontiers in Intelligent Robotics," hosted in Tokyo by the Japan Society for the Promotion of Science and Professor Inoue from the University of Tokyo, we heard about and saw humanoid robots that could fall down and get up again, humanoid robots whose motion was controlled by a network of tendons instead of the usual motors, and "social" robots that were programmed to interact with and entertain children. Information on these and other humanoid efforts can be found at *http://www.androidworld.com/prod01.htm*.

Along with a few friends and colleagues—Jessica Hodgins and Chris Atkeson from CMU and Marcia Riley from Georgia Tech—I spent a couple of weeks in Kyoto working with DB, yet another humanoid robot, but more about that shortly.

WHY HUMANOIDS?

One obvious question to ask is, "Why all this interest in humanoid robots?" Here are some of the reasons I heard during my visit:

(1) Man-made artifacts in the world have largely been designed with people in mind. A humanoid robot should be able to go wherever we go, reach whatever we can reach, use whatever tools we can use. Some researchers believe that a humanoid may be the most compact general-purpose robot possible.



The humanoid robot DB in action. The paper towels under his feet are to soak up leaking hydraulic fluid!

(2) People may be better able to relate to robots that look something like ourselves and that can communicate in a similar manner.

(3) It should be easier to teach robots that are similar to ourselves. Ideally, we would like to be able to just show the robot how to do something and have it perform the task correctly.

(4) There is expected to be a growing need for robots in the home. The Japanese population is aging very rapidly compared to other nations, with 25% of the population expected to be over age 65 by the year 2020. There is a great deal of interest in service robotics, especially robots for the home, and many imagine that these robots will be at least partially humanoid in appearance.

(5) The basic desire to reproduce ourselves probably figures in at some level as an additional motivating factor!

THE HUMANOID ROBOT DB

Humanoid robotics research seems to fall into two main camps at the moment that can loosely be described as navigation and upper-body skills. A large proportion of humanoid robotics research focuses on the problem of transport—getting the robot simply to traverse hard floors, carpeted floors, and sloped floors and move up and down stairs is difficult, and biped robots cannot yet walk on uneven terrain. A completely separate set of problems has to do with giving the robot the ability to manipulate objects and communicate in an expressive manner. The two sets of problems are often decoupled: if navigation skills are the primary research focus, manipulation skills of the robot may be limited; if manipulation or expressiveness is the primary research focus, the robot may have wheels or be fixed in place.

The robot DB was constructed to study higher-level functions of the brain, such as learning from demonstration. DB has 30 controlled degrees of freedom, but true to form, DB cannot walk. Although its legs do move, it is rigidly mounted in place by a mechanical connection at the pelvis. Skills that have been explored using DB range from drumming to visual tracking to Okinawan folk dance in research supported by the Kawato Dynamic Brain Project at ATR in Kyoto, Japan (*http://www.erato.atr.co.jp/DB/home.html*).

EXPRESSIVE ROBOTS?

Our interest in DB is to try to understand how human performances can be adapted to the robot with as little quality loss as possible. We chose entertainment—storytelling—as a domain because this robot is well-suited to the task and because storytelling utilizes a wide variety of motions exercising a large portion of the performer's workspace.

It is not straightforward to map a human storytelling performance to DB because the robot has many fewer degrees of freedom than our human subjects, different limb lengths, a different range of motion, and different velocity limits. In fact, we can think of the process of transferring motion from the human actor to the robot as putting the motion through a series of filters, each of which introduces error.

These filters are:

▼ Capturing the motion. We used an optical motion-capture setup in Jessica Hodgins' lab at CMU. Eight cameras record motion of reflecting balls placed on our subjects as they perform. The main source of error at this step is motion of the reflecting balls with respect to the skeleton of the performer due to motion of the clothing or skin.

▼ Mapping the motion to an animation skeleton. After the motion is captured, it is mapped to a skeleton used for animation purposes. This process also introduces error, because the joints of the skeleton may not exactly match the joint locations of the performer, and the skeleton does not allow the same types of motions as a person. It cannot, for example, capture the full flexibility of the human spine.

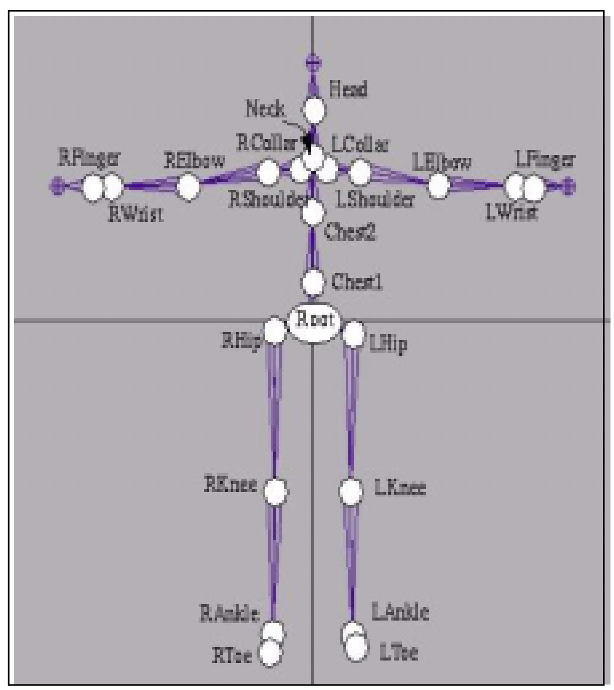
▼ Restricting the motion to the degrees of freedom of the robot. The robot has fewer degrees of freedom than the animation skeleton. The most substantial difference is that the robot has a completely rigid torso.

▼ Restricting the motion to the robot's joint limits. The robot has a much smaller range of motion than the human subjects. One perceptually salient example is the fact that the elbow joint cannot be fully extended.

▼ Restricting the motion to the robot's velocity limits. The robot also has somewhat restrictive velocity limits. Velocities are limited when the robot nears the maximum torque that can be supplied by the motors.

▼ Controlling the robot to track a desired trajectory. The robot will not go exactly where it is commanded to go, which leads to another source of error.

Jessica Hodgins' motion-capture lab at CMU



Our animation skeleton has joints in the locations shown

Our first pass, which we implemented while in Japan, was to obtain as similar a "rendering" of the original motion as possible. This meant throwing out degrees of freedom that the robot does not have, locally compressing and smearing the motion to fit within joint and velocity limits respectively, and adding a learning phase so that the robot would perform the required trajectory as accurately as possible. The figure shows a snapshot of some of our results in a single frame of the robot and human performers doing the children's song "I'm a Little Teapot."

How do we evaluate these results? In other words, how do we assess the extent of the damage done to the original performance by each level of filtering? We are designing a user study to be run this fall that will test subjects' ability to match motion samples with videos of the original actor performances. (Was this motion obtained from this actor?) We expect that the task will be trivial when the motion sample is taken from raw motion-capture data, but that it will become difficult when we present motion that has passed through a number of filters. In particular, the filter for the joint limits of the robot often results in a substantial change to the motion. It is also possible that the mechanical appearance of the robot will be sufficiently distracting to make the task more difficult than if the motion were played through an animated humanlike character.

DB trying to follow a human performance of "I'm a Little Teapot"

Even before the user study results are available, we can see room for improvement in the robot's performance. One flaw is that we treat each degree of freedom (each motor) separately, scaling the motion of that degree of freedom into the joint and velocity limits of the robot. What a person observes, however, is the motion as a whole, and meaningful gestures that comprise that motion. These gestures are a result of the coordinated efforts of many joints, and so these joints cannot be treated independently. For example, the robot has a very restricted range of motion side to side (abduction/adduction) at the waist and a more generous range of motion forward and back (flexion/extension). This difference in limits, along with our relatively simple processing technique, results in the following artifact: when the robot bends to the tune of "tip me up and pour me out," it first leans mostly to the left or right, hits the abduction/adduction joint limit, and then continues to lean forward, making the tipping motion look more circular than linear. This sort of problem is easily solved if we have a model of meaningful gestures. The tipping motion could be mapped in its entirety into a linear motion within the robot's workspace.

Even though there is room for improvement in the robot's performances, I strongly believe in the importance of human examples for creating realistic motion of both animated characters and robots. On an orthogonal track, I am exploring how to design a robot hand that is mechanically practical and has similar force and stiffness capabilities to the human hand. I hope to show that designs that are similar to the human hand in these ways will facilitate teaching grasping and manipulation skills by example. In other words, it should be easier to grasp and manipulate objects successfully following human examples with such a hand than with existing hand designs that are not so strongly based on human anatomy, simply because the passive dynamics of the hand work in your favor in an anatomically motivated hand design. But perhaps this is the topic of a future article.

Jessica, DB and Nancy

FINAL THOUGHTS

This project was in part an excuse to visit Japan and spend some time working at ATR, but it did raise some very interesting questions that we look forward to addressing in future research, such as what characteristics or features of motion should be preserved in a mapping from human actor to animated character or robot. I also enjoyed the overview of the current state of the art in humanoid robotics research. There are of course many issues to be resolved before we have C3PO wandering around, not the least of which is safety of large autonomous devices that are about as stable as inverted pendulums. Humanoid robotics research has a great deal of support behind it right now, however, and with the added ability easily to capture large datasets of motion examples, we can expect to see humanoid robots becoming more graceful, expressive, and perhaps somewhat useful in the near future.

Mark Dieterich, senior systems administrator for the graphics group, is also an amateur radio operator. When disaster strikes, the Federal Government activates the Radio Amateur Civilian Emergency Services (RACES) and places its operators on standby. After the WTC tragedy, New York City's operators were immediately called into action to provide non-secure radio networks for emergency personnel working at ground zero.

It soon became necessary to introduce fresh operators from surrounding areas and Rhode Island operators were put on standby. When the call came the weekend after the attack, Mark and four others took their equipment to the Red Cross's initial staging area in Westchester County. From there they were bused to the Red Cross in Brooklyn and thence to the site in lower Manhattan.

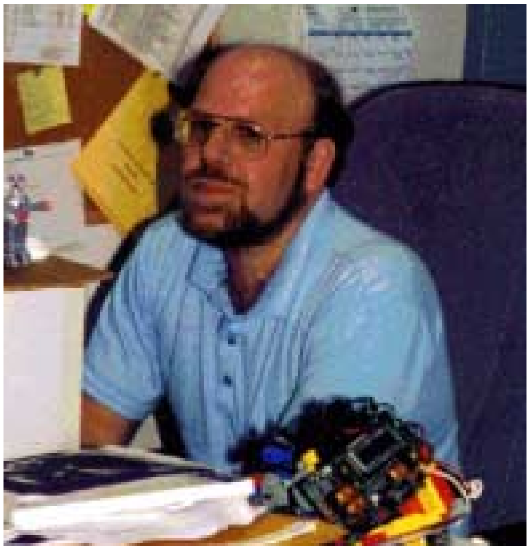
Cell phones in the area were next to useless and a stable communications network was critical for smoothly coordinated rescue efforts. The amateurs operated on a single frequency with a controlled flow of information and requests going to a net controller and then out to other operators assigned to various tasks. Mark was responsible for directing calls to various Red-Cross-run shelters: respite centers for police, firemen and other workers, shelters for dislocated people and secure government shelters. He also spent a day 'tailing' a Red Cross executive around the site.

For the last four days of his week-long stay, Mark became the day-shift manager for amateur radio operators. His job was to brief and debrief teams going into and out of the field and to deal with non-routine problems. Most 12-hour shifts were in fact 18 hours long, as they still continue to be.

Mark was a dispatcher for his hometown (Pittsford, NY) volunteer ambulance company for over four years, so he has experience with emergency situations; however, he found working at the WTC site overpowering. As the week progressed, the tight security became increasingly evident, with MPs on guard outside Red Cross headquarters. Said Mark, "If you knew New York before, it's changed—the atmosphere—people are making eye contact in the subway and are talking to one another instead of reading." Anyone involved with rescue and cleanup has found restaurant owners refusing to accept payment, taxi drivers who turn off their meters when a worker gets in; for workers, the subway is free. Being of use has helped him cope with the enormity of the tragedy.

SPINNING THE SEMANTIC WEB

The folks from ***conduit!*** contacted me a while back and asked if I would consider writing a short article for them. Having become a ***conduit!*** fan and being happy to repay the CS Department and faculty for all the time and energy they put into educating me so many years ago, I agreed.



Jim Hendler, PhD '86

At the time, I didn't realize the challenge of combining a discussion of my work with some biography—but the two are intertwined and the editor asked for both, so here's my best shot.

Let me start with a little biography. Upon completing my doctorate at Brown in 1986, I joined the faculty at the University of Maryland and started a research group in artificial intelligence. With a few lucky breaks and a lot of great students, I managed to get promoted a couple of times and became a full professor with a large research group. Much of my group's work centered on scaling up work started at Brown and trying to show that AI could be a valuable player on the then-emerging World Wide Web. A language we developed, called SHOE for "Simple HTML Ontology Extensions," gained a certain cachet in a particular research community, and we duly published various boring academic papers, developed a number of small and inconsequential demos, and did all the other things needed for me to get promoted and my students to find academic jobs.

Somewhere about four years ago, however, I realized that although the work was going well, it was also going relatively nowhere. The World Wide Web is an unimaginably large cyberspace, and even the best academic efforts rarely get any real visibility on it. I thought that if I could get, say, 10,000 people using my stuff, it would be one of the most used academic AI products ever—but it would still be being used by fewer than one in every 100,000 web users! To really change the web, and as you'll soon see that's my goal, something bigger was needed.

Unfortunately, to do something on a larger scale there's not a lot of places to play. Academics wanting this sort of impact have generally had to choose between two alternatives: start your own company or join some large software concern (generally one located in Redmond, WA). Neither of these appealed to me, as corporate America has never been an arena I found attractive. But I found a third alternative—if you want to work at web scale, join the folks who made the web happen!

The Internet, as we now call it, started out as a project called the "ARPAnet" under the sponsorship of the Defense Advanced Research Projects Agency (DARPA[1]) of the U.S. government. DARPA also helped to support work in the creation of the World Wide Web and in tools for using web information. Further, DARPA is currently the government's largest funder of research in computer science. It was clear that to have the sort of impact I wanted, DARPA was a place to consider. So in late 1998, I took a three-year leave of absence from UMD and signed up to work at DARPA.

Now I'd like to introduce my partner in crime in our effort to change the world. In the late 1980s, a researcher named Tim Berners-Lee created a program called World Wide Web that caught on pretty well. With backing from various research

1. Why ARPAnet and not DARPAnet? Generally, under Republican administrations or Congresses the organization has been asked to focus on the science needs of the DoD and has been called DARPA. When the administration is Democratic, the D is dropped, and the agency becomes ARPA and focuses on dual-use technology. The name changes often enough that the official coffee mugs say "DARPA" on one side and "ARPA" on the other, so that employees won't have to remember which one to give visitors.

agencies including DARPA, Tim moved to MIT in the early '90s and in 1994 started the World Wide Web Consortium (W3C) to help define and standardize the languages on the web. The consortium now has a membership of over 500 companies, including all the largest players in information technology and web applications. Berners-Lee is the director of the W3C and remains one of the leading thinkers about the future of the web.

One of my first acts at DARPA was to talk to Berners-Lee about an idea of his called the "Semantic Web." The work my group had been doing in SHOE was aimed at some of his semantic web ideas, and I thought some joint effort between DARPA and the W3C could help move this technology out of the laboratory and into common use. Our interaction led to two new creations—a DARPA program called the "DARPA Agent Markup Language" (DAML) and a W3C activity called "Semantic Web Advanced Development."

I recently joined with Berners-Lee and another colleague to produce a *Scientific American* article[2] describing aspects of this new work—I summarize below some of that article. I also steal from a forthcoming article, "The Semantic Web: A Network of Content for the Digital City," coauthored with a 14-year-old whiz kid named Aaron Swartz[3].

2. T. Berners-Lee, J. Hendler and O. Lassila, The Semantic Web, *Scientific American*, May, 2001. (*http://www.sciam.com/2001/0501issue/0501berners-lee.html*)

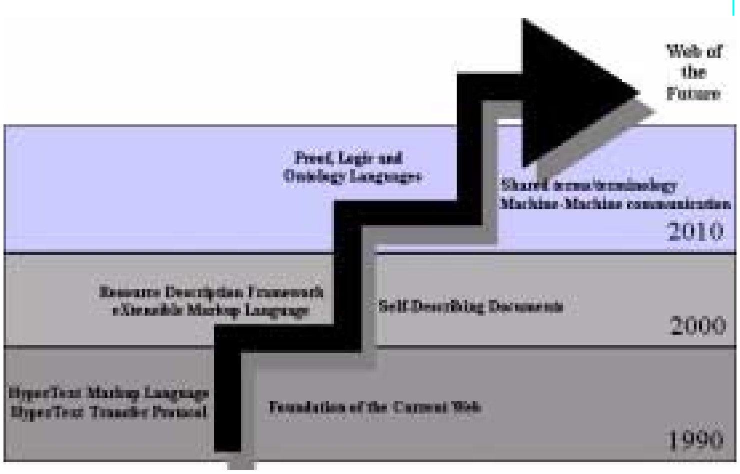


Figure 1: The evolving semantic web—new web languages and toolsets for revolutionary functionality on the World Wide Web.

The vision of the semantic web is not that of a new web but of new languages and functionalities that extend current web functionalities. On the Semantic Web, not just web pages, but databases, programs, sensors and even household appliances will be able to present data, multimedia, and status information in ways that powerful computing agents can use to search, filter and prepare information in new ways. New markup languages that make significantly more of the information on the web machine-readable power this vision and will make possible a new generation of technologies and toolkits.

The semantic web is expected to evolve from the existing web as these new languages and tools find their way into the marketplace. Figure 1, based on a figure in a recent *Nature* article[4] on the future of electronic publishing, shows the potential evolution of the semantic web. We are currently seeing new web languages like the eXtensible Markup Language (XML) and the Resource Description Framework (RDF) that let users produce more "metadata" about web resources. These metadata can be as simple as saying who produced the document and when, or complex enough actually to replace the web page with machine-readable information.

To understand the difference in using this emerging net, imagine going to a search engine and typing the query, "How many train lines are there in Japan?" If you use any of the popular search engines, you get many, many pages back (at this writing, between 122,000 and 99,000,000 answers were returned)—and few, if any, actually contain the answer to the query being asked. As a foundation for browsing the web and learning lots about Japanese transportation, this may be okay, but for answering the question—worthless!

3. A. Swartz and J. Hendler, The Semantic Web: A Network of Content for the Digital City, *Proc. 2d Annual Digital Cities Conference*, Kyoto, Japan, October, 2001. (*http://blogspace.com/rdf/SwartzHendler*)

4. T. Berners-Lee and J. Hendler, Publishing on the Semantic Web, *Nature* 410, 1023-1024 (26 April 2001) (*http://www.nature.com/nature/debates/e-access/Articles/bernerslee.html*)

Why, however, can't this answer be found? Many resources on the web could be used. First, some of the many documents found probably have this information, but current language-processing and search technologies are nowhere near good enough to find them. Second, web-accessible databases and programs could provide the answer, but word-based text matching is not sufficient to pull them out. Third, since each train line in Japan makes its presence known on the web in one way or another, a complicated program could be written to find these, identify them, and count how many there are—but writing such a program is a massive undertaking, far more effort than users would want to make for a single query.

"Sure, it's a long way from here to there—and there's no guarantee we'll make it—but the possibilities are endless, and even if we don't ever achieve all of them, the journey will most certainly be its own reward"

Academic office aide **Akina Cruz**, looking elegant in traditional African dress brought back on a recent trip to Ghana by her sister, Adeola Oredola, a senior at Brown. Despite its exotic look, this outfit is called simply a skirt and headwrap.

We can imagine a day when a query like this could elicit a very different sort of response. For example, a semantic web query tool could give replies like:

➡ *http://www.transit.co.jp/lines* says the number of train lines is over 5000.

➡ There is a database that can provide that number; please provide an authorization number.

➡ There is a web service that can compute that number; please provide 500 yen for the answer.

➡ I can get you an approximate answer by search and filtering, but it will take about 4.5 hours to compute.

The goal of semantic web research is to develop the languages and tools to make answers like this possible. (Details on current research can be found in a number of places; two of particular use are a general information site about the semantic web at *http://www.semanticweb.org* and the site describing the ongoing DARPA work at *http://www.daml.org*.)

Does this sound like a crazy science-fiction dream or a lot of hype? A decade ago, who would have believed a web of text, conveyed by computer, would change the way we live and work? This new vision unites old players such as Berners-Lee and DARPA with new visionaries such as young Aaron Swartz, who recently summed it up well. Speaking about the semantic web, he said, "Sure, it's a long way from here to there—and there's no guarantee we'll make it—but the possibilities are endless, and even if we don't ever achieve all of them, the journey will most certainly be its own reward."

As for me, I'm now back at the University of Maryland where I'm Director, Semantic Web and Agent Technology at the newly formed Maryland Information and Network Dynamics Laboratory. This lab is an attempt to develop new interaction modes between academia and industry; we hope to create an organization that can not only perform research into new web and network ideas, but also help transition these ideas into use more quickly.

I look forward to hearing from Brown colleagues, old and new, interested in the semantic web vision.

hendler@cs.umd.edu

NEW CS FACULTY MEMBER

Ugur Cetintemel

Ugur spent his first ten years in Ankara, Turkey, before moving to Izmir on the west coast. Izmir is Turkey's third largest city, very beautiful and, according to Ugur, the best city in Turkey to live in. His father is a constitutional judge (i.e. not involved with litigation), his mother a bank manager, now retired, and his older brother works in the insurance industry. He returned to Ankara to attend Bilkent University as a CS major, when he was 19. Given the stiff university entrance examinations and intense competition for university places in Turkey, Ugur's full scholarship award was a major accomplishment.

After receiving his MS at Bilkent, Ugur came to the US for graduate work in CS at the University of Maryland, College Park. He started working for Mike Franklin, a database professor, who was soon lured to Berkeley. Then in the middle of his third year Ugur changed advisors and topic area and started studying distributed systems under Pete Keleher. Ugur's current research explores data management issues for advanced distributed systems, such as mobile databases, sensor networks, and peer-to-peer systems. His broad goal is to design and build scalable algorithms and infrastructures that will efficiently support next-generation distributed applications and services. He will be teaching a topics course on mobile and ubiquitous computing in the spring semester.

Ugur is a huge sports buff and a great supporter of the soccer team Fenerbahce—"Go yellow canaries!" Besides being captain of his college soccer team, he also played basketball, enjoyed skateboarding (yes, he watched some of the Gravity Games), biking and bodybuilding. His favorite reading subject is popular physics, followed by history as a close second. He's also a classical rock fan (Pink Floyd was one of his early favorites, although he couldn't understand a word at the time!). Ugur looks forward to getting together with Pascal Van Hentenryck, a one-time provincial-level soccer player in Belgium and senior faculty member, to play soccer and perhaps start a CS team.

Ugur's wife, Gamze Tunali, is still working in Maryland at a networks company. She hopes to join him in Providence before the end of the year. She and Ugur were classmates at the Bilkent University; they were married in Maryland and remarried the following year in Turkey with both families present. Ugur spends almost every weekend in Maryland with Gamze; consequently, he hasn't even seen a WaterFire yet, but is keen to do so.

Living in Providence means everything is within walking distance and easily accessible—no more 45-minute commutes. He enjoys the many activities and good restaurants, and sees Providence as a small city but an eclectic one. Gamze loves it and is a big fan of the "Providence" TV series. She is an expert on Turkish cuisine, enjoys eating out and has a weakness for dessert! Fortunately, she can burn off the extra calories by indulging in her other favorite occupations, playing tennis and working out. They are both big sushi fans and are eager to learn about the best sushi restaurants in town. Gamze is an amateur Turkish folk dancer; she hopes to keep up with her dancing via her connection with Andy van Dam's son-in-law, who is a professional folk dancer.

Although they still feel they're in transition mode with all their possessions in Maryland, they're looking forward to settling down together in their new Providence apartment and becoming real Rhode Islanders.

EFFICIENT LOW-COST AUTHENTICATION OF DISTRIBUTED DATA AND TRANSACTIONS

Roberto Tamassia

INTRODUCTION

Security is an essential requirement of distributed platforms for business applications. We address the problem of authenticating high volumes of data and transactions in non-trusted distributed environments. Current solutions are typically:

- *Centralized*—and therefore subject to network delay and denial-of-service attacks
- *Expensive*—to build, maintain, and operate because trusted data must be maintained in a secure environment, guarded 24/7
- *Non-scalable*—with limited throughput due to operating and economic constraints

We are developing a high-throughput system for authenticating data and transactions in non-trusted environments, at the network edge and outside the firewall. We use a unique, patent-pending approach for wide distribution of authentication information. As a result, we can dramatically lower the cost of authentication in such applications as wireless authorization, B2B e-commerce exchanges, distributed storage, end-to-end integrity, tamper detection, and certificate revocation checking.

This work is being conducted in collaboration with AlgoMagic Technologies, Johns Hopkins University, and the University of California, Irvine, and is supported in part by a grant from the Defense Advanced Research Projects Agency. My collaborators Steve Baron, Robert Cohen, and Rich Sneider at AlgoMagic Technologies and Mike Goodrich at the University of California, Irvine, contributed to writing this article.

For more information, visit *http://www.cs.brown.edu/cgc/stms/*

EXAMPLE APPLICATION SCENARIO

Imagine a *trusted financial portal* service (such as Yahoo! Finance) that provides dynamic and up-to-date financial information, including corporate and news feeds, market quotes, and online trading. Some key characteristics of this distributed application are:

- High volume of accesses, queries, and transactions
- Widely distributed clients (e.g., geographically dispersed and with varying access bandwidth)
- Information originating from various sources outside the portal (e.g., stock exchanges, government agencies, and corporate investor relations offices)

How can such a portal become fully trusted so that the integrity of all data and transactions is guaranteed authentic end-to-end? Today, the economics are daunting and the implementation would be largely impractical. For example, HTML pages may be dynamically constructed from several frequently changing information sources, based on user requests and user profiles. The volume and diversity of such requests make creating individual time-stamped digital fingerprints for any given page very difficult and time-consuming. Common techniques for load balancing, replication and multi-hosting subdivisions of a web site make using previously available trust approaches unrealistic, since access to sensitive data must be limited and strictly secured.

Our distributed authentication approach offers significant new capabilities and could dramatically change the economics of authentication for distributed business applications. Highlights of our approach include:

- It works with and leverages existing Internet infrastructure, protocols and computing platforms
- It is highly scalable in terms of both computational and economic cost
- It provides succinct and timely authenticity proofs for entire collections of data originating from diverse sources
- It widely distributes authentication information to non-secure replication locations, while strictly maintaining trust
- It responds to authentication requests by clients with minimal network latency and computational cost

We have developed a distributed authentication system called ***Secure Transaction Management System (STMS)***. A fully operational prototype of STMS has been implemented.

STMS TECHNOLOGY

Authenticated Dictionaries

The computing abstraction underlying STMS is a data structure called an ***authenticated dictionary***, which is a system for distributing data and supporting authenticated responses to queries about the data.

In an authenticated dictionary, the data originates at a secure central site (the ***repository***) and is distributed to servers scattered around the network (***responders***). The responders answer queries about the data made by ***clients*** on behalf of the repository.

It is desirable to delegate query answering to the responders for two primary reasons:

1. It is undesirable that the repository provide services directly on the network due to risks such as denial-of-service attacks

2. The large volume and diverse geographic origination of the queries require a distributed system of servers to provide responses efficiently (much like DNS)

Previous Approaches

The simplest approach to implementing an authenticated dictionary is to make the responders trusted parties. This ***trusted-responder approach*** is used, for example, in the online certificate status checking protocol (OSCP), where trusted OCSP responders answer certificate revocation queries posed by clients. The main disadvantage of the trusted-responder approach is that each responder must be placed in a secure location, with ensuing large operational costs. Thus, this approach is not scalable for economic reasons.

An alternative approach entails having the repository periodically sign a fingerprint of the current version of the entire database. The database itself, together with the signed, time-stamped fingerprint, is then sent to the client as a proof of the answer. This ***database-forwarding approach*** is used, for example, to authenticate certificate revocation lists. The database-forwarding approach allows non-trusted responders and thus has lower operational costs. However, it is computationally demanding for the client, which needs to process the entire database in order to validate an answer. Also, it is not scalable for communication reasons, since sending the entire database together with the answer uses considerable network bandwidth.

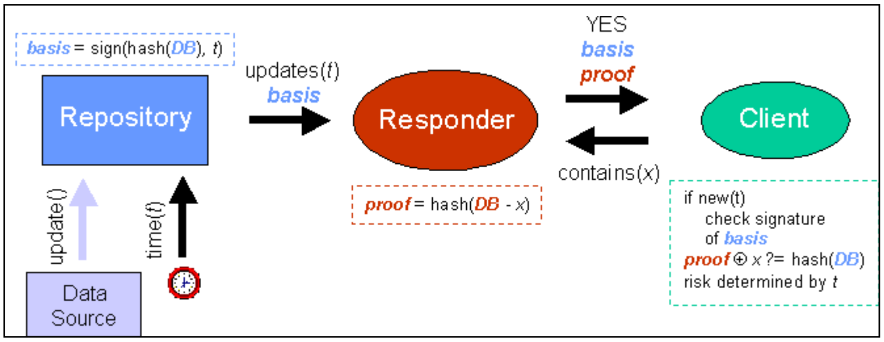
The STMS Approach

The main feature of STMS is that it maintains trust even when responders are located in insecure, non-trusted locations. That is, when a client makes a query to an STMS responder, it gets back not only an answer but also a proof of the answer. The client can easily validate the answer and determine that the responder has not been tampered with, while relying solely on trusted statements signed

"when a client makes a query to an STMS responder, it gets back not only an answer but also a proof of the answer"

by the repository. The design of STMS allows untrusted responders to provide verifiable authentication services on behalf of a trusted repository. This unintuitive yet mathematically provable fact is the key to achieving cost effectiveness.

The figure below shows a high-level view of the STMS parties and protocol. The repository sends periodic ***updates*** to the responders together with a special signed time-stamped fingerprint of the database called the ***basis***. A responder replies to a query with an ***authenticated response***, consisting of the ***answer*** to the query, the ***proof*** of the answer and the ***basis***. Informally speaking, the proof is a “partial fingerprint” of the database that, combined with the subject of the query, should yield the fingerprint of the entire database. A proof consists of a very small amount of data (less than 300 bytes for most applications) and can be validated quickly. The client finally evaluates the risk associated with trusting the answer using the freshness of the time-stamp.



Our patent-pending algorithms, which employ only standard cryptographic functions for signing and hashing, guarantee that the following tasks can be performed in near real time and with minimal use of computing resources:

- Creation of the new signed basis by the repository for each time quantum (the basis is incrementally updated)
- Update of the database copy residing at a responder for each time quantum (the copy is incrementally updated)
- Assembly of the proof by a responder for each query (the proof is obtained by combining precomputed partial proofs)
- Query formulation by a client
- Proof verification by a client

The key benefits to the STMS approach include:

- Answers given by the responders are as trustworthy as if they came from the repository
- Data is distributed close to the clients, minimizing network delays
- Deploying responders is inexpensive
- The repository is protected from risks such as denial-of-service attacks

An additional unique advantage of STMS is that it supports ***historical persistence***; that is, one can perform queries on past versions of the database, which makes possible a variety of nonrepudiation applications.

STMS comes with Java and C++ toolkits for building clients, including modules for validating responses. For maximum reliability and security, cryptographic functions are performed with the available standard cryptographic libraries of Java and Windows. We also provide an XML-based protocol for querying a responder over an HTTP connection. We are currently working on a Web services approach in which STMS is accessed through the SOAP protocol.

The table on the opposite page compares the STMS approach with the trusted-responder and database-forwarding approaches with respect to operational cost, run-time performance, scalability, and support for historical persistence.

APPLICATIONS

As a foundation for delivering a wide range of trust services, STMS brings authentication to the network edge and can be viewed as a distributed low-cost “authentication cache” for high-volume transactions. Classes of trust applications in which STMS can be exploited include:

- User authentication and access control for large-scale corporate portals, private exchanges, and Virtual Private Networks—User signon can be accomplished without the need to contact a remote central server.

- Wireless two-way authentication—Mobile device identity can be confirmed by gateways, network services, and applications, and devices can also confirm the identity of selected network services and applications to establish two-way trust.

- Trusted XML Web Services—Two-way trust can be established between the parties involved in a "chain" of Web Services. Keys for signed or encrypted portions of XML can be obtained from widely distributed servers (STMS responders) rather than centralized systems.

- Certificate status checking (e.g., for SSL, secure email, and code certificates)—Identity and status can be mutually verified by both sender and receiver of email, publisher and user of code components, client and server in a secure dialogue.

- Validation services for business partner networks (e.g., healthcare providers)—Access rights, roles, and derived trust assertions can be quickly checked against a localized store of consolidated credentials. Public networks can be employed instead of incurring the cost of a private value-added network.

- End-to-end integrity for content collection and distribution systems (the example scenario presented at the beginning of this article).

- Non-repudiation services for electronic commerce—Third-party verification and audit capabilities can be implemented to support large-scale B2B networks (e.g., corporate portals, exchanges, trading systems), as well as high-volume B2C systems (e.g., media distribution, digital tickets, electronic payment systems).

	Cost	Performance	Scalability	Persistence
Trusted-responder	High	High	Low	No
Database-forwarding	Low	Low	Low	No
STMS	Low	High	High	Yes

- Secure DNS—Host name to IP address mappings could be upgraded to include digital signatures. The data-distribution approach would be essentially similar to that used today. Widely distributed trusted DNS servers (as STMS responders) would not have to be located in secure settings.

- Tamper detection for Web sites—Web page digital fingerprints can be made available as a reference. The computational load and delay of signing every page on each web server at all replicated locations are avoided.

"Access rights, roles, and derived trust assertions can be quickly checked against a localized store of consolidated credentials"

- Tamper detection for operating systems—System file fingerprints can be made available as a reference on a localized basis for large groups of client systems.

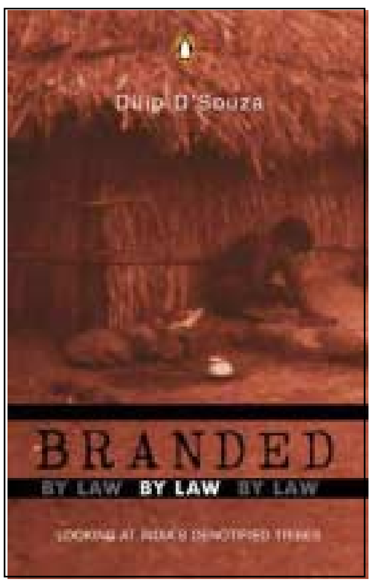
- Tamper detection and license checking for software applications—Software is increasingly distributed or accessed over the network. Metered access can be effectively implemented, identity and status of servers and clients can be mutually confirmed, and component fingerprints can be quickly verified.

ALUMNI EMAIL

DILIP D'SOUZA, MS '84

In August, Penguin India published my first book: *Branded by Law* (see *www.penguinbooksindia.com/books/aspBookDetail.asp?ID=4567*). It's about some of India's most forgotten people—its denotified tribes, once actually defined as criminal. But more than that, it is also a look at prejudice, a phenomenon that shapes their lives every day and also greatly interests me for various reasons.

Now that it's out, I get asked all the time: CS, but now you write? What's the story?



I came to writing late, true. It never occurred to me as a possible career in my university years, nor in my years in software. But when I returned to India in 1992, I tried writing for a lark. Got published a few times. Wrote some more, then more regularly. And pretty soon it was clear to me that I had found what I truly wanted to do with my life. Write.

Yet I never regretted my time in CS, because it strikes me as marvelous training for writing. Not everybody would take this route, I know. But the lessons from CS serve me well when I write. At its best, and I'm sure ***conduit!***'s readers will agree, programming is an intensely creative process. Good software is every bit as satisfying and elegant as a good piece of writing. And of course the word "elegant" is one all programmers aspire to. Software described that way is an exhilarating mix of functionality, leanness, clear thought and a certain beauty.

I try to write like that: clearly, simply. Cut to the core of issues, find connections, be sure of myself. These are the attributes of writing that I value, that I aspire to. (Also to having someone tell me someday that I write "elegantly".)

All this, in many ways, is where this book of mine came from. Too much happens around us that is based on unquestioned stereotypes people hold about each other. One such hounds denotified tribes: the impression that they are all vicious criminals. In reality, they are not particularly more (or less) criminal than anyone else, but that means little. So via the experience of these forgotten people, I try to get my readers to ask questions about prejudice, to ask questions above all of themselves. I try to do this with those tools I first learned about in CS: reason, clarity, cut the fluff.

And the interesting thing is that while a lot of people see it as startling, this apparent "switch" from CS to writing, I see it as a smooth and even logical transition. It's almost as if I write, and write the way I do, because of my years in CS. I owe that to Brown and to several people who feature regularly in these pages. So please read my book! Tell me what you think. *dilip@alumni.brown.edu*

RICHARD HUGHEY, PhD '91

Hi all! This past year I have bumped into a few Brown folks. The most surprising was at a July computational biology conference (ISMB) overhearing talk of rubber chickens. On turning around, I mentioned having one in my office (one decaying original rubber chicken, and three plastic replacement juggling chickens!). The speaker turned out to be Sonia Leach, not only a Brown student but also occupying my old office—555. We chatted about what we were doing, and I was able to clear up her puzzlement about how a VLSI chip plot wound up in the window of 555—it's from my dissertation. I've gone on to supervise the design, construction, and use of another parallel processor at UC Santa Cruz.

Ethan Miller (ScB '87) joined our CS department this year; I remember seeing his name a few times as a grad student, perhaps in 169 (operating systems).

Being an involved graduate student has caught up with me, and now I'm chair of our computer engineering department. I ran into Tom Dean and Jeff Vitter at the CRA meeting last March as a pre-chair and, not ducking quickly enough, the position caught up with me two months ago. I won't go into all the pleasures of the position, as then no graduate student would ever get involved in anything for fear of eventually becoming a department chair!!

Parts of the job, like the planning of research and academic programs, are quite fun—CE has five positions this year, starting the ramp-up to UC's "Tidal Wave II" enrollment surge. I've also been developing BS (starting 2001-2), MS and PhD (starting 2002-3, we hope) curricula for Bioinformatics degrees with my colleagues Kevin Karplus (protein structure prediction) and David Haussler (putting together the human genome, among other things). We're planning to build this into a department in a couple of years, and are currently recruiting three faculty in bioinformatics and related areas.

Otherwise, Santa Cruz is a great place—with a campus of redwoods, I quickly overcame my trepidation at interviewing on the west coast! *rph@ce.ucsc.edu*

JOSH MILLER, AB '96

Hello out there to the Brown CS Community—Well, I decided to mail a little change-of-address notice to Suzi Howe and she craftily cornered me and asked me to submit a quick update to ***conduit!*** on what I've been up to, not realizing that such a request would elicit a dual confession of unprecedented proportions. So here it goes: 1) I haven't done anything related to computer science since I graduated and 2) despite that, I still read my ***conduit!*** cover to cover (including the technical articles) every time it comes in the mail. Weird, eh? So what exactly have I been up to? Well, in 1996, when most of my graduating class (including the philosophy majors) went straight from Brown to cushy jobs working in the booming Silicon Valley tech economy, I began working for Tufts University Hillel in Boston, and then a year later moved to UC Berkeley Hillel in California, where I was earning a non-profit salary living just above the poverty line. Hillel is the international organization that serves the needs of Jewish college students (I logged many hours as a student at the Brown Hillel when I wasn't in the CIT), and in my misleadingly titled job of "Program Director" I spent my days meeting Jewish students, planning and attending fun events with them, and playing my guitar. In addition, because of my extensive computer science background, I was also regularly called upon to teach my co-workers how to perform highly technical computer-related tasks such as: sending group emails, copying files to and from our local area network, replacing printer toner cartridges, and of course, programming the VCR. All in all, the job was a great experience for me.

I enjoy working with people, and find that I fit in comfortably in the non-profit scene, so with that in mind, I have decided to go back to school to brush up on my management skills before trying my luck as a director of my own non-profit organization. So, I have just recently left Berkeley and moved to Evanston, Illinois, where I will be starting as a business student at the Kellogg/Northwestern School of Management. Ironically, it seems that a large number of my classmates have CS undergrad degrees and are former dot-commers who decided to sell their BMWs and apply to business school when the economy collapsed and their companies went under! I guess the old adage is true, "If you don't know where you're going, any path will take you there."

Despite my unusual path, I have always been glad I went the CS route at Brown, and in a true liberal arts spirit I believe that my whole CS experience—TAing, group work, and even building computer systems—was a great educational preparation for my work since graduating. I'd love to hear from folks, and encourage other Brown CSers on nontraditional routes to write in and tell their stories! Josh Miller, *jmiller@songleaders.org*

JON MOTER, ScB '99

Spike forwarded the following, which Jon Moter posted to *brown.cs.stupid*:

With Microsoft's new version of Office, they're finally killing off the annoying Paperclip Assistant. As a marketing move to push MS Office XP, they've created a site for Clippy, bemoaning his own fate and looking for a job. It's actually fairly amusing: *http://www.officeclippy.com*. He has a resume (*http://www.officeclippy.com/resume.html*) with past work and education experience. Lo and behold, it turns out he graduated from Brown in '94 with an Art/Semiotics degree. Apparently he "graduated *cum laude*, with a performing arts thesis that involved twisting myself into a representation of Michelangelo's David." Always good to see Brown alums in prominent positions. Jon "You still have an account???" Moter.

Said Spike: "So...even though the paperclip's not from Brown CS, perhaps it deserves mention in ***conduit!*** (or to our lawyers...)"

SCOTT SMOLKA, PhD '84

Thought this photo might be of interest to ***conduit!***'s readers: three of Paris's Ph.D. students: me, Alex Shvartzman Ph.D. '92 and Dina Goldin Ph.D. '97. I was in Boston visiting Dina; we're working on a joint paper with Peter Wegner about Turing machines, transition systems and interaction. Alex was visiting Nancy Lynch's group at MIT. The restaurant is somewhere in Kendall Square. All the best, Scott. *sas@cs.sunysb.edu*

l to r, Scott Smolka, Alex Shvartzman and Dina Goldin

27th IPP SYMPOSIUM VISION-BASED INTERFACES

Symposium host, Michael Black

The paradigm of human-computer interaction based on windows, icons, menus, and pointing devices has changed little since its development in the early 1970s. With the exception of some limited voice-recognition technology, a computer's knowledge of the external world comes through this screen/keyboard interface. In contrast, one-on-one human interactions involve a great deal of context in which the speakers are aware of their shared environment and the objects in it as well as each other's facial expressions and gestures.

This sort of rich interaction is not possible with today's computers due to their limited intelligence and their inability to perceive the world outside them. The 27th Industrial Partners Program Symposium, held last May, brought together leading researchers from companies that are attempting to revolutionize the interaction between machines and humans by endowing computers with perceptual capabilities. Speakers from Microsoft, Intel, Compaq, Mitsubishi, and IBM spoke about the fundamental science needed to enable computers to "see" and understand their users. They also presented some novel "perceptual user interfaces" that explore how these abilities might change the way computers and humans interact. We first heard about the "EasyLiving" project from **Dr. John Krumm of Microsoft Research**. At their Redmond research facility, Microsoft has built a "living room" with computer screens embedded in the walls that can provide information and entertainment (figure on next page). The room is also equipped with cameras and other sensors that detect the location of people in the room, de-

IPP Symposum speakers from l to r: John Krumm, Microsoft Research; Michael Black, Brown; Matthew Brand, MERL; John MacCormick, Compaq; Jim Rehg, Compaq; Myron Flickner, IBM Almaden; Gary Bradski, Intel Corporation

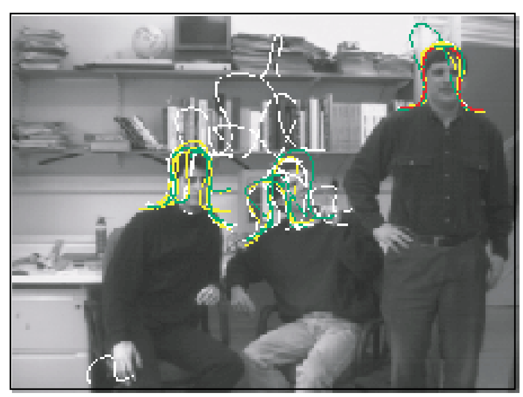
termine their identity, and track them as they move about. The project is exploring the kinds of services that might be sup-

Microsoft Research, EasyLiving Project. This "intelligent" living room senses people in it

ported by this sort of "intelligent room." For example, the room can "learn" individual preferences that are used to change the lighting, heating, or music selections. Additionally, when a known person enters the room, their computing environment follows them with their email, bookmarks, and instant messages.

Supporting these applications are a number of cameras that compute the three-dimensional structure of the room and detect the location of people. A tracking system combines information from the cameras and pressure sensors in the furniture to track multiple people in the room at one time. These features provide a testbed for exploring novel interfaces for product groups at Microsoft.

Reliable detection and tracking of people in complex, changing environments such as the living room of the EasyLiving project is still an open problem. **Dr. John MacCormick of Compaq's Systems Research Center** spoke about their probabilistic methods for people tracking. Their BraMBLe system builds probabilistic models for what the background (room) and foreground (people) look like. These models are used in an elegant probabilistic formulation that exploits Bayesian probability theory to combine information measured from a camera with past measurements and prior knowledge of the world. The result is a

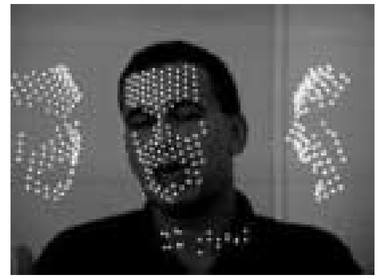


Compaq System Research Center. Probabilistic tracking of multiple objects in a scene "real-time" tracker (figure above) that can detect multiple people from a single camera, determine how many people are in the scene, and track them even when they partially occlude each other.

Dr. Jim Rehg talked about related work at **Compaq's Cambridge Research Laboratory** that uses body tracking in a "smart kiosk." This kiosk uses video cameras to detect people that are near it and track their motion (figure at left). A display screen has a realistic-looking human face that talks to people and encourages them to approach the kiosk. An installation of the kiosk in the Cybersmith Cafe in Harvard Square allowed Compaq to assess the potential market for kiosks with sensing capabilities. This installation also let Compaq experiment with different features and evaluate this type of perceptual user interface. The researchers observed that the interactive talking kiosk drew people to it and that high-quality content was the most important factor in holding their attention. Furthermore, the experiment revealed that interactive entertainment was more compelling in this application than information services. To improve the interaction with the kiosk, Dr. Rehg's recent research is focusing on probabilistic methods for detecting whether or not a person in front of a camera is actually speaking by analyzing the motion in the video in conjunction with the auditory signal.

Compaq Cambridge Research Laboratory. Smart kiosk installed in Cybersmith Cafe

While the above systems focused primarily on detecting and tracking people as crude "blobs," many types of human-to-human interaction exploit detailed information about the face and hands. Recognizing information about facial expression requires an analysis of facial motion, which is challenging due to the complex deformations the human face can undergo. **Dr. Matthew Brand of the Mitsubishi Electric Research Laboratory (MERL)** described a novel mathematical formulation of this motion estimation problem that uses matrix algebra to propagate uncertainties in image measurements in such a way that information loss is minimized and optimal estimates of the facial motion can be derived. He showed impressive results of tracking a face in a monocular video sequence while recovering its three-dimensional shape and motion (figure below).

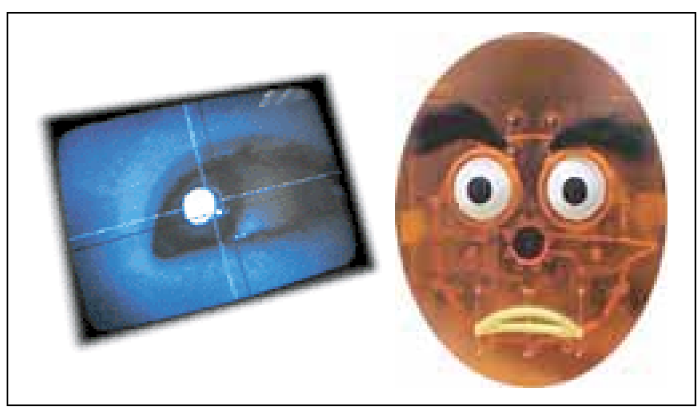


Mitsubishi Electric Research Labs. 3D facial motion tracking with uncertainty

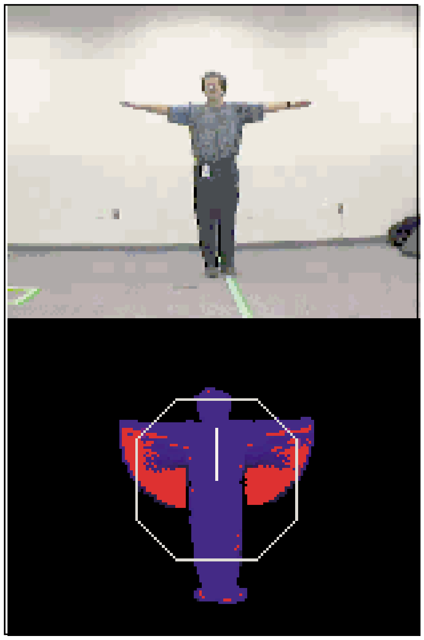
The performance appears more robust than previous systems and the information derived from the motion estimation can be used for both expression recognition and the animation of 3D graphics models of faces.

The motion of the eyes is of particular interest in human-computer interfaces. **Dr. Myron Flickner of the IBM Almaden Research Center** described their "Blue Eyes" project for detecting and tracking eyes using infrared light and a video camera (figure below, left). By locating the pupil and reflections on the cornea, they are able to track the human gaze accurately.

To explore new perceptual interfaces, IBM's new robot Pong can detect faces and facial expressions as well as express them (figure below, right). Pong has speech recognition and synthesis software that lets it interact with a user while a camera hidden in its nose detects faces and analyzes facial expressions. Dr. Flick-



IBM Almaden Research. l, BlueEyes gaze tracking; r. lifelike Pong robot ner gave a live demonstration of Pong and talked about other projects at IBM exploring "attentive" user interfaces that combine physiological measurements, eye gaze, and body language that they hope will make computers more widely useful.



Intel. Human activity analysis from the Intel Open Source Computer Vision Library

The above methods have one simple thing in common that distinguishes them from traditional interfaces—the visual processing is computationally intensive. Intel is in the business of selling computation, so their interest in these new perceptual interfaces is not surprising. **Dr. Gary Bradski from Intel's Microprocessor Research Labs** presented their efforts to develop an open source computer vision library to support the growth of novel applications. This library of vision software provides a huge number of basic and advanced methods that support everything from low-level segmentation to full-body tracking and probabilistic action recognition (figure at left).

While the form of future interfaces cannot be predicted, it is clear that many of our partner companies believe that they will change radically and that they will exploit new perceptual capabilities. In particular, this workshop explored computer vision technologies that might give computers information about the visual world occupied by their users. With the active research on this problem in academia and industry, it is a fertile area for partnerships like those supported by IPP.

For more information about these projects, see:

Microsoft, EasyLiving project: *http://research.microsoft.com/easyliving/*

Compaq, Smart Kiosk: *http://crl.research.compaq.com/projects/kiosk/default.htm*

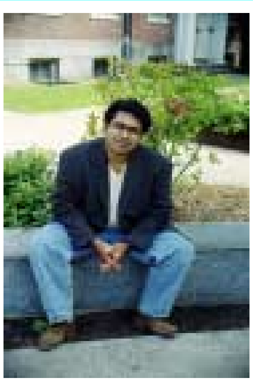
MERL, facial motion: *http://www.merl.com/people/brand/*

IBM, Blue Eyes *http://www.almaden.ibm.com/cs/blueeyes/*

IBM, Pong *http://www.almaden.ibm.com/almaden/media/image/pong.html*

Intel, Open Source Computer Vision Library: *http://www.intel.com/research/mrl/research/opencv/*

HOW TO CREATE THE BEST OO PROGRAMMERS



Shriram Krishnamurthi

Got you to look, didn't I? I promise I'll get to this subject at the end. Maybe you want to just jump right there—go ahead!

BACKGROUND

The TeachScheme! Project was conceived in frustration. Our group at Rice University had spent several years overhauling Rice's introductory curriculum. Rice had adopted an approach based on Abelson and Sussman's seminal work at MIT but, despite its sublime content, it wasn't doing the job for us: it provides immense insight into structuring *systems* but, we felt, gave little insight into structuring *programs*.

In response, Matthias Felleisen, Corky Cartwright and their students (including me) built a new curriculum for introductory programming. Like Abelson and Sussman, we steered clear of the nagging terminology and low-level details that dog so many introductory programming texts. Unlike them, we stressed the systematic construction of programs from crisp and accurate definitions of data—the stuff of OOP, in other words, without the verbiage and overhead of OO. This let us take students with no prior programming experience through all the interesting topics—lists, trees and other useful data types, higher-order functions and non-local control, as well as the routine stuff of scope and mutation—with a very particular emphasis on program design.

Over the years, three features have driven our success.

When asked to elaborate on some jpegs of him windsurfing, Tom Dean said, "You can ask Keith Hall and Stu Andrews. I walked down to the beach a few blocks from our house on a Saturday a couple of weeks back and found Keith and Stu throwing a frisbee, with Stu's windsurfer lying nearby. Stu asked me if I wanted to give it a try and I said sure. I used to do quite a bit of wind-surfing, having planned vacations around wind-surfing at Maui, Columbia River Gorge, the Sea of Cortez in Baja Mexico and numerous other windsurfing meccas. My short stint on Stu's board was the second time I've windsurfed in three years. I was lousy but Stu and Keith were supportive."

Keith is a fourth-year PhD student, Stu is in his third year.

1. We developed a methodical response to that most frustrating of questions, "Can you help me find my bug?" Our work presents a six-step design *recipe* that begins with an understanding of data, then proceeds to understanding the black-box behavior of the function, then to deriving its template from the data. Only in the fifth of the six steps does the student actually write the function. (The last, in case you're wondering, is testing.)

This methodology not only gave our TAs and us a grading rubric, it also gave students a valuable debugging checklist. To our surprise, we found that most student errors had little to do with their code: their errors were at a much earlier phase, usually in defining the structure of their data. If you don't understand the data you have, you obviously can't hope to process it very well. We suddenly developed a much more sophisticated understanding of student errors (a big help when planning what to revisit in lecture, for instance). For their part, students now had a much more useful debugging methodology than making random syntactic changes and putzing about in a debugger.

2. We built a beginner-friendly programming environment, DrScheme ("Doctor Scheme"). DrScheme is an outgrowth of several years of observing how students program in labs. Its innovations are the fruit of several PhD theses. From a pedagogic perspective, one feature towers over others. It exposes the fundamental mismatch between texts, which provide a stratified view of languages, and environments, which toss the student into a linguistic mass for which they're often completely unprepared. DrScheme instead provides *language level*s that a student can increment as her skills grow. In particular, each language level is careful to provide feedback using only the language and terminology a student is expected to know at that level. In contrast, pretend you're in your first week as a programming student, type `'wage * hours = salary'` into a C++ environment, and make sense of its response. (We've observed high-school students nearly reduced to tears from trying to decipher repeated feedback of this sort.)

3. We exploited the multiplatform, graphical nature of DrScheme to build numerous libraries to support interesting exercises. The libraries help students

write simple games, build very rudimentary animations (no, no, not like Andy's or Spike's!), try a little Web programming, create file-system managers, and so on. The libraries plug into our Scheme's graphics and systems APIs, but students write all the critical control elements. Most of these are extended exercises that grow in complexity, letting students peel away layer upon layer. In short, we expose them to two important software engineering principles—model-view-controller designs and iterative refinement—all while they're having fun!

THE EARLY DAYS

We've taught this material since 1994 at Rice, and it's been a great success. Enrollments skyrocketed; perhaps more importantly, the number of women grew disproportionately. Many who'd taken programming classes in high school (especially women) said they enjoyed the more structured and goal-driven curriculum, in contrast to the seemingly random activity they'd experienced before. When students mastered even our rapid pace of material, we fried their brains in bonus lectures.

> *"Enrollments skyrocketed; perhaps more importantly, the number of women grew disproportionately"*

One small group of high school students, however, always resisted this material. Over time, we came to realize most of these were students who'd had Advanced Placement computer science (AP CS) courses. Most AP students recognized that, by about the sixth week, our course was in terrain they found completely unfamiliar. A small group, however, resisted the call to treat the material at its own level of abstraction. Common questions and complaints included, "Isn't that just a linked list?" "Where's the null pointer?" "Why won't the number overflow?" "Isn't recursion bad for the stack?" "This doesn't behave like a computer." What amazed us was not simply how frequently these students complained, but how misinformed they were.

When we eventually set to studying the AP CS material and talking with high school teachers, we realized why these students had these views. The AP CS material has a lowest-common-denominator feel; indeed, almost uniquely among the AP disciplines, many leading universities (including Brown and Rice) don't recognize scores in this subject. This lack of recognition stems, in part, from a huge disparity in depth between the high school and college levels.

Despite the best intentions of its framers to remain at a principled level, we found that teachers in the field were forced to deal with a myriad of low-level details of syntax and machine organization, to the point where most never got to useful data structures or other concepts. Many teachers reported that material we'd considered routine—programming over trees, for instance—was presented gingerly, at the end of two semesters, and only the best students were expected to get it absolutely right; most never extracted themselves from the long dark night that is debugging. In contrast, we considered it a failure if virtually every student, aided by the design recipes where necessary, didn't nail this material.

Matthias Felleisen then asked the question we hadn't dare pose: Why can't we teach this material in high schools? What's the real difference between a high-school student and a first-year college student anyway, save for a summer of frolic? (Not *our* students, who're too high-minded to spend their summers that way, but maybe someone else's.) Despite what our viewbooks would like to suggest, students aren't magically transformed when they walk through a Van Wickle Gate or a Sallyport. So why not take this to high schools?

GROWTH

Yep: we had no idea what we were in for. We were used to college educators, to whom you can mail a book with a friendly note. But high school teachers—especially in computer science—are so overworked, it's utterly unfair (and impractical) to simply mail them a manuscript. (Computer science professors joke about being asked to fix printers; for teachers, it's part of their job.) In the furious market of the late 1990s, many computer science teachers left for industry; the ones remaining had no training in the discipline at all (and *they* were teaching C++!). They were valiantly keeping their classes alive, but learning a whole new approach, unaided, was out of the question.

In the summer of 1996, therefore, we brought in a high-school teacher for a few days of intensive training. We had no idea what she knew or how to teach her, and I imagine those few days were pretty rough on her. But she survived it, so the next year we brought in two. (Exponential growth! We had visions of Malthus.) As our first big test, in 1998 I recruited a group of 18 teachers from Texas computing conferences to attend a week-long summer workshop at Rice. Looking back, we still knew very little about our audiences; it's a wonder that many of those teachers are still with us. (What doesn't kill us makes us stronger.)

Suddenly, TeachScheme! had become real. We were conducting school visits, observing how students learned C++ and Scheme. Somewhere along the way, we made a semantic shift: our on-line "lecture notes" grew into a "book." Having trained the teachers, we became victims of our limited success: as teachers began to use the material in classes, we were obliged to support them. We had to write and publish solutions. We began to write a programming environment user's guide. We added new libraries and exercises. We revised the language levels. The codebase grew to over half a million lines.... In 2000, our demand grew too large to fit into one workshop, so we ran two. We drew teachers from as far afield as Mexico, Saudi Arabia, Switzerland and Japan. (Okay, so maybe Mexico isn't all that far from Houston.) We also began to benefit from the diaspora of PhDs: Matthew Flatt ran a small workshop at his new home, the University of Utah.

This year, the workshops themselves went national. We ran workshops in four different states, training nearly a hundred teachers. The largest, hosting nearly forty teachers, took place in July, 2001 at Brown, run jointly by my colleague, Kathi Fisler, of Worcester Polytechnic, and me.

CARPODACUS MEXICANUS ON 5TH FLOOR

Grad student Jasminka Hasic played host to a family of house finches during the spring/early summer. The pair began building their nest atop the open window of her 5th floor office; consequently, the window had to remain open come wind and weather. In order to discourage the birds from flying into the room and becoming disoriented or hurt, Jasminka, herself a newly hatched American citizen, used her giant-sized American flag to cover the window and darken the room. The nest proved a remarkable, though messy, feat of engineering, teetering on the edge of oblivion. The female finch laid four eggs, all of which hatched. The highly vocal babies were pretty distracting at feeding time, but very charming. Despite concern that the babies' first attempt at flight would be their last, each one fledged successfully.

We, of course, merely soaked up the credit; our helpers made it work smoothly. We had three great returning teachers and five superb students: three PhD students from Brown (Greg Cooper, Rob Hunter and Dave Tucker), and one each from Cornell and Northeastern. This student group makes us feel like we've taken the first step towards closing an educational cycle.

One of the major attractions of our curriculum is that it costs teachers, students and schools nothing. We distribute software and support material free on the Web. Our text, *How to Design Programs*, is published in hardback by the MIT Press, but we asked to be able to distribute it free on the Web, and they generously agreed. Some poorer schools, both in the U.S. and abroad, use the book entirely through this medium. Finally, with generous support from the National Science Foundation, U.S. Department of Education, Exxon and CORD, an educational nonprofit that's adopted our curriculum for its Academies of Information Technology, we pay expenses and stipends to virtually all teachers who attend workshops.

OOP

We're (fortunately) lowering the exponent in our growth curve. More importantly, we're seeing signs of the change we set out to accomplish. A growing population of teachers recognizes that there's an alternative to what they teach. Our "repeat customer" ratios (teachers who use the curriculum more than once) are huge. Almost every week, we hear from either a teacher who struggled before who says he's thriving now, or one who was comfortable before but whose students are now building projects beyond her expectations. Perhaps one way to see that TeachScheme! has come of age is that prominent universities other than our own, both in the U.S. and abroad, have adopted our curriculum.

"One reason for TeachScheme!'s growing university adoption is its superb conduit to true object-oriented programming"

One reason for TeachScheme!'s growing university adoption is its superb conduit to true object-oriented programming. Corky Cartwright at Rice worked with us to revamp the second course to teach students algorithms and design pattern-based programming in Java. The beauty of this scheme is that once students have internalized the TeachScheme! design recipe, it takes just about a day to start producing authentic Java! Scheme easily morphs into code obeying the Interpreter and Composite patterns; their Scheme experience with higher-order functions has prepared them well for inheritance-based abstraction; even the Visitor pattern is an easy step away. Students get through most of the Gang of Four book, in addition to the usual algorithmic material, in their second semester.

A few high-school and college faculty who've adopted this route are finding that their students internalize OOP *better* through this two-step approach than they did starting with Java. By the time they confront their first '`public static void main`' the students are already masters of useful data structures and several introductory algorithms. Having already learned one language, the second one comes a lot quicker—because the Scheme we teach maps easily to Java. The design recipes entirely take the mystery out of recursion (sorry!) in a way that no prior approach we know of has. In short, there's no excuse for not teaching challenging and interesting topics.

People sometimes ask me whether functional programming isn't unintuitive; perhaps students should really only see it as (college) juniors. My evidence to the contrary is simple. I point to hundreds of students, starting from as young as the 7th grade, at schools big and small, poor and rich, in the U.S. and in several foreign countries. These students write rich, interesting and even fanciful programs. We hope it isn't just the language: the environment certainly helps, and teachers tell us the design recipe wins big, too. But the bottom line is, while functional programming may not be intuitive to professors, it sure is to kids. (And it seems to be increasingly intuitive to programmers in everything from JavaScript to Python—as their programmers hit on the same problems functional programmers did two decades ago, their languages have rapidly expanded in their functional offerings, as numerous articles on these languages attest.)

THE FUTURE

What's next? We're building many more extended exercises and improving the quality and features of DrScheme. These improvements induce difficult research problems behind the scenes, with the benefit (or danger, if you don't like the attention!) that the solution will immediately download to thousands of desktops in the next release. The challenge we face as researchers and programmers is the same one that actors face: always to stay in character. How do you build a powerful type-inference engine that doesn't baffle a 10th grader? Stay tuned!

But I think much more is at stake. I've believed for some years now that computer science has come to reside at the heart of a true liberal arts curriculum. What matters isn't the dust and noise: the details of word sizes and endianness. Rather, it's the notion of computation, what Abelson and Sussman called a *procedural epistemology*. This is why computer science has gone from bridesmaid (think of those early number-crunchers) to guest of honor (think biology).

Our curricula must reflect this shift. We need to give students more than just the syntactic details of some language; we must train them to harness this beast called computation. If anything, the non-majors need it more than the majors. In one semester (if that's all they'll give us), we must teach them principles they will recognize for decades to come. To do this, our courses have to get at the heart of computation as quickly as possible and stay there as long as they can. People must internalize these ideas so well that they recognize them in whatever they choose as a concentration.

Brown is leading the way here. Last year, Lisa Cozzens '01 completed a senior honors thesis under my supervision in which she built several extended exercises implementing basic processes from biology. That material caught the eye of a new audience: biology teachers. Two of them attended the workshop at Brown this past summer. Now they've begun to teach programming to their students, too. The revolution has begun.

Acknowledgments: Thanks to my partners in crime, especially Matthias Felleisen, Robert Bruce Findler, Matthew Flatt and Kathi Fisler. Thanks also to the other TeachScheme! staff, and to the teachers who've endured us!

URLs: The Project:
http://www.teach-scheme.org/
The Text:
http://www.htdp.org/

fac.activities@cs.brown.edu

Michael Black. In the spring Michael organized an Industrial Partners Program Symposium on Vision-based Interfaces (details in this issue).

He has recently been awarded two grants: an NSF ITR grant on "The Computer Science of Biologically Embedded Systems," (a collaboration with Elie Bienenstock in Applied Math and John Donoghue in Neuroscience) and an ONR grant to study "Motion Capture for Statistical Learning of Human Appearance and Motion" as part of the DARPA Human-ID project.

In July Michael attended the International Conference on Computer Vision held in Vancouver. He was one of the Area Chairs for the conference and his students presented two papers. Hedvig Sidenbladh presented a paper on "Learning Image Statistics for Bayesian Tracking," while Fernando De la Torre talked about "Robust Principal Component Analysis for Computer Vision." Along with David Fleet from Xerox PARC, Michael had an invited paper, "Probabilistic Detection and Tracking of Motion Boundaries," in the Distinguished Paper Track at the International Joint Conference on Artificial Intelligence held in Quebec during August.

During a busy spring of travel, Michael gave an invited talk on human motion tracking at Workshop on the Convergence of Vision, Video, and Graphics in Berkeley and talks on brain-computer interfaces at the Microsoft Research Vision Symposium and at the Workshop on Vision-Based Perceptual Interfaces hosted by the Interactive Institute in Stockholm, Sweden. He was back in Sweden in August to give an invited talk at the 2001 Stockholm Workshop on Computational Vision, held on an island in the Stockholm archipelago. Additionally, he gave colloquia on human motion analysis at the University of Rochester, the University of Western Ontario, the University of Pennsylvania, and New York University.

This fall Michael is teaching a new course on brain-computer interfaces that is drawing curious students from departments such as physics, neuroscience, applied math, cognitive and linguistic sciences, and engineering, as well as computer science.

Eugene Charniak. Eugene presented his paper "Immediate Head Parsers for Language Modeling" at the 2001 Association for Computational Linguistics (ACL) Conference, this year held in Toulouse. Eugene's technical article in the last ***conduit!*** was based on this paper. The ACL conference is generally considered to be the top conference in this area, and this year for the first time it awarded a "best paper" award. Eugene's paper shared the award with one other group.

Tom Dean. Tom spent half of August in Seattle, attending meetings of the board of trustees for IJCAI Inc., giving talks at IJCAI-related and satellite meetings and workshops including an invited talk at ATAL-2001 (Agent Theories, Architectures and Languages), and trying to run the department and CIS from his hotel room. (Tom was appointed interim Vice President for Computing and Information Services as of July 1; the appointment runs through next June, at which point he hopes to turn over the job to a permanent CIO.)

Amy Greenwald. Like last year, one of the highlights of Amy's summer was her participation in the Distributed Mentor Program, sponsored by the Computing Research Association. Her mentorees, Victoria Manfredi and Julia Farago, visited from Smith College and Harvard University, respectively. In addition, Amy traveled from eastern to western Canada. In May, she presented a paper at Autonomous Agents in Montreal. In August, she chaired a workshop entitled "Economic Agents, Models, and Mechanisms" in Seattle, from where she took a backpacking holiday in British Columbia (a.k.a. BC a.k.a. Bug City). And in the middle of the summer, Amy attended ICML (International Conference on Machine Learning) at Williams College with David Gondek and Keith Hall, both PhD students in their fourth year.

Shriram Krishnamurthi. In addition to having the usual amount of fun, Shriram ran a successful TeachScheme! workshop at Brown (see article on page 19). He and Kathi Fisler of WPI hosted nearly 40 high school teachers from around the country and abroad, training them in an innovative computer science curriculum. He has more at *http://www.teach-scheme.org/*.

David Laidlaw. David has been awarded an NSF CAREER grant, the agency's most prestigious award for junior faculty. His work under this grant will focus on shape modeling and its applications coupled with the development of a methodology for teaching the skills needed for successful multidisciplinary research projects. The education plan consists of David's course, 'Interdisciplinary Scientific Visualization,' and a research group; both are aimed at undergraduates and graduate students alike. The research effort includes development of computational tools for capturing geometry, representing it within the computer, and using those representations for specific applications in archaeology and biological modeling.

Nancy Pollard. Nancy was recently awarded an NSF CAREER grant on "Quantifying Humanlike Enveloping Grasps." Nancy's work under the grant will focus on quantifying humanlike enveloping grasps with the goal of creating credible hand use for digital characters. Producing realistic digital humans has been called the last frontier in the march toward graphical realism, and Nancy believes that the last frontier in creating digital humans is generating believable hand motion. In pursuit of this goal, she proposes a tendon-based quality measure for humanlike enveloping grasps, and she plans to evaluate this quality measure (1) for ability to discriminate between grasps, (2) as a predictor of grasp forces, and (3) for use in modeling grasp acquisition. Because of the strong emphasis on human anatomy, this research has the potential for additional impact outside graphics and animation in areas including ergonomics (tool design), robotics (robot hand design), and anthropology (research in human hand evolution and tool use).

This summer she was invited to attend the first Asia-Pacific Advanced Studies Institute, with the theme "New Frontiers of Intelligent Robotics," in Tokyo, Japan. She is on junior sabbatical leave this year, and will be spending much of the year at Carnegie Mellon University in Pittsburgh. She and her students will be using the motion-capture lab at CMU and working with Jessica Hodgins' group on problems related to realistic animation of human motion and control of robots from motion-capture data.

The 40 TeachScheme! workshop participants gather for a group shot outside the CIT building

John Savage. John chaired the Inaugural Faculty Program Committee that assembled 20 exciting panels reporting on faculty research at President Ruth Simmons' inauguration this October. He is currently Chair of the Faculty and is now in his second year as an elected officer of the Faculty. As of January 1, John will be withdrawing as Director of the Industrial Partners Program to be replaced by Michael Black; however, he will work with Michael over the next year to bring him up to speed.

Eli Upfal. Eli was on the program committee of the 13th International Symposium on Fundamentals of Computation Theory in Riga, Latvia, and the VIII International Colloquium on Structural Information and Communication Complexity in Spain. He participated in a Dagstuhl meeting on "Design and Analysis of Randomized and Approximation Algorithms" in June and was an invited speaker at a workshop in honor of Allan Borodin's 60th birthday at the University of Toronto, also in June.

David Laidlaw, John Hughes and Andy van Dam. Brown presented four papers at this year's Symposium on Interactive 3D Graphics (John Hughes was the program co-chair with Carlo Sequin of U.C. Berkeley, and Science and Technology Center collaborator Mary Whitton was the general chair). Joe LaViola (Ph.D. student of Andy van Dam) presented work that he did with Daniel Acevedo Feliz and Daniel Keefe (Ph.D. students of David Laidlaw) and Robert Zeleznik (graphics staff) presented work on "Hands-Free Multi-Scale Navigation in Virtual Environments." Then Dan Keefe presented a paper on "Cave Painting," a joint project with Daniel Acevedo Feliz, Tomer Moscovich (Ph.D. student of John Hughes), David Laidlaw and Joe LaViola. In Cave Painting, an artist uses virtual brushes and paints to create an artwork that lives in the 3D space of the four-wall Cave.

Michael Kowalski (graphics staff) presented a paper on "User-Guided Composition for Art-Based Rendering," joint work with John Hughes, Cynthia Beth Rubin (of RISD) and Jun Ohya of ATR Research in Japan.

Finally, Takeo Igarashi, a postdoc in the graphics group, presented his work with Dennis Cosgrove of CMU on "Adaptive Unwrapping for Interactive Texture Painting" —work inspired by a desire to paint colors onto his now-famous "Teddy" models.

Andy van Dam. Andy participated in two 60th birthday celebrations: On May 29 he gave the keynote address "User Interfaces: Disappearing, Dissolving, and Evolving" at the Celebration Colloquium for Prof. José Encarñaçao, founder and director of Fraunhofer's Graphics Research Institute and a faculty member at the Technical University of Darmstadt. In October his paper "Reflections on Next-Generation Educational Software" was published in a book honoring the career of Prof. Bernard Levrat, computer science professor at the University of Geneva and president of the Swiss Virtual Campus. In addition, he continued his own celebration of time by hiking the heights and the depths of the earth—the Alps and the Grand Canyon.

Stan Zdonik. Stan was co-presenter at the 27th Very Large Database Conference (VLDB) in Rome of a half-day tutorial on Data Management for Pervasive Computing.

FIVE CS FACULTY AWARDED NSF's ITR GRANTS

The National Science Foundation has announced 309 awards designed to preserve America's position as the world leader of computer science and its applications. CS's five newly funded projects were selected from over 2000 competitive proposals.

Michael Black's grant is for his work on ***"The Computer Science of Biologically Embedded Systems."*** Biologically embedded systems that directly couple artificial computational devices with neural systems are emerging as a new area of information technology research. The physical structure and adaptability of the human brain make these biologically embedded systems quite different from computational systems typically studied in Computer Science.

Fundamentally, biologically embedded systems must make inferences about the behavior of a biological system based on measurements of neural activity that are indirect, ambiguous, and uncertain. Moreover, these systems must adapt to short- and long-term changes in neural activity of the brain. These problems are addressed by a multidisciplinary team in the context of developing a robot arm that is controlled by simultaneous recordings from neurons in the motor cortex of a subject. The goal is to model probabilistically the behavior of these neurons as a function of arm motion and then reconstruct continuous arm trajectories based on the neural activity. To do so, the project will exploit mathematical and computational techniques from computer vision, image processing, and machine learning.

This work will enhance scientific knowledge about how to design and build new types of hybrid human/computer systems, will explore new devices to assist the severely disabled, will address the information technology questions raised by these biologically embedded systems, and will contribute to the understanding of neural coding.

Eugene Charniak's ITR grant is for work on ***"Learning Syntactic/Semantic Information for Parsing."*** The project is so named because the structural information to be learned often falls at the boundary between syntax and semantics. For example, is the fact that "Fred" is typically a person's first name a syntactic or semantic fact? Does the fact that the "New York Stock Exchange" has as part of its name the location "New York" fall under syntax or semantics? What about the similarity between the expression "[to] market useless items" and "the market for useless items"? These are some of the topics that come up in this research.

As for the "for parsing" portion of the title, the intention is to learn the above kinds of information in a form that current statistical parsers can use so that they can output more finely structured parses. However, this is not meant to suggest that parsing is the sole use for this sort of information—exactly the opposite is the case. For example, more and more systems for automatically extracting information from free text use coreference detection and "named-entity recognition" (e.g., recognizing that "New York" is a location but "New York Stock Exchange" is an organization). There is evidence to suggest that both coreference and named-entity recognition can be improved with the finer level of analysis to be made possible by this research. Or again, "language models" (programs that assign a probability to strings in a language) are standard parts of all current speech-recognition systems. There is now evidence that suggests that finer-grained syntactic analysis can improve current language models. Thus this research will enable a wide variety of systems to make better use of language input and thus make these systems more accessible to a diverse user pool.

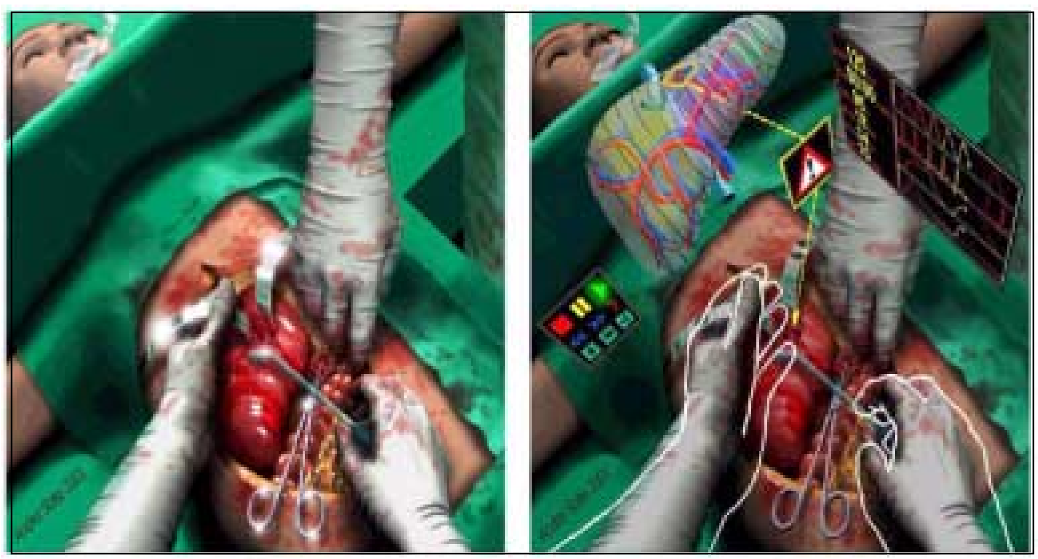
Eli Upfal has been awarded a five-year grant for his work with Harvard's Michael Mitzenmacher on ***"Algorithmic Issues in Large-Scale Dynamic Networks."*** They will develop a theoretically well-founded framework for the design and analysis of algorithms for large-scale dynamic networks, in particular, for the Web and related dynamic networks, such as the underlying Internet topology and Internet-based peer-to-peer ad hoc networks. We plan to develop rigorous mathematical models that capture key characteristics and can make reliable predictions about features such as connectivity, information content, and dynamic of these networks. We plan to apply this framework to test existing algorithms and construct improved new algorithms.

The main benefits of developing the mathematical models of the Web structure and dynamics will be the improved theoretical foundation for the design, analysis and testing of algorithms that operate in the Web environment. The tangible results of this work will be models that can be subjected to experimental verification, analyses of algorithms based upon these models, new algorithms that benefit from these analyses, and, finally, proof-of-concept demonstrations and experimental evaluations of such algorithms.

Pascal Van Hentenryck has been awarded a large ITR grant, over four years, for work in conjunction with ***Eli Upfal*** and researchers at MIT and Georgia Tech on ***"Stochastic combinatorial optimization."*** There are many real optimization problems for which no solutions are known. For instance: a snowstorm is approaching Chicago and United Airlines, at its Operations Center, must plan how to cancel and reroute its flights. Although substantial information on weather forecasts, plane and crew status, passenger itineraries, and hotels is available electronically, this information is not exploited in a scientific way. Instead, humans make ad-hoc decisions based only on their experience. Why is this the case, especially when the airline is a very sophisticated user of optimization for planning? Because the application is a large-scale stochastic combinatorial optimization problem for which no known algorithm produces good solutions in reasonable time. Our fundamental research in this relatively unexplored area will have two complementary thrusts: (1) exploiting the orthogonal strengths of constraint and mathematical programming to tackle the hard combinatorial problems arising in stochastic optimization (e.g., multi-stage or Monte Carlo approaches) and (2) studying stochastic combinatorial substructures that are amenable to efficient solutions or approximations.

Andy van Dam's three-year grant, ***"Electronic books for the tele-immersion age: a new paradigm for teaching surgical procedures,"*** will directly involve researchers from the University of North Carolina, Chapel Hill, as well as Brown and indirectly researchers from UPenn, working with those from UNC on realtime 3D model acquisition and reconstruction. Their work on tele-immersion will provide a dramatic new medium for groups of people remote from one another to work and share experiences in an immersive 3D virtual environment, much as if they were co-located in a shared physical space. Immersive "time machines" will add a further important dimension, that of recording experiences in which a viewer, immersed in a 3D reconstruction, can literally walk through the scene or move backward and forward in time. This work will focus on a societally important and technologically challenging driving application, teaching surgical management of difficult, potentially lethal injuries.

They will develop a new paradigm for teaching surgical procedures: immersive electronic books that in effect blend a time machine with 3D hypermedia. Their goal is to allow surgeons to witness and explore (in time and space) a past surgical procedure as if they were there, with the added benefit of instruction from the original surgeon or another instructor as well as integrated 3D illustrations, annotations, and relevant medical metadata. The trainee should be able freely and naturally to walk around a life-sized, high-fidelity 3D graphical reconstruction of the original time-varying events, pausing or stepping forward and backward in time to satisfy curiosity or allay confusion. The researchers will bring together experts in the respective disciplines and leverage their prior work in tele-immersion in order to achieve these goals.



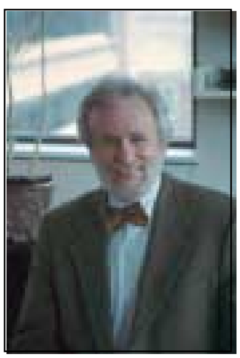
Tele-immersion technology will let surgeons-in-training move naturally within a life-sized, high-fidelity, 3D graphical reconstruction of the surgery, pausing or stepping forward and backward in time to assist in learning

Takeo Igarashi, a postdoc in the Graphics Group here working with John Hughes, hit the worldwide news October 18. The BBC News Online, no less (*http://news.bbc.co.uk/hi/english/sci/tech/newsid_1606000/1606175.stm*), picked up a system he's to present at the ACM UIST conference in Orlando in November. What's more, slashdot.org (*http://slashdot.org/article.pl?sid=01/10/18/1245232&mode=thread*), which calls itself "News for Nerds" and is read, John Bazik says, by "millions of geeks like me," just grabbed the story and featured it; it's now exciting lots of commentary, both insightful and amusing, in slashdot's feedback section.

Takeo's innovative idea is to use simple human sounds, like grunts and sighs, for controlling computers. Conventional voice-recognition software is still not accurate or efficient enough for general use in an interface. Takeo's system works by measuring the pitch and duration of grunt-like sounds like "ah" and "umm." Thus, Takeo suggests, in scrolling through a document on the Internet you could say "move down, ahhhh": the document would continue scrolling as long as the sound continued. You could increase the scrolling speed by raising the pitch of your voice, and stop scrolling altogether by stopping speaking. It's much easier to Undo a command with a quick "uh oh" than with a mouse!

Said Takeo of his system, "I personally do not think this is useful in office environments—it is really annoying for office mates! I basically designed it for isolated situations such as controlling computers while driving a car or rock climbing. I also think the technique is ideal for entertainment applications. I implemented a simple video game using the technique, and children love it." You can find Takeo's paper and demo video at his homepage, *http://www.mtl.t.u.-tokyo.ac.jp/~takeo/research/voice/voice.htm*.

CHARNIAK UNPLUGGED



Eugene Charniak

I was recently invited to participate in a panel discussion with the title "Is it Art Yet?" The main panelist was a composer, David Cope, who has written a computer program that composes music in styles ranging from Bach to Stravinsky that has fooled experts. The question for the panel was captured by the title.

I would say that my performance on the panel was forgettable if I could only remember what I said. Actually, most of the other panelists were only so-so as well. But fortunately our invited guest, Professor Cope, was great. He is a funny, articulate guy, with a combination of down-to-earth personality and controversial opinions that make him ideal for a panel discussion. But what I found most intriguing was the personal story behind the creation of this program.

It seems that in the early eighties Prof. Cope received a commission for an opera. He received (and spent) the advance, but unfortunately suffered a major composer's block. Finally, after a long period of time, he hit upon the idea of creating a computer program to compose for him. So he taught himself Lisp and AI (I was pleased to learn that he used Drew McDermott's and my textbook) but was not happy with the result, which did not sound very much like his music.

He then did what any good scientist does when confronted by a too-difficult problem—he found a simpler one. He figured that while he was not sure what properties made his music sound like "him", he did know the properties that made Bach sound like Bach and Mozart like Mozart. The program he eventually came up with, however, was not restricted to any particular composer. Rather, one gives it a database of musical works, and the program tries to discover the similarities and dissimilarities and then use them to create more music with the same features.

Eventually Professor Cope finished the commission. The work, he says, is about 60% him, 40% computer. However, from his point of view all of the work is his. Indeed, these days it can be hard to tell where one leaves off and the other begins. Now when he composes he gives the program a database of his own work and after composing a bit asks the machine to suggest, say, what the next two measures should be. If he does not like the suggestion he can keep asking for other alternatives.

His major work at the moment is not in his own modern idiom but rather an opera about the life of Mahler, written (with the help of his program) in the style of Mahler. Mahler, besides being a composer, was an active conductor, and although he frequently conducted operas, he never wrote one. David Cope intends to correct this historical mistake.

I teach CS002, our department's computer literacy course. During the semester the students, among other things, learn the basics of HTML, spreadsheets, etc. For the final project in the course the students have the option of a more extensive project in any of the above, or learning a new software package. One of last year's students briefly put her project on the web. She and a friend bought the web address *IllicitBundleOfJoy.com* where one was able to fill out a form specifying the sex, skin color, eye color, price one is willing to pay, etc. However, all you got for your pains is a notice that it is a joke. If she didn't mind getting the FBI after her, it would have been a better joke if she'd asked for MasterCard or Visa.

About 15 years ago or so, during the early spring, the tabloids had a field day when it was discovered that several Brown students (female) were involved in a prostitution ring. "Ivy League Madam!" or some such. Brown immediately went into full-court damage control, with multiple press conferences all saying that what students did on their own time was beyond our control, that this was not exactly what we thought the life of the mind was all about, and that we did prefer students without criminal records.

No, Suzi Howe did not dress to match the tablecloths on IPP Symposium day, but everyone who stepped off the elevator thought she had!

At the time I thought that Brown was overreacting, but now I am not so sure. A key fact in the above is that it occurred in early spring—the time when students (and their parents) are deciding where to go to college. I have reconsidered because this last spring I was one such parent. In particular, one of the schools my son was considering was Hamilton, a small, selective, liberal arts college in upstate New York. In early spring the *New York Times* published an article about cloning humans prominent in which was one Professor Boisselier, identified as a professor in the chemistry department at Hamilton. Worse, the organization whose cloning laboratory she runs, the Raelians, is headed by a fellow named Rael, who was abducted by aliens and seduced by female alien robots, and who believes on the basis of this that the destiny of the human race is to make itself perfect and immortal through the use of science.

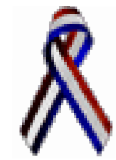
After some investigation it turned out she was a visiting faculty member, she resigned from her position, my son decided to go to Colby, and he thought the whole thing was a hoot. But I mentioned this to a neighbor who is the spokesperson for the Rhode Island School of Design. She responded that a few years ago, again in the spring, several RISD students were arrested by the Boston police after nearly finishing painting an entrance to an MBTA station pink. A picture appeared on the front page of the *Boston Globe* with the headline 'Caught Pink-Handed'. When the students were asked what they were doing they responded that this was their 'site-specific' art work for one of their RISD courses. Naturally, this fell in my neighbor's lap. A reporter with whom she had dealt before called to ask for RISD's position. She responded that RISD did not have a position, but would he take a comment off the record? Off the record, RISD's position was 'What shade of pink?'

Of the more than 5,000 feared dead as a result of the terrorist attacks on September 11, at least six were Brown alumni.

As far as we know, none of them had connections with this department. Members of the Brown University community paused during three events in the October 12-14 celebration of President Ruth Simmons' inauguration to remember and honor these alumni.

For details and a list of those lost, see:

http://www.brown.edu/Administration/News_Bureau/2001-02/01-041.html



conduit!

A publication of
The Computer Science Department
Brown University

Inquiries to: ***conduit!***
Department of Computer Science
Box 1910, Brown University
Providence, RI 02912
FAX: 401-863-7657
PHONE: 401-863-7610
EMAIL: sjh@cs.brown.edu
WWW: http://www.cs.brown.edu/publications/conduit/

Suzi Howe
Editor-in-Chief

Katrina Avery
Editor

Jeff Coady
Technical Support

Department of Computer Science
Brown University
Box 1910, Providence, RI 02912, USA

conduit!

NON-PROFIT
U.S. Postage
PAID
Providence, RI
Permit #202

Printed on recyled paper

Address changes welcomed

conduit!

A publication of
The Computer Science Department
Brown University

Inquiries to: ***conduit!***
Department of Computer Science
Box 1910, Brown University
Providence, RI 02912
FAX: 401-863-7657
PHONE: 401-863-7610
EMAIL: sjh@cs.brown.edu
WWW: http://www.cs.brown.edu/publications/conduit/

Department of Computer Science
Brown University
Box 1910, Providence, RI 02912, USA

conduit!

NON-PROFIT
U.S. Postage
PAID
Providence, RI
Permit #202

Printed on recyled paper

Address changes welcomed