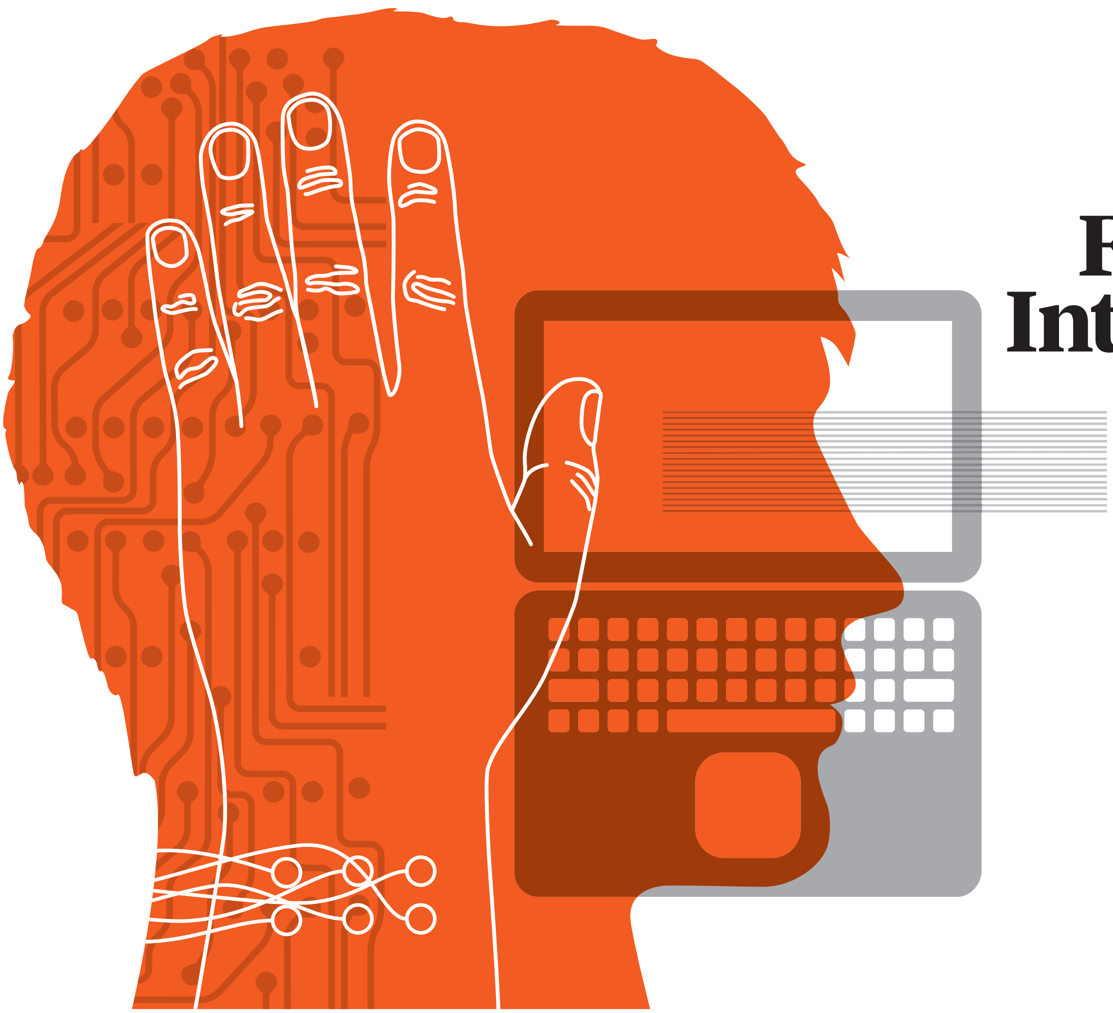
Crossroads The ACM Magazine for Students

SUMMER 2010 ISSUE 16.4

XRDS



The Future of Interaction

Profile: Hiroshi Ishii Looks for What's Tangible

Computers that Connect Directly to the Brain!

Pen-Based Computing: Old Dog, New Tricks

XRDS.ACM.ORG

Association for Computing Machinery

/STUDENT

Microsoft.com/Student is the digital center for students wanting to see what Microsoft has to offer.

www.microsoft.com/student

MAKE SCHOOLWORK LESS WORK

The Imagine Cup is the world's premier student technology competition. While competing for cash and prizes, students gain real life experiences, make new friends, and turn their ideas into reality. The brightest young minds join together and, using technology, take on the toughest problems facing our world today.

www.imaginecup.com

THERE'S ALWAYS TIME FOR FUN

Microsoft Student Rally is a digital competition where students share their creativity, learn new stuff, have fun and win some great prizes.

www.microsoftstudentrally.com

DreamSpark is simple; it's all about giving students Microsoft professional-level developer and designer tools at no charge so you can chase your dreams and create the next big breakthrough in technology – or just get a head start on your career.

www.dreamspark.com

YOUR SKILLS YOUR FUTURE

Want to work in IT? Take advantage of the Microsoft Students to Business (S2B) program to help you find your dream job at one of the company's 395,000 partners. Follow the Career Channel to learn which positions are in demand before choosing your career path in the Student Career Portal. Let S2B help you land your dream job!

www.microsoft.com/studentstobusiness

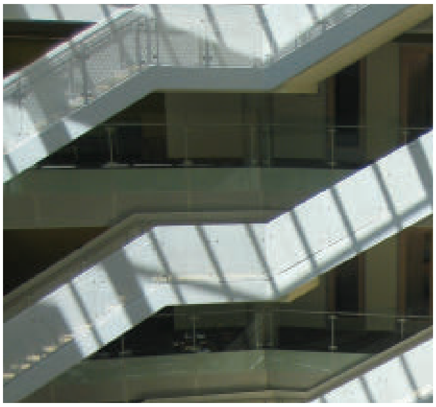
LEARN WORK PLAY

Microsoft

XRDS

Crossroads
The ACM Magazine for Students
SUMMER 2010 VOL.16 · NO.4

The Future of Interaction



begin

features

end

Association for Computing Machinery

XRDS

Association for Computing Machinery

ACM
Association for Computing Machinery
2 Penn Plaza, Suite 701
New York, NY 10121-0701 USA
+1 212-869-7440

CONTACT
General feedback: xrds@acm.org

For submission guidelines, please see http://xrds.acm.org

Letters to the Editor: xrds@acm.org

Connect with us on Facebook: http://tinyurl.com/XRDS-Facebook

Comment via Twitter: #xrds

SUBSCRIBE
Subscriptions ($19 per year) are available by becoming an ACM Student Member www.acm.org/membership/student

Non-member subscriptions: $70 per year http://store.acm.org/acmstore

ACM Member Services
P.O. Box 11450
New York, NY 10286-1405 USA
+1 800-342-6626
+1 212-626-0500

Postal Information
Application to mail at Periodical Postage rates is pending at New York, NY and additional mailing offices.

Offering# XRDS0163
ISSN# 1528-4981 (print)
ISSN# 1528-4982 (electronic)

25TH ANNUAL ACM CONFERENCE ON SYSTEMS, PROGRAMMING, LANGUAGES, APPLICATIONS: SOFTWARE FOR HUMANITY

MARCH 25, 2010
Submission deadline for
OOPSLA Research Papers,
Practitioner Reports,
Educators' and Trainers' Symposium,
and proposals for Tutorials,
Workshops, Panels

APRIL 23, 2010
Onward! Papers and Essays

JUNE 24, 2010
Submission deadline for Posters,
Demonstrations, Doctoral Symposium,
Onward! Films, and Student Research
Competition and Volunteers

LOCATION
John Ascuaga's Nugget Hotel
Reno/Tahoe Nevada USA

COLOCATED CONFERENCES
Onward!
Dynamic Language Symposium (DLS)
Pattern Languages of Programs (PLoP)
and more

CONFERENCE CHAIR
William R. Cook, UT Austin
chair@splashcon.org

OOPSLA PROGRAM CHAIR
Martin Rinard, MIT
program@splashcon.org

For information, please contact
ACM Member Services Department
1-800-342-6626 (US & Canada)
+1-212-626-0500 (global)
info@splashcon.org

SPLASH/OOPSLA is sponsored by
ACM SIGPLAN and SIGSOFT

WWW.SPLASHCON.ORG

ACM SIGPLAN/SIGSOFT
announce
OOPSLA
is now part of
SPLASH

SPLASH

RENO 2010
OCTOBER 17-21

LETTER FROM THE EDITOR

The Future of Interaction
Interfaces Everywhere

We've been a bit preoccupied with interfaces as of late. For one thing, you're holding the launch issue of *XRDS*, ACM's magazine for interfacing with the student population. In many ways, this is a brand-new publication—new format, new content, and new vision. But it's also an evolution of *Crossroads*, and as such, it's staying true to a 15-year legacy of student-centric content.

Introducing XRDS

You've probably already noticed we've donned a fresh new look, created by world-renowned design firm Pentagram and the dedicated staff at ACM headquarters. However, you'll quickly discover cosmetic changes are only a small part of this redesign. You might have also noticed we've put on a bit of weight. On the next 50 or so pages, you'll discover a dozen new columns, things like "Advice" (page 7), a tutorial called "Hello World" (page 50), "Labz" (page 52), and much more.

These columns, headed up by a dedicated group of departments editors (see masthead on page 2), were carefully selected and designed to get useful information into your hands and help you connect with the organizations, opportunities, and other students that matter most to you. These columns will be recurring, so you will be able to jump right to the information you find most useful in every issue.

Our goal is to make *XRDS* the premier news and information platform for students interested in computer science, computer engineering, information systems, interaction design, and similar fields. This is one of ACM's chief missions, and we intend for this magazine to be a bold step toward fulfilling that promise.

Interested in helping us realize this vision? *XRDS* is not only a magazine for students, but also run by students. That means we need you! If digging up leads for feature articles, writing columns, or reporting from conference floors sounds exciting, we want to hear from you.

> **"No longer do we think of computing solely as sitting in front of a desktop computer with a keyboard and mouse. Computing occurs in cars, while we're walking or riding the subway…"**
> *Chris Harrison*

Email us (xrds@acm.org). Join our discussions on our Facebook group page (http://tinyurl.com/XRDS-Facebook). Chatter with us via Twitter by using "#xrds" in any tweet.

Interfaces for Input

We've decided to kick off *XRDS* with an issue dedicated to a highly relevant and rapidly evolving subject: interfaces for input—where they are now, and where they'll be going soon.

Advances in electronics, both in computational power and reduced cost, have allowed computers to pervade almost all aspects of our lives. No longer do we think of computing solely as sitting in front of a desktop computer with a keyboard and mouse. Computing occurs in cars, while we're walking or riding the subway, at a kiosk in the airport, and even on interactive tabletops and walls. To fully unleash the true potential of these computing modalities, researchers are developing new ways for us to get information into these rich platforms—what we generally refer to as input.

This topic is particularly close to my heart and forms the core of my present PhD research. I think about ways to enable (small) mobile devices to "steal" (large) everyday surfaces for input. Consider, for example, a cell phone sitting on a desk. Why reach over and press some diminutive button to silence an incoming call when you could simply issue a finger gesture right on the (large) table in front of you? Or imagine a music player strapped to your upper arm while out for a jog. Why reach over to interact with some tiny scroll wheel when you could use the nearly two square feet of skin surface area on your lower arm for finger input?

It's All Happening Now

That might sound like science fiction, but these projects have already been published at UIST (the Symposium on User Interface Software and Technology) and the annual SIGCHI conference (or the Special Interest Group on Computer-Human Interaction), two premier ACM conferences that you, as an ACM student member, get discounted entrance to, by the way.

That's just the tip of the input iceberg. We've got six feature articles from top researchers covering everything from tangible tabletops and pen input, to micro-device interactions, and brain-computer interfaces. Intrigued? Keep reading…

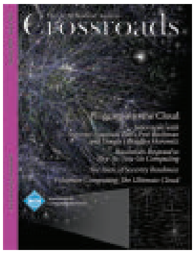
Chris Harrison

Chris Harrison is a PhD student in the Human Computer Interaction Institute at Carnegie Mellon University. He is a Microsoft Research PhD Fellowship recipient, and has worked as several industry labs, including IBM Research, AT&T Labs, and Microsoft Research. More about his background and work is available at www.chrisharrison.net.

Cloud Computing

Although I am by no means an expert in the field, I read David Chiu's article "Elasticity in the Cloud," (issue 16.3, Spring 2010) and found it provocative and topical. Cloud computing seems to be the dominant paradigm of computing today and the elasticity that Mr. Chiu writes about is dead on. Keep up the good work!
Clint Benjamin, *Email*

The article "Cloud Computing in Plain English" (issue 16.3) was a great introduction to the topic, and the references were very interesting overall. I think that the pervasiveness of cloud computing now and the enormous impact it has on our lives will force us to rethink how we use it going forward and to find the sweet spot between local and cloud computing that we haven't quite figured out yet.
Hesham Wahba,
Email



I liked the issue on clouds, especially the article on research perspectives ("Clouds at the Crossroads: Research Perspectives," issue 16.3). I would like to see more specific issues being addressed, e.g., what challenges come up in offering database as a service (e.g., Amazon S3) over the cloud, and the scope of data aggregation in distributed databases over the cloud.
Bibudh Lahiri,
Iowa State University, U.S., Facebook

I was really excited to see this issue (issue 16.3)! Haven't had a chance to read it yet... but it's on the agenda for this week. :) Am having to design a data center for a networks class I am in and the "cloud" is part of that design so this was very timely for me. Thanks for putting this one together.
Mary Melloy,
Facebook

The security issue of cloud computing ("State of Security Readiness," issue 16.3) raises the most concern for myself but not as described in the article. If cloud computing grows as predicted, what is in place from preventing the companies controlling the cloud from changing policies or charging new fees? Once you're in the cloud, how do you get out?
Teo Fernandez,
Email

Reading *ACM Crossroads*, it's crazy to think how much of the computing we do is in the cloud now, compared to, say four years ago. #xrds
Jason(jwiese),
Pittsburgh, Twitter

MapReduce

We introduced cloud computing and MapReduce into our undergrad database programming course this year. After the initial conceptual hurdle, the students became very comfortable applying these techniques to computing problems, and we're optimistic that programming for the cloud will part of the toolkit our students graduate with.
Shaun Kane,
University of Washington, U.S., Email

Once you're in the cloud, how do you get out?

Teo Fernandez

Future Issues

I'd be glad to see a separate issue on "opinion mining/ sentiment analysis." Opinion mining/sentiment analysis is a challenging text mining and natural language processing problem. The main aim of sentiment analysis is to discover opinions embedded in social media, news, and other online avenues, and hence hear the true voice of customers. Due to its great business value, opinion mining has received a great amount of interest in the past decade. There has been an explosion in research, both in academia and industry, and various firms like Lexalytics currently offer opinion-mining services.
Denzil Correa,
Facebook

XRDS, International

Awesome! chegou minha *ACM Crossroads*. Achei que nem fosse vir :)
Translation: Awesome! My *ACM Crossroads* arrived. I thought it was not going to come :)
Henrique Pereira (ikkebr),
Santa Mari, Brazil, Twitter

ImagineCupの広告がある *ACM Crossroads* Spring 2010
Translation: There is an announcement for ImageineCup in *ACM Crossroads* Spring 2010.
A graduate student (naka4),
Japan, Twitter

How to contact XRDS Send a letter to the editors, an amazing photo that highlights your work or something interesting on your campus, or other feedback by email (xrds@acm.org), on Facebook by posting on our group page (http://tinyurl.com/XRDS-Facebook), via Twitter by using #xrds in any message, or by post to ACM Attn: XRDS, 2 Penn Plaza, Suite 701, New York, New York 10121, U.S.

PHOTO SPOTLIGHT

Students at the 33rd ACM ICPC Asia Hefei Regional competition in China.

Photo by Jim Grisanzio (www.flickr.com/jimgris)

begin

UPDATES

About the Redesign
The Path to the New XRDS

Two short years ago, ACM members saw *Communications of the ACM* undergo a remarkable transformation, in both editorial content and artwork.

A few months ago, a major reshuffle took place at *Crossroads*, too. The goal was to reshape the editorial team into a diverse, interconnected, and highly energized crew who could reignite this magazine and push it screaming forward into the new decade.

In November 2009, a few *Crossroads* editors, including myself, assembled at ACM headquarters in New York, with Jill Duffy, senior editor at ACM and this magazine's new managing editor, and Scott E. Delman, group director for ACM's publications department, to completely revamp the magazine. We spent days analyzing *Crossroads*' strengths and weaknesses, while thinking about how we wanted to change it.

Looking through early issues, which date back to 1994, we saw that the magazine's founders assigned a theme to each issue, a topic that was of the utmost interest to computer science students, and which also gave the magazine a sense of cohesion. Step number one in our redesign effort was to return to that founding vision.

(l-r) Luke Hayman, Ryan K. L. Ko, Tom Bartindale, Chris Harrison, Scott Delman, and James Stanier.

As we incorporated this change, we realized a more important one was taking place just beneath the surface. We were moving away from "student journal" and toward "the ACM magazine for students." It's a subtle distinction to some, but this publication is for you and should be exciting for you to read.

While *Crossroads* will continue to accept student-submitted articles, the editorial team now also invites feature articles by inspiring people in computer science, written especially with students in mind. We hope that they will spark you to enroll in a new course or grad program, or seek out a fresh path in your career.

Despite the huge revamp of content, probably the most noticeable change is the design. While in New York, we met with Luke Hayman from Pentagram design firm. We told him we wanted *Crossroads* to feel inviting, contemporary, and young, and for articles printed on these pages to look beautiful. After several weeks, Hayman and his colleague Rami Moghadam expertly put into place the design you see before you.

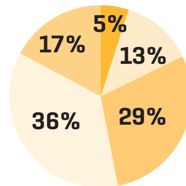
A key motivator in choosing the final look and feel was finding something that would make you, the lifeblood of this magazine, want to be involved. This is your publication, and we want you to help shape its future.

We want to know what you think. Talk to us via our Facebook group (http://tinyURL/XRDS-Facebook), post on Twitter using #xrds, or email us directly at xrds@acm.org.

You may have noticed that the name has changed as well. How did I forget? This isn't *Crossroads* anymore. Welcome to *XRDS*. —*James Stanier*

1980

David Bradley of IBM invents "Control-Alternate-Delete"



iPhone Users by Age

13-17	5%
18-24	13%
25-34	29%
35-54	36%
55+	17%

136

Years since the QWERTY keyboard layout was invented.

BENEFIT

Microsoft Developer Academic Alliance

If you're a student ACM member, there are dozens of perks at your fingertips: free courses, serious discounts, and exclusive opportunities that *XRDS* doesn't want you to miss.

One of the most exciting benefits is the Microsoft Developer Academic Alliance. It's an agreement between Microsoft and the ACM to offer you free (repeat: *free*) software.

The Alliance offers development software, such as Visual Studio 2008 Professional, as well as Microsoft Project, Visual C++2005 Express Edition, and several other software packages. Coders will love this benefit, especially those using Microsoft platforms. Who wants to shell out for the latest IDE when they can download it for free? Some of the offerings are niche (heard of Office Groove? Anyone?) but seeing as it's free, you can experiment without spending recklessly.

The download system is straightforward and gives you an installation key, allowing the software to be installed on multiple machines.

The Academic Alliance belongs to a bigger scheme called the Student Academic Initiative Program, which includes some offerings from Sun and CA, too. See http://www.acm.org/membership/membership/student/sai_general/. —*Daniel Gooch*

ADVICE

Marketing Your Ideas
Don't Sell Yourself Short

You found a new algorithm? Why should we care? Somehow, you must attract our attention.

Richard Hamming, one of the founders of ACM, said it eloquently: "The world is supposed to be waiting, and when you do something great, they should rush out and welcome it. But the fact is everyone is busy with their own work."

Marketing may seem like a dirty word to engineers and scientists, but it is a necessary evil.

TAKE YOUR TIME

Young scientists tend to rush their presentations. They work four months to a year on a project, yet they wait until the last minute before writing their paper and rehearsing their presentation—when they rehearse it at all.

We all have seen Steve Jobs, the CEO of Apple, show up for a presentation in jeans, without slides or special effects. But consider how Jobs has a clear theme and a precise outline. He never rambles. He has smooth transitions from topic to topic. His talks only appear laid back. In fact, they are precisely choreographed and thoroughly rehearsed.

What about reports and research papers? Rushing their publication is trading quality for quantity. It is an unfortunate trade, as there is a glut of poor research papers, and too few high quality ones. Continuous writing, editing, and rehearsal should be an integral part of your activities.

"Do not underestimate email. It is the most powerful medium at your disposal. Yet, you have to use it wisely."

REACH OUT TO YOUR AUDIENCE

Scientists and engineers are most successful when their work is most available. Torvalds and Berners-Lee initiated Linux and the Web, respectively, by emails sent to mailing lists. Perelman finished proving the Poincaré conjecture by posting eprints on arXiv, an open electronic archive.

But posting your content and giving talks is hardly enough. You have to use good marketing. If you want people to attend your talks, make sure your title tells them why they should attend. Think about your audience. They want to know whether they should continue reading your paper or come to your talk. Convince them that you have something remarkable to tell them. Avoid jargon, acronyms, and long sentences.

Do not underestimate email. It is the most powerful medium at your disposal. Yet, you have to use it wisely. To get famous people to read your emails, study their work. Show appreciation for their results. Think of reasons why they might find your question or proposal interesting.

For more advice, be sure to read or listen to Richard Hamming's 1986 talk "You and Your Research." The transcript and audio file are online.

Biography

Daniel Lemire is a professor of computer science at the University of Quebec at Montreal (UQAM). He has a blog at daniel-lemire.com, where he sells his ideas every week.

In Search of a Natural Gesture

While computing has advanced exponentially, almost explosively, since the 1970s, input devices have only just begun to change. Why?

By Johnny Chung Lee

DOI: 10.1145/1764848.176853

In many articles discussing the future of computing, you are very likely to find either a reference to, or a motivational assumption based on, a continued projection of Moore's law. This article will make no attempt to deviate from that steady tradition.

The regularity of references to "the law" probably extends from the fact that human behavior and clever uses for technology are notoriously difficult to predict, but the technology itself has demonstrated an unwavering trend over the past four decades. This trend is, of course, enabled only by the astonishing accomplishments by the engineering teams within the companies that manufacture computing equipment. Nevertheless, it doesn't take a huge amount of clairvoyance or risk-taking to claim that the trend will extend a bit further.

However, interface technology has not enjoyed the seven orders of magnitude in improvement of performance that core processors have achieved since 1970. In fact, aside from a slightly improved mechanical construction and visual polish, the input and output devices connected to the average desktop computer today are virtually identical to the ones used by Douglas Engelbart in his 1968 presentation, later referred to as "The Mother of All Demos." While there have certainly been several improvements along the way, such as the graphical user interface, trackpads, flat-panel displays, and touch screens, we still fundamentally operate our computers using a single 2D pointing device and a keyboard.

Yet in just the past two or three years, it is not too difficult to find articles proclaiming the "death" of the mouse and keyboard, or finding new product announcements promoting new methods of input and output as its key selling feature. Why has there been a recent spike in enthusiasm for new interface technology? In my opinion, it is because we've recently crossed the inflection point—an inflection point driven by Moore's Law and the limited growth of human attention.

CONSUMPTION-PRODUCTION IMBALANCE

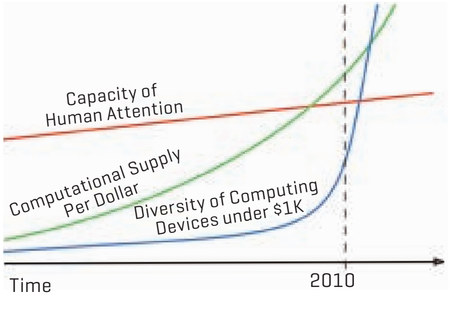
Over the past hundred years, the cognitive capacity for an individual to consume and produce information has stayed relatively constant or has increased only modestly. While technology has certainly made it much easier to saturate our input and output channels, the rate at which we can read, write, speak, listen, the density of pixels our visual system can resolve, the number of images we can meaningfully process per second, and the size of our fingers has not significantly changed. Yet the capacity for technology to supply us with content has grown in step with Moore's Law.

In the 1980s and 1990s, the consumer appetite for faster technology could not be satiated. But in recent years, the information supply has started to fundamentally surpass the ability of many people to absorb it. This creates a situation of computational surplus, which, economically, should dictate a substantial drop in cost.

Just a few years ago, a $100 laptop was considered a magnificent dream that would change the world. Now, it is possible to find a reasonably capable netbook for $100 on a good coupon day or even "free" with a network service contract. While it would have certainly been possible to manufacture $100 worth of computation in 1990, very few people would have found it satisfactory. That's not the case today. The average consumer's demand for more powerful technology has simply not kept up with the exponentially increasing supply.

Some have referred to this stall in performance demand as the era of "good enough computing." While "good enough" might suggest even further reduction in device cost, what's happening instead is it's becoming economically sensible to manufacture a wider variety of increasingly special purpose computers rather than expensive general purpose machines. For the price of a nice dinner, consumers can buy a computer that only plays music, only takes pictures, only shows maps, only plays games, only plays movies, or only lets you read the news. It's likely

Figure 1: The rise of diversification was a result of a computational surplus.



that we'll see a significant rise in the release of new form factors and targeted niche computing compared to what we have in the past. (**See Figure 1.**)

How is this relevant to interface technology?

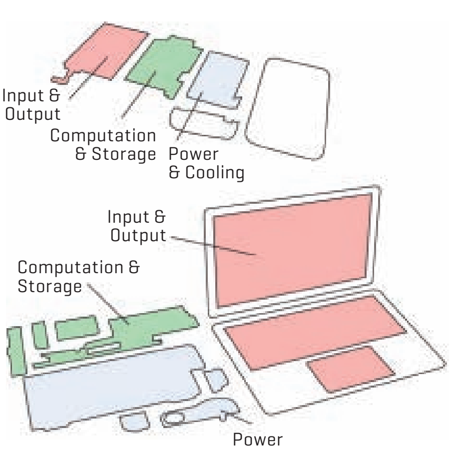
TASK-SPECIFIC DEVICES

As the diversity and specialization of devices increases, so does the diversity of interface technology. The best interfaces are typically task-specific. For example, the ideal interface for choosing a radio station while driving your car is not the same as the best interface for checking your email while sitting at your desk. As the distribution of computing shifts away from usage scenarios where a mouse and keyboard are acceptable, so does the adoption of alternative input methods, whether touch, motion control, location, gesture, voice, or some other physiological source.

If you look at the components within a modern laptop or mobile phone, you'll notice that there's actually very little "computer" in a computer today (**Figure 2**). The largest internal components of a modern laptop are already those dedicated to human input and output. As the physical space required for computation continues to fall or is even replaced with a high-speed network connection, the defining feature of the device and its suitable applications is the interface technology.

As a result, there is a very high demand in exploring novel ways of interacting with technology that permits alternative form factors, increases our capacity to express an idea, or improves our ability to absorb information. Computing will be defined by how we interact with the information rather than by the chipsets on the motherboard, or the operating system it runs. The quest to create new devices dedicated to solving each of our own specialized unsatisfied desires is largely lead by the search for better interface technology that can better understand what we want, when we want it, where we want it, in the way we want it.

Figure 2: The human interface hardware now dominates the form factor of many modern computing devices.



> **"Natural interaction is achieved through clever designs that constrain the problem in ways that are transparent to the user."**

IN SEARCH OF NATURE

A phrase that has slowly received increasing traction, at least in the commercial exploration of alternative input technologies, is "natural user interface" (NUI). While there's no widespread consensus about the exact definition, NUI generally refers to an interface that is highly intuitive and effectively becomes invisible to the user when performing a task. It is an interface that can easily and efficiently transmit an idea from a user's mind into an action on the computer with little additional effort.

Don Norman described the philosophy well when he said, "The real problem with the interface is that it is an interface. Interfaces get in the way. I don't want to focus my energies on an interface. I want to focus on the job."

Unfortunately, the term NUI has also been coarsely applied to refer to anything that is not a typical keyboard and mouse. It's important to acknowledge that the philosophy behind a natural user interface is not conceptually incompatible with a mouse and keyboard. However, it has become much more popular to use the term "natural" when referring to multi-touch interaction, motion sensing, gesture input, and speech recognition.

These input techniques certainly offer a higher potential for expressing an idea to a computer with less distortion and rigid structure typically required by a mouse and keyboard. However, gesture and speech interfaces, in particular, have resonated well with the popular imagination. The allure of these methods of input is that they provide a glimpse into an easy-to-imagine vision of one day being able to communicate with a computer as easily and fluidly as we communicate with another human being using these skills we practice every day.

Now, it's debatable whether communicating with a computer in the same manner that we communicate with other humans is truly the most desirable interface for all tasks. To get a reasonably accurate picture of what a voice-and-gesture-only system might be like, imagine if the only input control to your computer were a video chat to a high school student sitting in a distant room, and all you could do is describe what you wanted. After a few minutes of saying, "Click on that. Move that … no, not that one. The other window. I mean the browser window. Yeah. Make that bigger, I mean maximize it,"you will probably say, "where is my mouse and keyboard?"

An often unspecified detail of that vision is that the computer should be the embodiment of an exceptionally competent and omniscient human being with a reasonable personality that makes no recognition errors. But, for the sake of this article, I will concede there are components of that vision that are conceptually desirable enhancements to existing interface technologies and discuss some of its advantages and disadvantages. In particular, this article will discuss the gesture component in greater detail.

BODY MOVING

Body movements used to convey information from one person to another have been shown to be tightly coupled with simultaneous speech or co-verbal commands. According to researcher

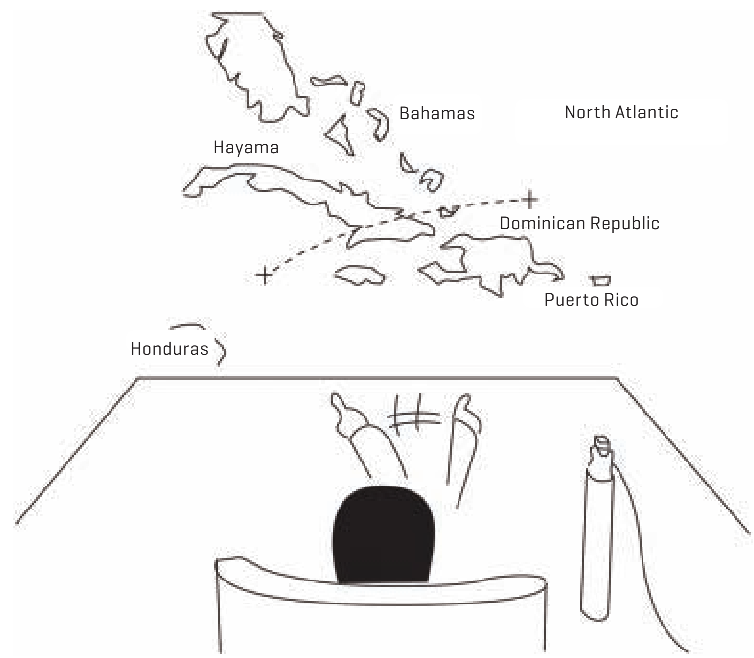
David McNeill's 1992 study, 90 percent of these types of communicative gestures are found to be associated with spoken language. Additionally, gestures often identify underlying reasoning processes that the speaker did not or could not articulate providing a complementary data source for interpreting a set of utterances.

Thus, gesture and speech go hand-in-hand in daily human-to-human communication, and it would be appropriate for any interactive system that attempts to provide a similar level of fluidity to be designed with that in mind.

Such systems that combine more than one mode of input are often called multimodal interfaces. A well known example demonstrating the power of combining speech and gesture is the Put-That-There system created by Richard A. Bolt in 1980 shown in **Figure 3**. This system allowed the operator to sit comfortably in a chair, point his arm at a distant location on a large display wall and issue verbal commands such as "move that" and then pointing at a different location, continue the command "... there."

The gesture provided object focus and task parameters, and the speech component provided the task selection and event execution. These modalities complement each other's strengths, combining the physical specificity of pointing with the random access temporal nature of speech.

Figure 3: Richard Bolt's Put-That-There system combined speech and gesture input.



GESTURE CHALLENGES

While a number of prototype systems that used gesture alone have been demonstrated to be reasonably functional, many of these systems typically relied on a unique set of hand or body poses that must be learned by the user, which trigger a small number of pre-assigned actions.

In 1986, Jean-Luc Nespoulous identified three classes of communicative gestures that have come into common use: mimetic, deictic, and arbitrary. Mimetic gestures are motions that are intended to be representative of an object's shape or behavior. For example, indicating the shape of a person's beard, the size of a box, or the act of tumbling. Deictic gestures are used to provide context or explanatory information such as pointing at the object of conversation, or indicating the direction for an action to be taken. Arbitrary gestures are learned motions typically used in specific communication settings, such as the hand signals used in airplane guidance, baseball pitches, or infantry coordination.

In the context of a gestural interface prototype, arbitrary gestures are highly popular choices in research systems because they can be easily designed to be distinctive for the sake of recognition and segmentation. But, these gesture sets tend to require significant user training, and they map to a rigid set of commands.

In general, performing complex communicative or manipulation tasks using free-air gestures alone without tactile feedback or co-verbal commands is actually quite unnatural. However, there may be opportunities to take advantage of the expressive power of deictic and mimetic gestures to augment or supplement interaction tasks because users will have a tendency to produce these gestures without additional prompting or training. Unfortunately, these gestures are not typically easy to segment and are subject to high variability between individuals.

In 1989, Alex G. Hauptmann attempted to study the degree of consistency and variability in unprompted gestures when users attempted to perform a three-dimensional spatial manipulation task. The users were asked to try to perform a translation, rotation, or scaling operation on a virtual wireframe cube rendered on a computer display. Upon completion, a human operator observing the hand gesture would attempt to simulate the resulting output. While there were coarse similarities in the type of gesture performed for each of the three tasks, individuals varied significantly in the number of fingers used, the position and orientation of their hands, the number of hands used, and the alignment of the hand movements.

Hauptmann made no attempt to make the computing system recognize and track these gestures as real interactive controls, which weakens the validity of certain conclusions as interactive feedback would certainly impact user behavior. However, the findings do indicate that a fully functional system would have to accommodate a high degree of variability between users.

Bolt attempted a partial implementation in 1992, but this system only provided constrained object rotations and single-handed object placement in a limited application scenario using two six-degree-of-freedom-tracked gloves with articulated fingers and heavily relied on co-verbal input for action selection and control. Furthermore, the variations Hauptmann observed occurred in the constrained scenario where people were seated within 1 meter of a computer display and were prompted to perform a simple spatial operation on a single object. As the assumptions are pulled back on this problem, the opportunity for variation goes up exponentially, such as allowing multiple objects on the screen simultaneously, using non-spatial actions such as changing the object color, varying the seated posture relative to the screen or standing at different distances, allowing multiple users to attempt simultaneous control, and even choosing to perform other peripheral tasks within the tracking volume without activating the system. Variations in body shape, size, and cultural background only exacerbate the difficultly in interpreting a given gesture, or finding a common gesture for a given desired action.

The complexity of a gesture recognition system is roughly proportional to the complexity of the input vocabulary. For example, if all that is desired is either motion or non-motion, there are a variety of sensors such as accelerometers that can provide a simple data stream that is relatively easy to examine to obtain this signal. In the case of an accelerometer, other data such as the magnitude and direction of motion, or the static orientation relative to gravity are moderately easy to extract. However, as the desired expressiveness of the input system goes up, so must the complexity of the gesture system.

In effect, the device must have an understanding of the world that is not only capable of distinguishing the target input set, but all other similar gestures, in order to recognize that they are not part of the input set. Otherwise, the number of false positives may be unacceptable. If the goal is to recognize a "jump," simply looking for vertical movement would be insufficient if a "squat" should not be considered a "jump." Should sitting down and then standing up be considered a jump? What about walking? Is a one legged-kick a jump? What about individuals who jump at different heights?

"Computing will be defined by how we interact with the information rather than by the chipsets on the motherboard, or the OS."

GUIDING GESTURES

In this respect, freeform gesture recognition shares many of the difficulties of unstructured speech recognition. Many spoken words have very similar acoustic properties. People speak with different accents and dialects. There are multiple ways of expressing the same thought. Recognizing isolated words without additional context information is generally unreliable. Recognizing speech in the presence of other noise can significantly reduce accuracy. Identifying when the user wants to engage and disengage with the system can be challenging. Without the speech equivalent of push-to-talk, prompted input, or escape keywords, gesture interaction suffers from the "Midas touch" problem of excessive accidental or false activations.

Alternatively, naively applying rigid structure and increasing recognition requirements would negate any potential benefit from user intuition and simply replace it with frustration from excessive false negatives. Understanding the strengths and weakness of a particular input method is fundamental to understanding what combination of tools will make for a successful user experience. The design should provide just enough guidance using other techniques to prevent the user from falling into the poor performing areas of gesture and speech recognition. If done correctly, a relatively small amount of recognition work can provide a delightful experience giving the illusion that the technology has merely understood their intention.

For guidance toward solutions to this problem, it's helpful to revisit the philosophies behind "direct manipulation" that made the graphical user interface successful, as described by Ben Shneiderman in 1983. One of the tenants of the design was that displaying immediate visual feedback for input is essential. This is the communicative common ground between yourself and the device that indicates it has an understanding of what you are doing, and that you understand what it is doing in response to your actions. The number of degrees of freedom that can be captured in the feedback should be as high as reasonably possible, giving users the information they need to quickly adjust their input to accomplish the desired output.

Interactive objects should have a visual representation with understandable metaphors. The visual appearance of an interactive object should provide some affordance as to the actions or gestures to which it can respond. The interface should clearly provide rapidly accessible, complimentary, and reversible commands.

NATURAL IS IN THE DESIGN

Regardless of the technology being used, a good interface experience is one that is able to capture the intent of a user's behavior with as little distortion as possible. While gesture and speech technologies offer greater potential for us to express our thoughts and ideas without thinking about the constraints of the interface, accurately reconstructing those ideas within the computer does not come from technology for free. Natural interaction is achieved through clever designs that constrain the problem in ways that are transparent to the user but fall within the capabilities of technology. A good user experience is achieved only through the hard work of individuals with domain expertise and empathy for those who are not like themselves.

Biography

Johnny Chung Lee is a researcher in Microsoft's Applied Sciences Group exploring interface novel software and hardware technology. In 2008, he graduated from Carnegie Mellon University with a PhD in human-computer interaction and was named into *MIT Technology Review*'s TR35. He is known for his video demonstrations of alternative applications for the Nintendo Wii remote that have received more than 10 million views.

Pen-Based Computing

Pens may seem old-fashioned, but some researchers think they are the future of interaction. Can they teach this old dog some new tricks?

By Gordon Kurtenbach

DOI: 10.1145/1764848.1764854

When I entered graduate school in 1986, I remember reading about the idea of using a pen as an input device to a computer. Little did I know that the idea had been around for long time, from the very early days of modern computing. Visionaries like Vannevar Bush in his famous 1945 article "As We May Think" and Ivan Sutherland's SketchPad system from the early 1960s saw the potential of adapting the flexibility of writing and drawing on paper to computers.

The heart of this vision was that the pen would remove the requirement for typing skills in order to operate a computer. Instead of typing, a user would simply write or draw, and the computer would recognize and act upon this input. The rationale was that by supporting this "natural" expression, computing would be accessible to everyone, usable in broad range of tasks from grandmothers entering recipes, to mathematicians solving problems with the aid of a computer.

Like many visions of the future, this one was inspiring but not wholly accurate. Certainly some of the key technologies to enable pen-based computing have come into fruition and have been adopted widely. However, the dream of ubiquitous handwriting and drawing recognition has not materialized. One can argue that this type of technology has yet to mature but will in the future.

What's fascinating about pen-based computing is how it is being used in alternative ways from the original vision, which was only a slice of the rich variety of ways a pen can be used in human-computer interaction. This article is about those other things: the ways in which pen input to a computer has been found to be valuable, along with where it is going.

"The original vision of pen-based computers was that they would bring the benefits of physical paper and pen to computer interaction… allowing people to interact more 'naturally' with the computer instead of typing."

PRACTICALITIES

There are some very practical issues that have dramatically affected the adoption of pen-based systems in the marketplace. Earlier work on computer input techniques, coming from a heritage of data entry, largely abstracted away some of the practical differences to present a more programmatic or "data centric" view of computer input devices.

Early work on interactive computer graphics by Foley and Wallace classified mouse and pen input as pretty much the same thing: both provide an x and y location and are capable of signaling an event (a pen press or mouse button press). However, later, researchers (such as Buxton) documented many subtle but important differences that affect the suitability of an input device for a specific task. For example, the mouse has some very practical properties that make it a successful and ubiquitous input device for desktop computers. It is a very efficient pointing device and allows the cursor location to remain unchanged when buttons are clicked.

Similarly, the pen has a set of very practical properties that define the contexts in which it will be effective. For example, one annoying aspect of pens is that they can be misplaced or lost, a problem that is exacerbated in the mobile device context. But it can be overcome by tethering the pen to the computer, or alternatively, the computer industry has recognized in many situations pointing by touch, without a pen, is sufficient.

The reverse has been used to an advantage too. For example, electronic white boards like Smart Board use multiple pens as an easy way to switch between ink colors when drawing.

Another practical but subtle and vexing issue with pens is that they require picking up. What happens when users want to switch hands or switch which input device they're holding? Some users become adept at keeping the pen in their hand while typing or using the mouse, but that is a generally inefficient and inelegant solution.

The key observation is that there is a rich set of issues and preferences surrounding any particular computer input situation and, in many of these cases, even if perfect handwriting and drawing recognition were available, the pen would still not be a preferred choice. Pen input is not an effective input technique if you cannot find the pen, or if one can simply type faster than write or draw.

THE ART OF SKETCHING

When is the pen a good choice? One task where the pen has a fundamental advantage is drawing. But even then, one has to be very careful to identify the precise drawing task.

The original vision of the user drawing diagrams and having the computer recognize and replace the rough drawings with formal structured graphics is not what the pen does best. Historically, the Achilles' heel has been getting the computer to recognize properly what the user has drawn. However, if the goal is to create formal structured drawings, then why not create formal

Figure 1: (a) Pen input is essential to the art of sketching. Pen-based tools like SketchBook Pro used with a Wacom tablet make powerful free-form sketching and painting tools that capture the manual skills of an artist.

(b) The same activity is still compelling in small formats (like the iPhone), even drawing with the finger. (c) The quality of drawing that can be performed with SketchBook Mobile for the iPhone is very high.

Figure 2: The user can interact with marking menus in two ways. (a) Selection can be made by popping up menus; or (b) once the user has memorized a path through the menus, selection be made much faster with a quick mark representing the path.

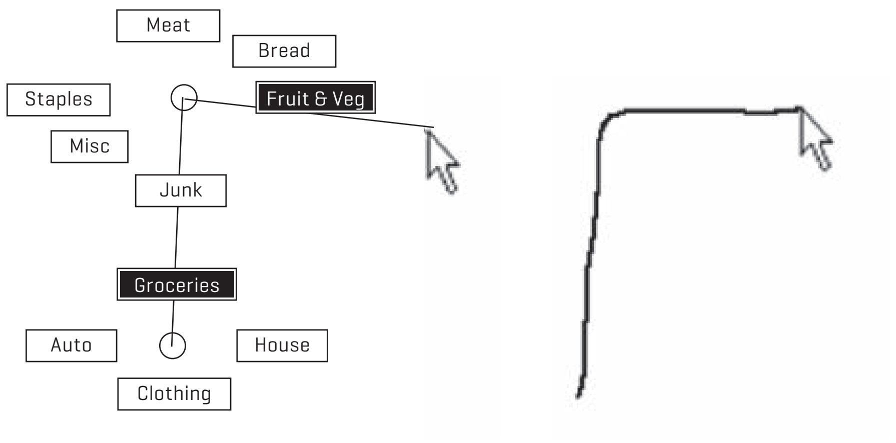


Figure 3: Marking menus use the spatial mapping between a vocabulary of zigzag marks and a hierarchy of radial menus. This permits easy to draw marks to be associated to arbitrary menu commands.

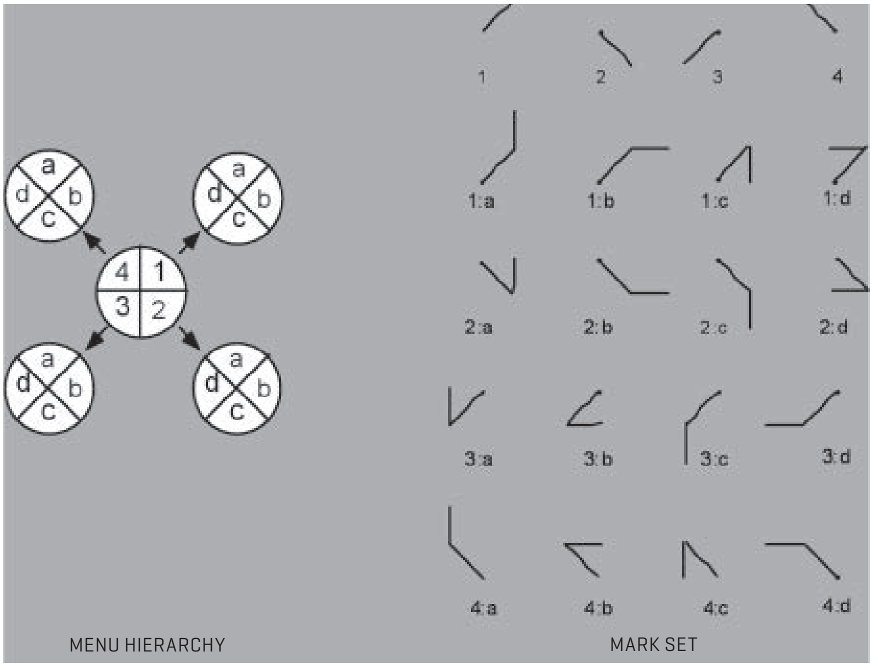


Figure 4: The Unistrokes character entry technique reduces letter input to single stroke marks that are easy to draw and easy to recognize.

structures directly? Unfortunately for the pen, mouse-based computer interfaces for drawing structured graphics are effective and arguably have become a more "natural" way to create them. Nowadays reverting from mouse to pen would displace industry practices.

However, if you need hand-drawn graphics, like the kind you make when drawing with a pen on paper, pen-based input shines. While this may appear to be trivial example, its value is constantly misunderstood and underestimated. The appeal of sketching never fails to amaze me. I was a member of the original development team that created the first version of Autodesk SketchBook Pro, a paint application designed specifically for sketching and capturing hand-drawn ideas with all the sensitivity of drawing with pencils, pens, and markers on paper (see **Figure 1**). Recently, I was approached with the idea of making a version of SketchBook for the Apple iPhone. While I thought it was a cool technical challenge, I could not imagine a really good reason why someone would want to use it.

Fast-forward several months and the more fully featured paint program SketchBook Mobile went on sale on the iTunes App Store. Its popularity was a pleasant surprise and it became one of the most downloaded applications on the store. What was even more amazing was what people were drawing with it. **Figure 1b** and **1c** show an example. Obviously, the artists creating these works were enamored with sketching even on this small format, or rather, they were interested in sketching because it was this format – the iPhone.

PEN-BASED INPUT≠EASE OF USE

While the pen allows high-quality sketching, it also sets the user's expectation for symbol recognition. Give pens to users, and they expect to be able to input handwriting and symbols and have the computer recognize them. I learned this lesson years ago demonstrating a little program that allowed the user to create circles, squares, and triangles by drawing them with a pen. I was demonstrating the program as part of our laboratory's "open house" to the general public. People came by my booth and I explained how you could use a pen to input commands to the computer and showed examples drawing circles, squares, and triangles where the computer replaced the hand-drawn objects with "perfect" ones. I then asked people to try it themselves. Surprisingly, most people didn't try drawing circles, squares, or triangles but tried writing their names. People's expectations were that anything could be recognized and this overrode any of my instructions beforehand. People expect a system with very general symbol recognition the moment a pen reaches their hands.

The subtle lesson here is that pen input or symbolic input is not inherently "easy to use" because it does not reveal to the user the capabilities of the system. This is a critical insight and it is this property of "self-revelation" that makes modern graphic user interfaces "easy to use"—specifically, by displaying graphics like text, icons, pictures, and menus, the computer "reveals" to users what they can and can't do, where, and when. We can think of graphical interaction widgets like buttons, and menus as "self-revealing." The method for finding out what functions are available and invoking those functions are combined into the same entity, namely, the button or menu.

Symbolic markings made with a pen are not self-revealing. A user can draw any shape or symbol and there is nothing intrinsic in that interaction that shows the user what marks the system recognizes or what they do. One simple method of "revealing marks" is to display a "cheat sheet," a list that shows the correspondence between a mark and the command it invokes. For example, a cheat sheet might show that drawing "C" copies and drawing "V" pastes.

Much research on pen-based user interfaces involves systems in which symbolic marks are used. Early work by C.G. Wolf used the symbolic language from paper and pencil proofreaders editing marks to design a system to edit text using a pen to mark up a document. Similarly, a system called MathPaper supports inputting mathematical formulas using the pen. This approach holds the promise of commands being easy to remember and perform. It's easy to remember "C" is for copy and it is also very quick to draw the "C" with the pen. However, complications arise when many symbolic marks are needed, and there is no prior existing vocabulary of marks. If more than a dozen "command marks" are needed it becomes difficult to design meaningful ones. In this case, it would be more effective to use existing graphical user interface techniques of icons, button, or menus for these commands.

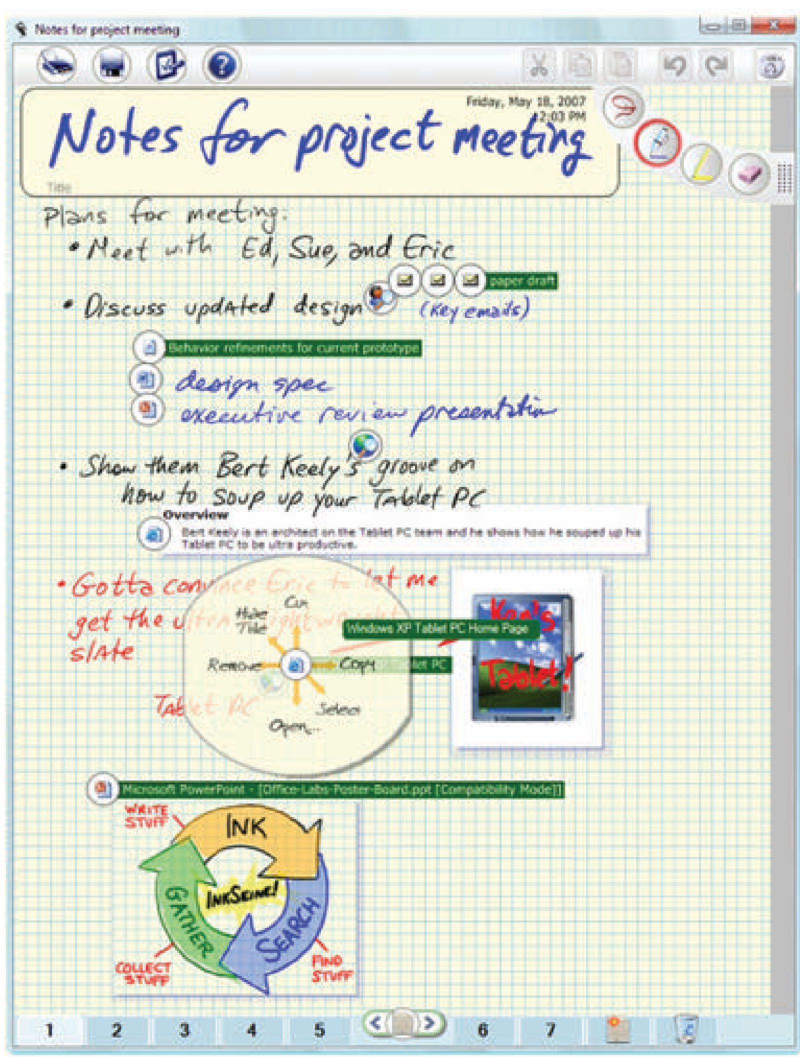
MARKING MENUS

A technique called marking menus was developed to address this problem. Marking menus combine a vocabulary of marks with a pop-up graphical menu to allow a user to learn and use marks that are easy and fast to draw. Like icons and buttons, marking menus interactively reveal the available functionality, and the computer recognition of the marks is simple and reliable. Essentially, marking menus combine the act of revealing functions to the user and drawing a mark.

A simple marking menu works as follows: As with a regular pop-up menu, a user presses the pen down on screen and a menu pops up. This menu differs from the linear menus we commonly see in that the menu items are displayed radially in a circle surrounding the tip of pen. A user can select a menu item by moving in the direction of the desired item and lifting the pen (**Figure 2**).

Marking menus are designed so

Figure 5: InkSeine combines the advantages of the free-flow of a pen and paper notebook with direct manipulation of digital media objects.



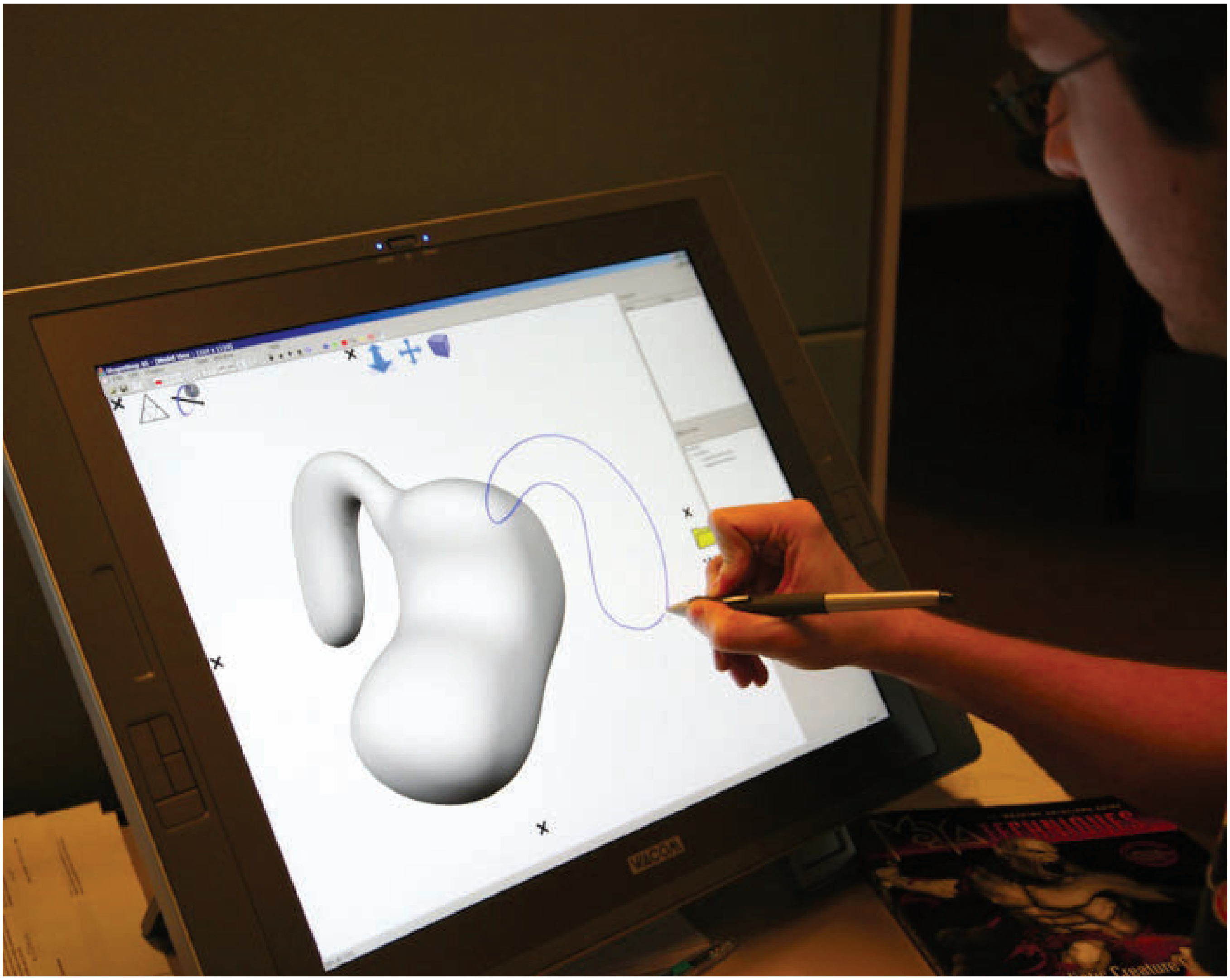
that a user does not have to do anything special to switch between selecting from the menu and using a mark. If the user presses the pen down and hesitates (waiting for the system to show what's available) the menu is displayed and selection can be performed in the usual way by pointing to menu items. However, if the user presses the pen down but does not hesitate and begins to move right away, a mark is drawn. This way, a user can gradually move from selecting a command via the menu to selecting by drawing a mark. Novices to the system can pop up the menu to recall the location of a particular menu items. With practice, the user memorizes the location of menu items and can select them quickly by making a quick mark in that direction. Novice use is a rehearsal of expert performance. Research has shown that selection with a mark can be up to ten times faster than popping up the menu, making this technique very useful for frequently selected menu items.

Marking menus takes a meaningless vocabulary of zigzag marks and makes correspondence to hierarchical menu items (see **Figure 3**). The result is a technique for providing an easy-to-learn, easy-to-draw, shorthand symbol set to access arbitrary computer functions. Much research has been conducted on marking menus, and they have been successfully deployed in commercial CAD applications.

Other similar clever ways of exploiting pen input beyond recognizing traditional handwriting and symbols have been explored. The text entry systems Graffitti and Unistrokes resulted from research that analyzed what types of symbols are easy for a user to draw and easy for the computer to recognize, in hopes of supporting handwriting input that is easier, faster, and more reliable than traditional handwriting. **Figure 4** shows how Unistrokes redesigned the alphabet to support this.

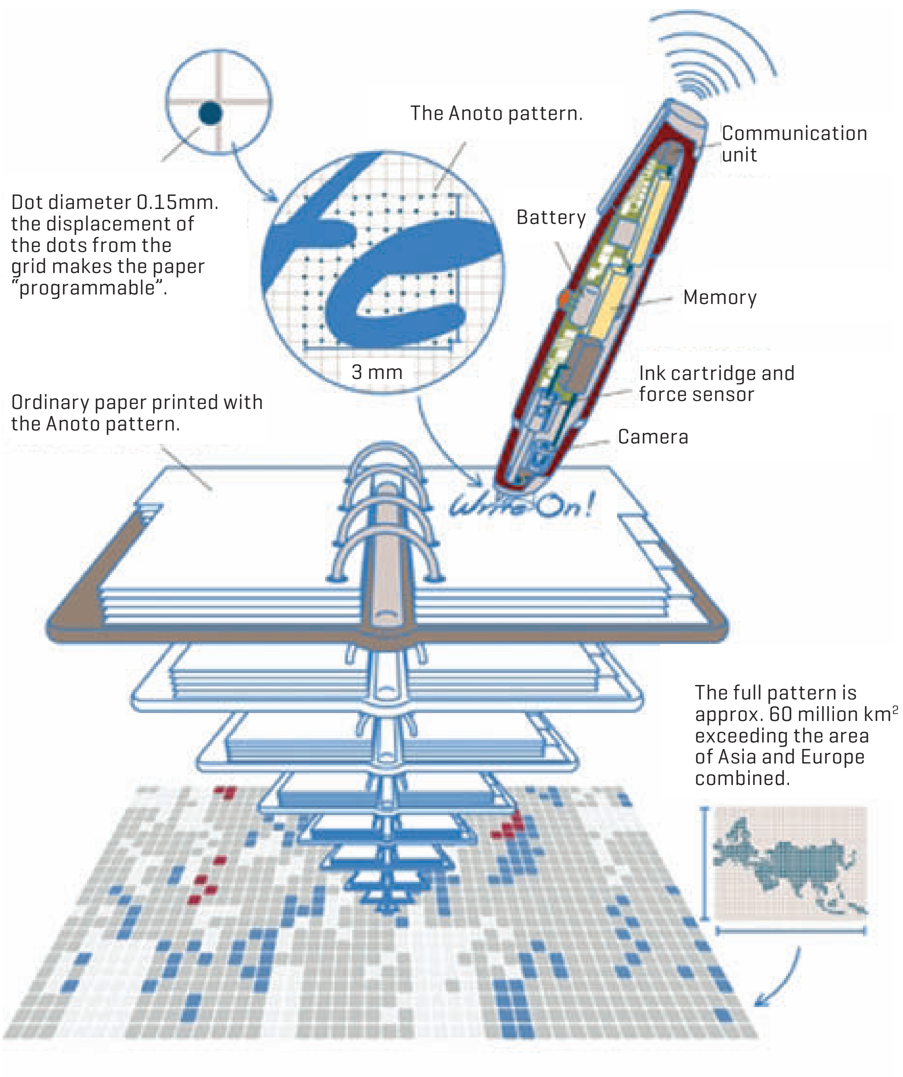
The concept behind marking menus has also been applied to handwriting input. Shumin Zhai and other researchers at the IBM Almaden Research Center

Figure 6: The application ShapeShop allows 3D shapes to be created quickly by inputting strokes and converting them to 3D graphics. Here the outline of the dog's ear is drawn and will subsequently be translated into a "blob" that corresponds to the shape of the input stroke and then connected to the dog's head.



developed the SHARK system, a graphical keyboard on which the user can input words by dragging from key to key with the pen. The path being dragged essentially creates a symbol that represents a particular word. Experiments on SHARK have showed that the user can learn this method of input and become proficient with it, and with practice, the rate of input can match touch-typing rates. As with marking menus, SHARK is a compelling example of how researchers are endeavoring to exploit a human's skill with the pen and ability to learn, to create human computer interactions that go beyond emulating traditional paper and pen.

Figure 7: A small camera in the tip of an Anoto pen allows the pen to sense and record what has being written and where.



PEN, THE GREAT NOTE-TAKER

Pen input research has also focused on the pen's ability to be used to fluidly switch between text input, drawing and pointing. Inspired by how people combine both drawing and handwriting in paper notebooks, these types of systems attempt to recreate and amplify this experience in the digital world. Ken Hinckley's InkSeine application is a prime example. With InkSeine a user can quickly throw together notes where ink, clippings, links to documents and web pages, and queries persist in-place. **Figure 5** shows an example. Unlike a mouse and keyboard system, where a user must switch between keyboard and mouse for text entry and pointing, InkSeine supports all these operations in a free form way, reminiscent of paper notebooks. (InkSeine is available as a free download from http://research.microsoft.com/en-us/redmond/projects/inkseine/.)

PEN INPUT 3D

Pen input can also be used to create 3D graphics. The goal of this work is for a user to be able to draw a perspective view of an object or scene and have the computer automatically recognize the 3D shape and recreate it accurately. A simple example is drawing a box in perspective and the computer automatically recognizes it as cube structure and creates the corresponding 3D geometry of a cube. The user can then rotate the object to see it from different viewpoints.

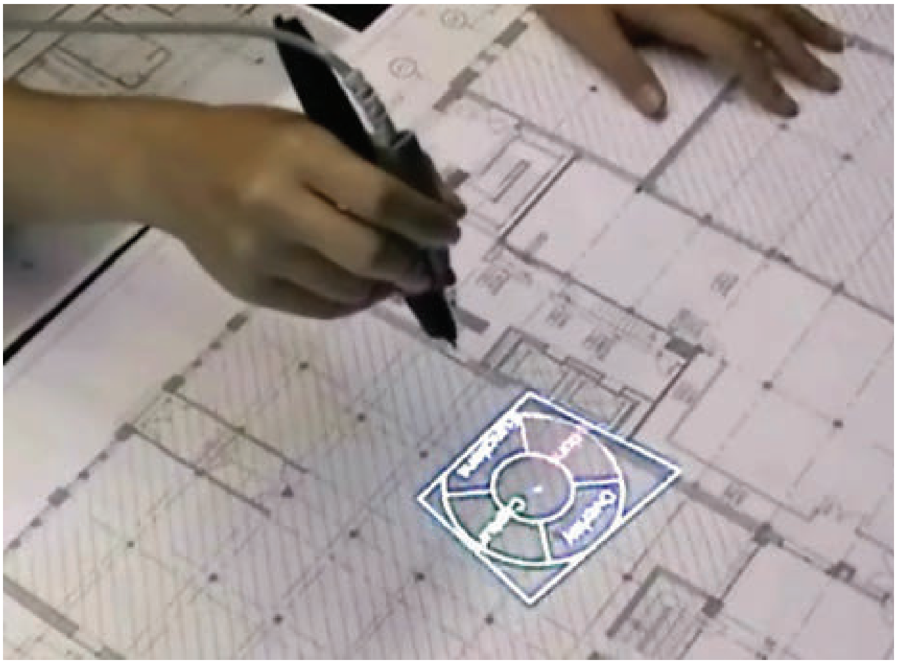
In general, this is a computer vision problem—recognizing shapes and objects in the real world—and much progress on this general problem remains. However, researchers have made progress with systems where pen input is interpreted stroke by stroke into three-dimensional objects. Research systems such as ILoveSketch and ShapeShop (shown in **Figure 6**) allow drawing on planes in 3D space or directly on to 3D surfaces.

BEYOND

Recently multi-touch systems, where the computer screen can sense multiple finger touches simultaneously, became a very hot topic for commercial usage. Some multi-touch systems are capable of sensing touch and pen

> "Researchers and designers have made the key observation that the pen interactions are distinctly different from touch interactions and can be exploited in different and interesting ways."

Figure 8: The PenLight research prototype: A digital pen senses its location on a building blueprint and a micro projector mounted on the pen displays an overlay of information accurately positioned on the blueprint.



input. Work by Balakrishnan has shown systems where the dominant hand holds the pen and non-dominant hand controls the frame of reference (e.g., rotation of the image being drawn on) are effective and desirable ways of interacting. Work by Hinckley has shown the benefits of a simple design in these multi-touch and pen systems where "the pen writes," "touch manipulates," and "the two combine for special functions."

Ultimately, how to use the pen in combination with multi-touch will be largely determined by needs of applications. However, researchers and designers have made the key observation that the pen interactions are distinctly different from touch interactions and can be exploited in different and interesting ways.

Research has looked at ways of augmenting the pen with special hardware functions to create "super pens." The Anoto pen is a digital pen with a tiny camera embedded in its tip (**Figure 7**). The pen can store handwriting, markings, and pen movements, and identify on which document they were made and precisely where the markings are. This allows the world of paper and digital technologies to be combined.

This concept has been adapted from paper to physical 3D objects. Guimbiere and Song developed a computer-aided design system where a user could make editing marks on 3D physical objects using an Anoto pen. The objects, printed with a 3D rapid prototyping printer, have the same type of invisible markings as the paper allowing the pen to identify the object and the location of markings. Drawing a doorway on the side of physical model of the building causes the doorway to be added to the virtual model.

"Some super pens, like Anoto's, can store handwriting, markings, and pen movements, and identify on which document they were made and precisely where the markings are. This allows the world of paper and digital technologies to be combined."

Pushing these ideas further, research has explored augmenting these "super pens" with output devices, like a LED screen that displays a line of text and an audio speaker. The PenLight system pushes this even further and uses a micro projector mounted on the pen that allows it to act like an "information flashlight." With the PenLight, the camera in the pen not only knows the location of the pen relative to the document it is over, but with the micro projector it can overlay relevant information. **Figure 8** shows the Pen Light system being used over a blueprint for a building.

THE PEN TO COME

The original vision of pen-based computers was that they would bring the benefits of physical paper and pen to computer interaction by utilizing handwriting input and free-form drawing, allowing people to interact more "naturally" with the computer instead of typing. However, because reliable handwriting recognition wasn't available and people needed typeface text for a majority of tasks, the keyboard and mouse became more popular—so much so that the keyboard and mouse now seem to be the "natural" means of interacting with a computer.

But the pen remains essential for some tasks, like sketching and free form idea input, and in these applications it has found success. Furthermore, researchers explored ways of using pen input beyond emulating pen and paper, such as marking menus and PenLight, and this research has resulted in useful, successful and interesting technologies. Using this type of thinking, many interesting and inspiring explorations for pen input remain ahead.

Biography

Dr. Gordon Kurtenbach is the director of research at Autodesk Inc., where he oversees the Autodesk Research group whose focus is on human-computer interaction, graphics and simulation, environment and ergonomics, and high-performance computing. He has published numerous research papers and holds more than 40 patents in the field of human-computer interaction.

Interactive Surfaces and Tangibles

Tap. Slide. Swipe. Shake. Tangible user interfaces have some scientists toying around with stuff you can really put your hands on.

By Sergi Jordà, Carles F. Julià, and Daniel Gallardo

DOI: 10.1145/1764848.1764855

In the last decade, human-computer interaction research has witnessed a change in focus from conventional ways to control and communicate with computers (keyboard, joystick, mouse, knobs, levers, buttons, etc.) to more natural and unconventional devices, such as gloves, speech recognition tools, eye trackers, cameras, and tangible user interfaces. With the advent of the Internet and the ubiquity of personal computers in the 1990s, the graphical user interface (GUI) emerged as the pervasive interface that both users and designers had to deal with. At the same time, albeit lesser known at that time, computing started to progressively move beyond the desktop into new physical and social contexts as a result of both technological advances and a desire to surpass the WIMP—window, icon, menu, pointing device—limitations.

Nowadays, when "multi-touch" and "the two-finger pinch-zoom" are part of a user's daily life, the area of tangible interaction seems to have finally entered the mainstream. Despite this, many do not realize that tangible interaction conveys much more than sharing photo collections or navigating Google Maps using our fingers.

In this article we give an overview of tangible interaction and tangible user interfaces, their conceptual origins, and some of their seminal implementations. We'll also describe some of the more popular current tendencies such as multi-touch and interactive tabletops and surfaces. We will conclude by discussing some of the key benefits of this type of interaction, surveying some of the application areas where TI is showing more promising results.

> "While multi-touch mobile phones should increase the richness of interaction of these interfaces, at the moment we only find basic gestures."

TANGIBLE INTERACTION

In a way, tangible interaction can be seen as an extension and deepening of the concept of "direct manipulation," a term that was first introduced by Ben Shneiderman in 1983 within the context of office applications and the desktop metaphor [**32**]. Since then, it has also become closely associated with GUIs and WIMP-based interaction. While WIMP-based GUIs always incorporate direct manipulation to some degree, the term does not just imply the use of windows or even graphical output. The idea behind direct manipulation was to allow users to "directly manipulate" objects presented to them, using actions that would loosely correspond to the physical world, assuming that real-world metaphors for both objects and actions would make it easier for them to learn and use an interface.

Tangible user interfaces combine control and representation in a single physical device [**34**]. With GUIs, users interact with digital information by selecting graphic representations (icons, windows, etc.) with pointing devices, whereas tangible interaction emphasizes tangibility and materiality, physical embodiment of data, bodily interaction, and the embedding of systems in real spaces and contexts.

Hiroshi Ishii at the MIT Media Lab coined the term tangible user interface in 1997 [**17**], although several related research and implementations predate this concept. Ishii picked the abacus as the source of inspiration and the ultimate tangible interaction metaphor because, unlike pocket calculators or computers, in the abacus, input and output components coincide and arithmetical operations are accomplished by the direct manipulation of the results.

Following this captivating idea, Ishii envisioned TUIs as interfaces meant to augment the real physical world by coupling digital information to everyday physical objects and environments, literally allowing users to grasp data with their hands, thus fusing the representation and control of digital data and operations with physical artifacts. (See also in this issue a profile of Ishii on page 49.)

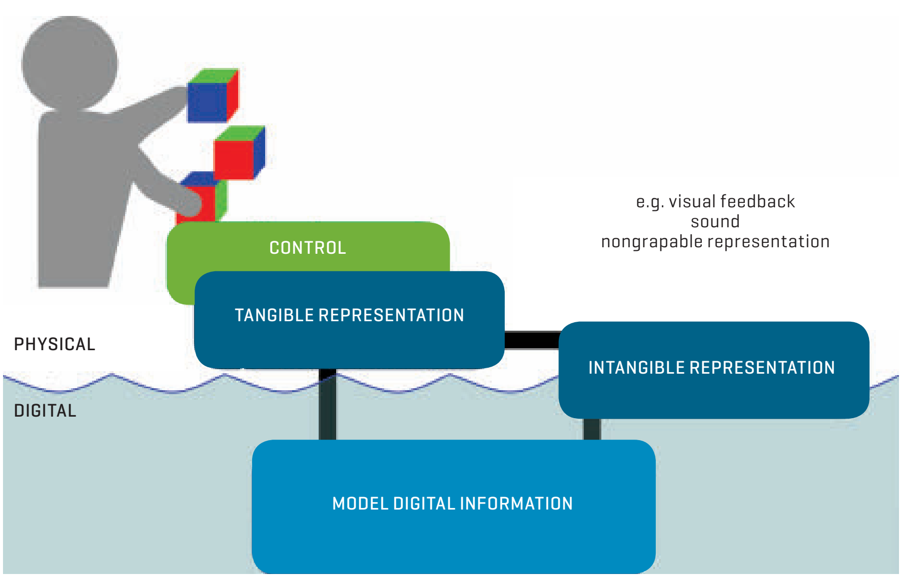
PIONEERS: 1983-1997

The use of the nowadays-ubiquitous pinch gesture so cherished by Apple iPhone users dates back at least to 1983 [**35**]. At that time, the media artist and engineer Myron Krueger, passionate about unencumbered rich gestural interaction (no mice, no gloves) was already working with vision capture systems that were able to track the users' full bodies, as well as their hands and multiple fingers, allowing them to interact using a rich set of gestures [**20, 21**]. Whereas in his work Video Place (1983) users interacted with a big screen, predating Sony's EyeToy (2002) by 20 years (see **Figure 1**), Video Desk (1983), based on a horizontal desktop configuration, showed the use of pinch and other two-finger and two-hand gestures for redefining objects'shapes, scaling them or translating them.

One year later, the same year the first Macintosh computer was released, Bill Buxton started researching multi-touch and bimanual input [**5**], first at the University of Toronto and later at Xerox Park and at the Alias|Wavefront company, developing some prototypes such as a multi-touch tablet [**22**]. In 1995, Buxton together with then PhD students Fitzmaurice and Ishii, demonstrated how the sensing of the identity, location and rotation of multiple physical devices on a digital desktop display, could be used for controlling graphics using both hands. This work introduced the notion of graspable interfaces [**11**], which were to become two years later "tangible interfaces." From the same research team is also the Active Desk (see **Figure 2**), one of the first tabletop systems that combined a sensing camera and a projector [**4**].

Perhaps conceptually closer to the notion of grabbing and manipulating information is the work of the architect and professor of design John Hamilton Frazer, who as early as in the late 1970s started working on what he called "Intelligent Physical Modeling Systems", developing a set of intelligent cubes that would know the position of its surrounding neighbors. The purpose of these first prototypes was to build a sketching system that would let people easily and intuitively work and prototype in three dimensions, thus permitting users of future buildings, without any special design knowledge, to be directly involved in their conception process [**12**]. Thirty years later, 2D or 3D "intelligent building blocks," such as David Merrill's 2D Siftables [**24**] (see **Figure 3**), still constitute an emerging and promising trend inside TI!

Figure 1: The model-view-controller in tangible interaction (inspired by Hiroshi Ishii).



CONSOLIDATION: 1997-2005

The TUI "consolidation phase" can be considered to start with the publication of Ishii and Ulmer's seminal paper "Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms" [**17**], and the subsequent work carried out by his team at the MIT Media Lab. A first student of Buxton, Ishii helped conceptualize the philosophy behind tangible interaction and, together with his own students at the Media Lab, he started to conceive and implement new developments that demonstrated the potential of TUIs.

Their Sensetable [**28**] used a top-down projected system and an electromagnetic tracking system for detecting the positions and the orientations

Figure 2: Bill Buxton interacts with the Active Desk.

of multiple wireless objects on the table surface. It became the base for most of the tabletop-related projects developed at the Media Lab in the following years, including the Audiopad, one of the first musical tabletops.

Around the same time, Mitsubishi Electric Research Laboratories started to explore multi-user and multi-touch tabletop interfaces with DiamondTouch [7], a rectangular table focused on supporting meetings and discussions, and the first and still only one capable of distinguishing between users.

The Reactable is a collaborative musical tabletop conceived and developed since 2003 at the Pompeu Fabra University in Barcelona (see **Figure 4**). It provides both multi-touch and tangible objects interaction by means of reacTIVision, an open-source, cross-platform computer vision framework for the tracking of fiducial markers and combined multi-touch finger tracking [**2**]. Specially developed for the Reactable project, reacTIVision

Figure 3: David Merrill's Siftables are known as "intelligent building blocks."

Photo by Amit Zoran; small portraits by Orit Zuckerman, MIT Media Lab

Figure 4: The Reactable.

is nowadays widely spread among the tabletop developer community. The Reactable was first presented in a concert in 2005 [**19**], but two years later it accomplished unparalleled mass popularity (considering it is an academic work), when millions of people watched the demo on YouTube. Not long after, Björk used it in her 2007 world tour.

PlayAnywhere [**36**] from Microsoft Research seems to be the first attempt at building a compact and portable multi-touch surface (see **Figure 5**). By handling a projector on the side instead of above the interface, PlayAnywhere also partially solved a common problem with most of the contemporary interfaces—occlusion. Its computer vision tracking technology, which analyzes the user's shadows to detect contact with the interface, was also innovative. With changes and additions, most of the work carried out by Wilson and his team at Microsoft Research would lead to the presentation of Microsoft's commercial tabletop, the Surface, in 2007.

Above everything else, the most influential work on the multi-touch field, arguably leading to its hype and consolidation, was Jeff Han's "low-cost multi-touch sensing through frustrated total internal reflection" [**15**]. He demonstrated a new, precise, and inexpensive technology for rear-projected multi-touch interfaces, along with gorgeous demo applications that captivated the common user. Han's demos on YouTube finally pushed multi-touch from the academic world fully into the larger technology space.

TABLETOPS' AFFORDANCES AND TYPOLOGIES

Several of the early aforementioned prototypes were in fact tables that often allowed their users to interact in two complementary ways: touching the table's surface directly, and manipulating specially configured real physical objects on its surface.

Typically, this type of interface allows more than one input event to enter the system at the same time. Instead of restricting input to an ordered sequence of events (click, click, double click, etc.), any action is possible at any time and position, by one or several simultaneous users.

Multi-touch interaction is arguably the most commercially successful capability of horizontal surfaces. And the "pinch-zoom" technique is only one example of hundreds of possibilities.

The other implicit capacity of table-shaped interfaces is the ability to literally support physical items on them. Users can interact with objects of volume, shape, and weight, and when the tracking system is able to identify these objects and track their position and orientation, the potential bandwidth and richness of the interaction goes thus far beyond the simple idea of multi-touch. Interacting with the fingers still belongs to the idea of pointing devices, while interacting with physical objects can take us much farther. Such objects can represent abstract concepts or real entities. They can relate to other objects on the surface. They can be moved and turned around on the table surface, and all these spatial changes can affect their internal properties and their relationships with neighboring objects.

Additionally, with the combination of the tracking of control objects on the table with projection techniques that do convert the table into a flat screening surface, these systems can also fulfill the seminal ideal of tangible interaction, of "adding digital information to everyday physical objects," allowing digital entities to coexist as fully digital non-physical form and as shared digital-physical form.

"Nowadays, when 'multi-touch' and 'the two-finger pinch-zoom' are part of a user's daily life, the area of tangible interaction seems to have finally entered the mainstream."

It should be stressed, though, that not all existing tabletops technologies allow dual interaction, and others, such as SMART Table 2003 and DiamondTouch, support only multi-touch. To be able to place objects on a surface, the tabletop must be horizontal—not tilted. In that sense, some recent studies analyze the benefits and disadvantages of tilted versus horizontal surfaces (for example [**25**]), suggesting that in many individual use cases, if tangible objects where not to be supported, tilted interfaces (like traditional architect's tables) seem to be more engaging and feel more natural [**26**].

Figure 5: PlayAnyWhere from Microsoft Research.

An early example of a tangible tabletop interface, which may allow us to unveil and better understand its potential, is Urban Planning (URP), a system developed as a town planning aid in the late 1990s at the MIT Media Lab [**3**]. URP is a town-planning simulator in which various users can analyze in real time the pros and cons of different urban layouts by arranging models of buildings on the surface of an interactive table, which represents the plan of a town or a district. The surface provides important information, such as building shadows at different times of the day.

URP is executed on an augmented table with a projector and a camera pointed at the surface from above. This system permits the detection of changes in the position of the physical objects on the table and also projects visual information concerning the surface. Other elements can be included, such as a clock to control the time of day in the system. URP detects any changes made in real time and projects shadows according to the time of day. All these properties could obviously be achieved with the usual mouse-controlled software on a conventional screen, but the interest of this simple prototype does not lie in its capacity to calculate and simulate shadows, but rather in the way in which information can be collectively manipulated, directly and intuitively. This example, although quite simple, already unveils some of the most important benefits of tabletop interaction: collaboration, naturalness, and directness.

MULTI-USER COLLABORATION

The social affordances associated with tables directly encourage concepts such as "social interaction and collaboration"[**16**] or "ludic interaction" [**14**]. Many researchers do in fact believe that the principal value of tangible interfaces may lie in their potential for facilitating kinds of collaborative activities that are not possible or poorly supported by single user technologies [**23**]. A research community has been

Figure 6: The Reactable, a collaborative musical tabletop, has a circular surface for interaction.

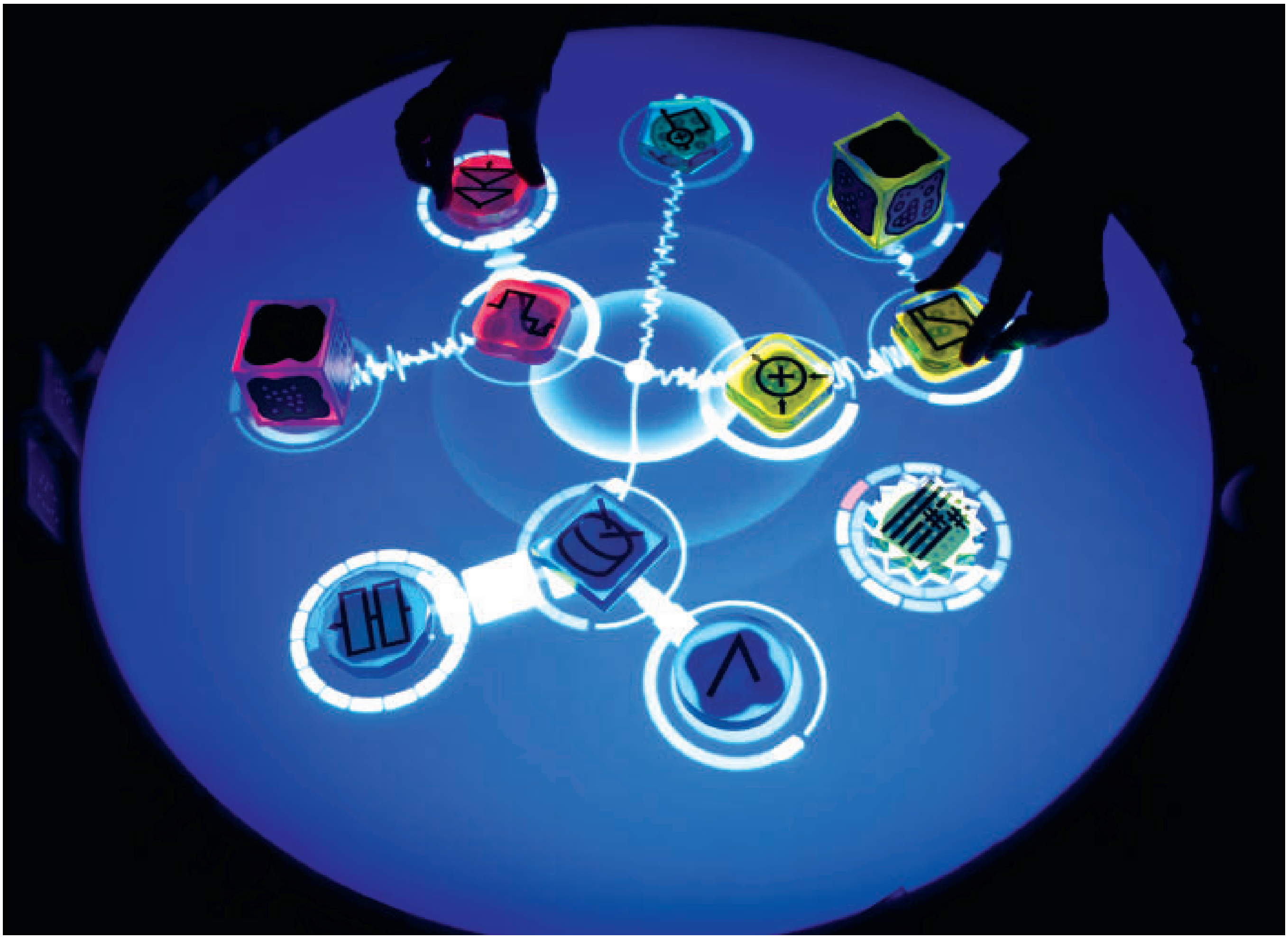
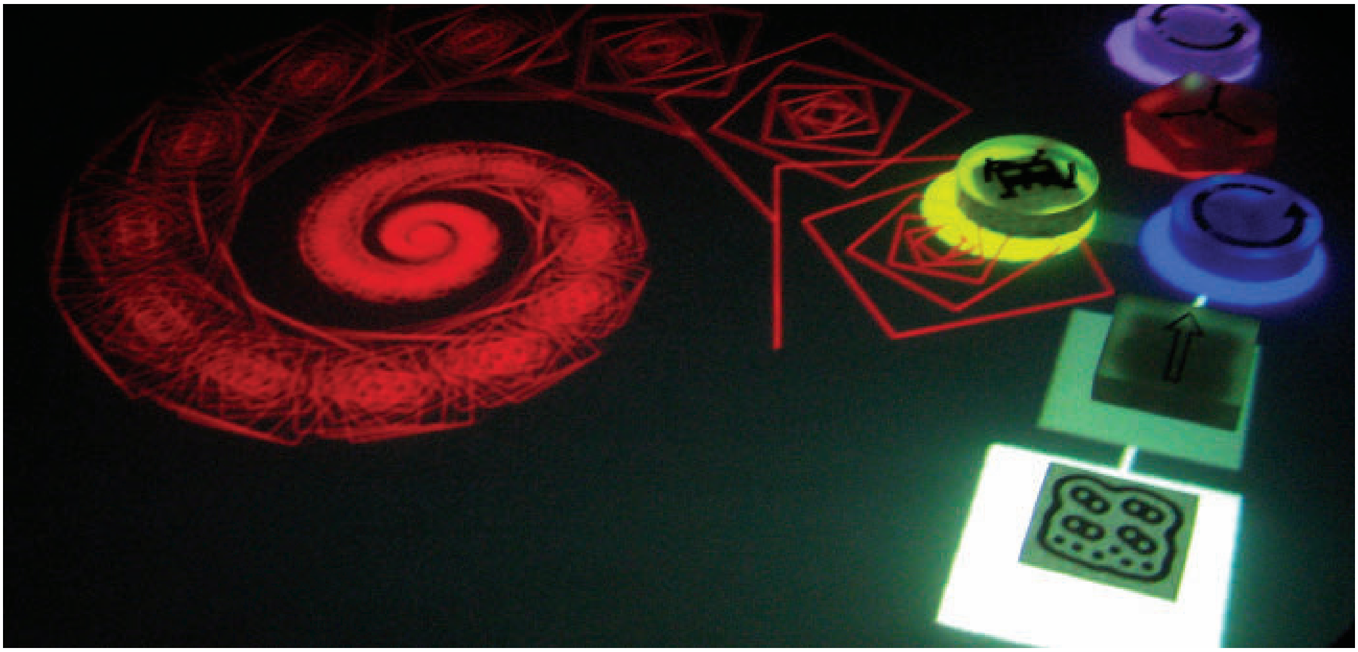


Figure 7: Turtan, a Logo tangible programming project, is circular like Reactable.



growing around these technologies and the concept of "shareable interfaces," a generic term that refers to technologies that are specifically designed to support groups of physically co-located and co-present to work together on and around the same content.

Until recently, most research on computer-supported cooperative work (a term coined by Irene Greif and Paul M. Cashman in 1984), has more often concentrated on remote collaboration. But if we restrict ourselves to collocated collaboration, the type of devices used (for example screens versus tables) seem to make a manifest difference.

Rodgers and Rodden have shown for example that screen-based systems inevitably lead to asymmetries concerning the access and the creation of information [**29**]. While these systems make possible for all participants to view the external representations being displayed (for example, through using whiteboards and flipcharts), it is more difficult for all group members to take part in creating or manipulating them. In particular, one person can often dominate the interactions by monopolizing the keyboard, mouse, or pen when creating and editing a document on a shared interactive whiteboard. Once a person is established in a particular role (for example note-taker, mouse controller) she or he tends to remain in it. Moreover, those not in control of the input device, can find it more difficult to get their suggestions and ideas across.

Rodgers et al. [**29**] have done some user studies around interactive tables, for learning the new opportunities for collaborative decision-making that shared interactive tabletops can provide. They conclude that collaborative decision-making can indeed be promoted by providing group members with equal access and direct interaction with digital information, displayed on an interactive table surface. They observe that these interfaces also foment discussion, and that the sharing of ideas and invitations to others to take a turn, to respond, confirm, or to participate, all tended to happen at the same time the participants were interacting with the table, supporting their opinions with gestures and the table responses.

Some tabletop implementations have even strengthened this collaborative aspect with idiosyncratic design decisions. Such is the case of the collaborative tabletop electronic music instrument Reactable [**18**] (see **Figure 6**), the Logo tangible programming project Turtan [13] (see **Figure 7**) or the Personal Digital Historian System [**31**], all of which are based on circular tables and use radial symmetry, for promoting collaboration and eliminating head position, leading voices, or privileged points of view and control.

"It is hard to think about shared controls when models and inspirational sources comes from WIMP-based, single-user interactive computer applications."

SHARING CONTROL

Sharing data between users, for example in the form photo collections (for example [**6, 31**]), probably constitutes nowadays, together with map navigation, the most popular demo for tabletop prototypes (for example Microsoft Surface). Communication in general is definitely about sharing data [**30**], but it is not about sharing documents or files—it is about sharing real-time, on-the-fly-generated data.

This idea of sharing control versus sharing data is indeed becoming more frequent on tabletops. One clear example is Reactable [**18**], a multi-user musical instrument that is better described as a contraption for sharing real-time control over computational actions, rather than for sharing data among its users.

This idea of sharing control versus the sharing of data is indeed strongly linked to music performance, but tangible applications with a similar philosophy are also becoming more frequent in non performance-related domains. The Patcher [**9**] presents a set of tangible resources for children in which tangible artifacts are better understood as resources for shared activity rather than as representations of shared information. Fernaeus, Tholander, and Jonsson [**10**] identify a "practice turn" in tangible interaction and HCI in general, that is moving from a data-centric view of interaction to one that focuses on representational forms as resources for action. Instead of relying on the transmission and sharing of data, the action-centric perspective is looking for solutions that emphasize user control, creativity, and social action with interactive tools.

No matter how influential this paradigm shift may be felt in a near future, the truth is that it is hard to think on shared control when models and inspirational sources comes from WIMP-based, single-user interactive computer applications. Much of the efforts taken until today in the field of CSCW have been in that direction, trying to convert single-user applications into multi-user collaborative applications. But sequential interaction has proved to be too inflexible for collaborative work requiring concurrent and free interactions [**1, 8, 27, 33**].

Sharing control is problematic and cumbersome in WIMP-based interaction. While synchronicity problems and inconsistencies caused by simultaneous accesses can be "solved" and managed, this interaction model does clearly not constitute the best inspirational source for the new types of collaboration we envision tangibles can convey.

MASS MARKET

If we were to look at the present commercial success of the technologies and interaction models we have described, it seems that the only one currently reaching the mass market is multi-touch. The new generation of mobile phones advertises the ability of recognizing multi-touch gestures, and the same happens to the now changing market of personal computers: the new versions of operating systems already add some support for multi-touch input on the screen.

While this should increase the richness of interaction of these interfaces, at the moment we only find basic gestures such as pinch-and-zoom and similar ones that are limited to two fingers. It's a starting point, but we're still missing the main benefits of multi-touch technologies: full hand, bimanual and multi-user interaction.

Sadly, these restrictions come from intrinsic limitations of cell phones and computer screens which are specifically designed for single user interaction. Additionally, in the case of phones, their small size limits full hand or bimanual interaction. On the personal computers side, developers are struggling for adapting multi-touch to the rigid structure of WIMP, which prevents the apparition of most of the features of multi-touch.

Tabletop interfaces favor multi-dimensional and continuous real-time interaction, exploration and multi-user collaboration. They have the potential to maximize bidirectional bandwidth while also contributing to delicate and intimate interaction, and their seamless integration of visual feedback and physical control allows for more natural and direct interaction. They are promising interfaces, but while some vendors are starting to push tabletop products on the market, their penetration is still minimal. However, we are confident that tabletops will be soon ubiquitous, not necessarily in the desktop, but on the meeting rooms, the stages, and wherever complex social interaction, exploration, and collaboration take place.

Biographies

Sergi Jordà holds a BS in fundamental physics and a PhD in computer science and digital communication. He is a researcher in the Music Technology Group of Universitat Pompeu Fabra in Barcelona, and a lecturer at the same university, where he teaches computer music, HCI, and interactive media arts. He is one of the inventors of the Reactable.

Carles F. Julià is a PhD student in the Music Technology Group at Universitat Pompeu Fabra. He teaches interactive systems to undergraduates and tangible and tabletop interfaces in a professional master in Interactive Musical Systems. He has a background in computing science, cognitive science, and interactive media.

Daniel Gallardo is a PhD student in the Music Technology Group at Universitat Pompeu Fabra, working on a thesis on a new tabletop prototype founded by Microsoft Research Lab (Cambridge, UK). He teaches CS and telecommunications, and a course in physical computing in a professional master in Interactive musical systems design. His background is in computer science, and he has a masters in cognitive systems and interactive media.

References

1. Begole, J., Rosson, M. and Shaffer, C. 1999. Flexible collaboration transparency: Supporting worker independence in replicated application sharing systems. *ACM Transactions on Computer-Human Interaction* (TOCHI), 6, No. 2. 95-132.
2. Bencina, R., Kaltenbrunner, M. and Jordà, S. 2005. Improved Topological Fiducial Tracking in the reacTIVision System. In *Proceedings of the IEEE International Workshop on Projector-Camera Systems*.
3. Ben-Joseph, E. and Ishii, H. and Underkoffler, J. and Piper, B. and Yeung, L. 2001. Urban simulation and the luminous planning table: Bridging the gap between the digital and the tangible. *Journal of Planning Education and Research*.
4. Buxton, W. 1997. Living in Augmented Reality: Ubiquitous Media and Reactive Environments. In *Video Mediated Communication*, K. Finn, A. Sellen & S. Wilber (eds.). Hillsdale, N.J.: Erlbaum, 363-384.
5. Buxton, W., Myers, B. 1986. A study in two-handed input. In *SIGCHI86* ACM. 321-326.
6. Crabtree, A., Rodden, T., and Mariani, J. 2004. Collaborating around collections: informing the continued development of photoware. In *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work* (Chicago, Illinois, November 6-10, 2004). CSCW '04. ACM, New York, NY. 396-405.
7. Dietz, P., Leigh, D. 2001. DiamondTouch: a multi-user touch technology. *UIST '01: In Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology*. 219-226.
8. Stefik, M., Bobrow, D., Foster, G., Lanning, S., and Tatar, D. 1987. WYSIWIS revised: Early experiences with multiuser interfaces. *ACM Transactions Office Information Systems*, 5, No. 2. 147-167.
9. Fernaeus, Y. and Tholander, J. 2006. Finding Design Qualities in a Tangible programming space. In *Proceedings CHI 2006*, Montreal.
10. Fernaeus, Y., Tholander, J., and Jonsson, M. 2008. Beyond representations: Towards an action-centric perspective on tangible interaction. *International Journal of Arts and Technology*.
11. Fitzmaurice, G. W., Ishii, H., Buxton, W. 1995. Bricks: laying the foundations for graspable user interfaces. In *CHI '95: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 442-449.
12. Frazer, J. 1995. An evolutionary architecture. Architectural Association.
13. Gallardo, D., Julia, C.F. and Jordà, S. 2008. TurTan: A tangible programming language for creative exploration. In *3rd IEEE International Workshop on Horizontal Interactive Human Computer Systems*, 2008. TABLETOP 2008. 89-92.
14. Gaver, W.W., Bowers, J., Boucher, A., Gellerson, H., Pennington, S., Schmidt, A., Steed, A., Villars, N. and Walker, B. 2004. The drift table: designing for ludic engagement. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems*, ACM Press. 885-900.
15. Han, J. Y. 2005. Low-cost multi-touch sensing through frustrated total internal reflection. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology* (Seattle, WA, October 23-26, 2005). UIST '05. ACM, New York, NY. 115-118.
16. Hornecker, E. and Buur, J. 2006. Getting a grip on tangible interaction: A framework on physical space and social interaction. In *Proceedings CHI 2006*. 437-446.
17. Ishii, H., Ullmer, B. 1997. Tangible bits: towards seamless interfaces between people, bits and atoms. In CHI '97: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 234-241.
18. Jordà, S., Geiger, G., Alonso, M., Kaltenbrunner, M. 2007. The reacTable: exploring the synergy between live music performance and tabletop tangible interfaces. In *TEI '07: Proceedings of the 1st International Conference on Tangible and Embedded Interaction*. 139-146.
19. Jordà, S., Kaltenbrunner, M., Geiger, G. and Bencina, R. 2005 The reacTable*. In *Proceedings of the International Computer Music Conference* (ICM05).
20. Krueger, Myron, W. 1991. *Artificial Reality II*. Reading, MA: Addison-Wesley.
21. Krueger, M. W., Gionfriddo, T., Hinrichsen, K. 1985. VIDEOPLACE—An artificial reality. In *CHI '85: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 35-40.
22. Lee, S.K., Buxton, W., Smith, K. C. 1985. A multi-touch three dimensional touch-sensitive tablet. *SIGCHI85* ACM. 21-25.
23. Marshall, P., Rogers, Y. and Hornecker, E. 2007. Are Tangible Interfaces Really Any Better Than Other Kinds of Interfaces? Workshop on Tangible User Interfaces in Context and Theory, ACM *CHI 2007*.
24. Merrill, D., Kalanithi, J., and Maes, P. 2007. Siftables: towards sensor network user interfaces. In *Proceedings of the 1st International Conference on Tangible and Embedded Interaction* (Baton Rouge, Louisiana, February 15-17, 2007). New York, NY. 75-78.
25. Morris, M.R., Brush, A.J., Meyers, B.R. 2008. A field study of knowledge workers' use of interactive horizontal displays. In *3rd IEEE International Workshop on Horizontal Interactive Human Computer Systems*, 2008. TABLETOP 2008. 105-112.
26. Muller-Tomfelde, C. and Wessels, A. and Schremmer, C. 2008. Tilted tabletops: In between horizontal and vertical workspaces. In *3rd IEEE International Workshop on Horizontal Interactive Human Computer Systems*, 2008. TABLETOP 2008, 49-56.
27. Olson, J., Olson, G., Strorrosten, M., and Carter, M. 1992. How a group-editor changes the character of a design meeting as well as its outcome. In *Proceedings of the ACM Conference on Computer-Supported Cooperative Work*. 91-98.
28. Patten, J., Ishii, H., Hines, J., Pangaro, G. 2001. Sensetable: A wireless object tracking platform for tangible user interfaces. In *CHI '01: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 253-260.
29. Rogers, Y. and Rodden, T. 2004. Configuring spaces and surfaces to support collaborative interactions. In O'Hara, K., Perry, M., Churchill, E. and Russell, D. (eds.) *Public and Situated Displays*. Kluwer Publishers. 45-79.
30. Shannon, C.E. 1948. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27, 379-423, 623-656, July, October. 1948.
31. Shen, C., Lesh, N. and Vernier, F. 2003. Personal Digital Historian: Story sharing around the table. *Interactions*, 10, No. 2, 15-22.
32. Shneiderman, Ben. 1983. Direct Manipulation: A step beyond programming languages. *IEEE Computer*, 16, 57-69.
33. Sun, C., Xia, S., Sun, D., Chen, D., Shen, H. and Cai, W. 2006. Transparent Adaptation of Single-User Applications for Multi-User Real-Time Collaboration. In *ACM Transactions on Computer-Human Interaction, 13*, No. 4. 531-582.
34. Ullmer, B. and Ishii, H. 2001. Emerging Frameworks for Tangible User Interfaces. In *Human-Computer Interaction in the New Millennium*, ed. John M. Carroll, Addison-Wesley. 579-601.
35. Westerman, W. 1999. Hand tracking, finger identification, and chordic manipulation on a multi-touch surface. University of Delaware PHD thesis.
36. Wilson, A. D. 2005. PlayAnywhere: a compact interactive tabletop projection-vision system. In *UIST '05: Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology*. 83-92.

Interfaces

Enabling mobile micro-interactions with physiological computing.

By Desney Tan, Dan Morris, and T. Scott Saponas

DOI: 10.1145/1764848.1764856

We have continually evolved computing to not only be more efficient, but also more accessible, more of the time (and place), and to more people. We have progressed from batch computing with punch cards, to interactive command line systems, to mouse-based graphical user interfaces, and more recently to mobile computing. Each of these paradigm shifts has drastically changed the way we use technology for work and life, often in unpredictable and profound ways.

With the latest move to mobile computing, we now carry devices with significant computational power and capabilities on our bodies. However, their small size typically leads to limited interaction space (diminutive screens, buttons, and jog wheels) and consequently diminishes their usability and functionality. This presents a challenge and an opportunity for developing interaction modalities that will open the door for novel uses of computing.

Researchers have been exploring small device interaction techniques that leverage every available part of the device. For example, NanoTouch, developed by Patrick Baudisch and Gerry Chu at Microsoft Research, utilizes the backside of devices so that the fingers don't interfere with the display on the front [**2**] (see also in this issue "My New PC is a Mobile Phone," page 36). In more conceptual work, Ni and Baudisch explore the advent of "disappearing mobile devices" (see [**7**]).

Other researchers have proposed that devices should opportunistically and temporally "steal" capabilities from the environment, making creative use of existing surfaces already around us [**9**]. One example of this type of interaction is Scratch Input, developed by Chris Harrison and Scott Hudson of Carnegie Mellon's HCI Institute. This technique allows users to place devices on ordinary surfaces, like tables, and then use them as ad hoc gestural finger input canvases. This is achieved with a microphone on the underside that allows the device to sense audio signals transmitted through the material, like taps and scratches [**4**]. These types of solutions work really well in situations where the user is situated (in an office, airport, hotel room), but is impractical when the user is on the go.

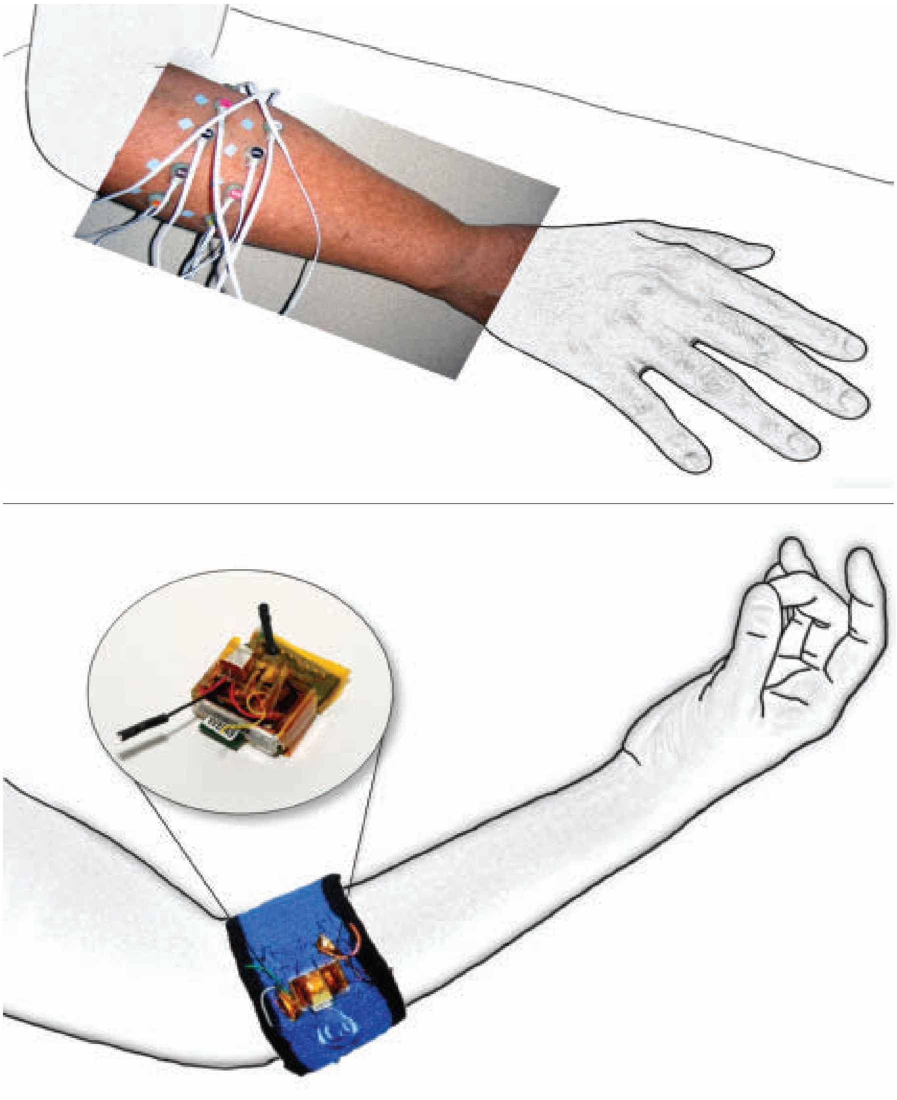
This mobile scenario is particularly challenging because of the stringent physical and cognitive constraints of interacting on-the-go. In fact, Antti Oulasvirta and colleagues showed that users could attend to mobile interaction bursts in chunks of about 4 to 6 seconds before having to refocus attentional resources on their real-world activity (see [**8**] for the full write up). At this point, the dual task becomes cognitively taxing as users are constantly interrupted by having to move focus back and forth. In a separate line of work, Daniel Ashbrook of Georgia Institute of Technology measured the overhead associated with mobile interactions and found that just getting a phone out of the pocket or hip holster takes about 4 seconds and initiating interaction with the device takes another second or so [**1**]. They propose the concept of micro-interactions—interactions that take less than 4 seconds to initiate and complete, so that the user can quickly return to the task at hand. An example of this type of interaction is Whack Gestures [**6**], created by Carnegie Mellon and Intel Labs researchers, where quite simply, you do things like whack the phone in your pocket to silence an incoming phone call.

"Micro-interactions could significantly expand the set of tasks we could perform on-the-go and fundamentally alter the way we view mobile computing."

We believe that such micro-interactions could significantly expand the set of tasks we could perform on-the-go and fundamentally alter the way we view mobile computing. We assert that while seemingly subtle, augmenting users with always-available micro-interactions could have impact on the same magnitude that mobile computing had on enabling a set of tasks

on the Go

Figure 1: To contract a muscle, the brain sends an electrical signal through the nervous system to motor neurons, which then transmit electrical impulses to adjoining muscle fibers, causing them to contract. Electromyography (EMG) senses this muscle activity by measuring the electrical potential between a ground electrode and a sensor electrode.



that were never before possible. After all, who would have imagined mobile phonse would make the previously onerous task of arranging to meet a group of friends for a movie a breeze? Who would have imagined when mobile data access became prevalent that we'd be able to price shop on-the-fly? Or resolve a bar debate on sports statistics with a quick Wikipedia search? Imagine what we could enable with seamless and even greater access to information and computing power.

To realize this vision, we've been looking at ways to enable micro-interactions. Often, this involves developing novel input modalities that take advantage of the unique properties of the human body. In this article, we describe two such technologies: one that senses electrical muscle activity to infer finger gestures, and the other that monitors bio-acoustic transmissions through the body, allowing the skin to be turned into a finger-tap-sensitive interaction surface. We conclude with some of the challenges and lessons learned in our work using physiological sensing for interaction.

MUSCLE-COMPUTER INTERFACES

Removing manipulation of physical transducers does not necessarily preclude leveraging the full bandwidth available with finger and hand gestures. To date, most efforts at enabling implement-free interaction have focused on speech and computer vision, both of which have made significant strides in recent years, but remain prone to interference from environmental noise and require that the user make motions or sounds that can be sensed externally and cannot be easily concealed from people around them.

Advances in muscular sensing and processing technologies provide us with the unprecedented opportunity to interface directly with human muscle activity in order to infer body gestures. To contract a muscle, the brain sends an electrical signal through the nervous system to motor neurons, which then transmit electrical impulses to adjoining muscle fibers, causing them to contract and the body to move. Electromyography (EMG) senses this muscle activity by measuring the electrical potential between ground and a sensor electrode.

"With the latest move to mobile computing, we now carry devices with significant computational power and capabilities on our bodies."

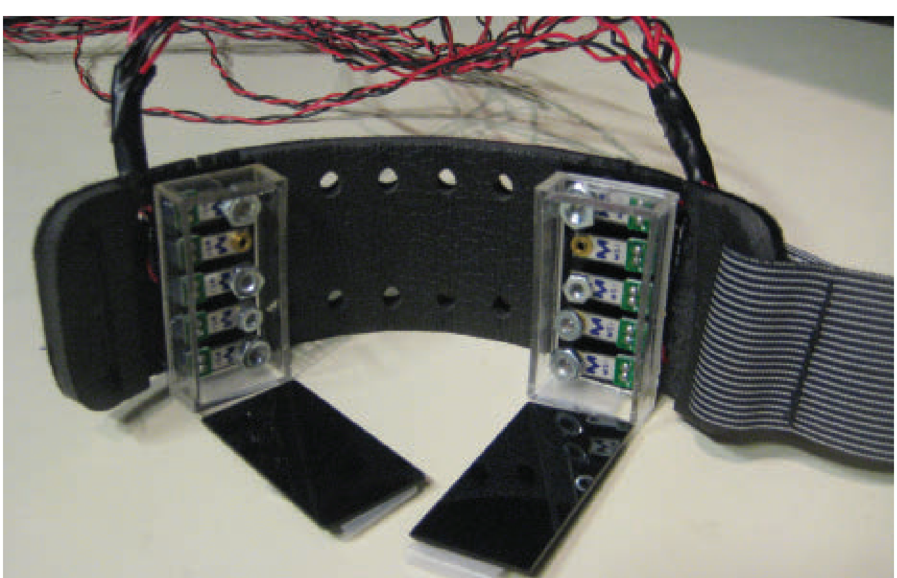
In our work, we focus on a band of sensors placed on the upper forearm that senses finger gestures on surfaces and in free space (see Figures 1 and 2). We have recently built a small, low-powered wireless prototype EMG unit that uses dry electrodes and that can be placed in an armband form factor, making it continuously wearable as an always-available input device. The signals from this device are streamed to a nearby computer, where features are extracted and machine learning used to model and classify gestures. However, this could also be done entirely on a mobile device.

Reasonably high accuracies can be achieved for gestures performed on flat surfaces. In one experiment with 13 novice users, we attained an average of 78 percent accuracy for sensing whether each of two fingers is curled, 84 percent for which of several pressure levels are being exerted on the surface, 78 percent for which of the five fingers have tapped the surface, and 95 percent for which of the five have lifted off the surface.

Similarly, in a separate test with 12 different novice users, we attain 79 percent classification accuracy for pinching the thumb to fingers in free space, 85 percent when squeezing different fingers on a coffee mug, and 88 percent when carrying a bag. These results demonstrate the feasibility of detecting finger gestures in multiple scenarios, and even when the hands are otherwise occupied with other objects.

For more details about this work, see [**10,11,12**]

Figure 2: Our prototype features two arrays of sensing elements incorporated into an armband form factor. Each element is a cantilevered piezo film tuned to respond to a different, narrow, low-frequency band of the acoustic spectrum.



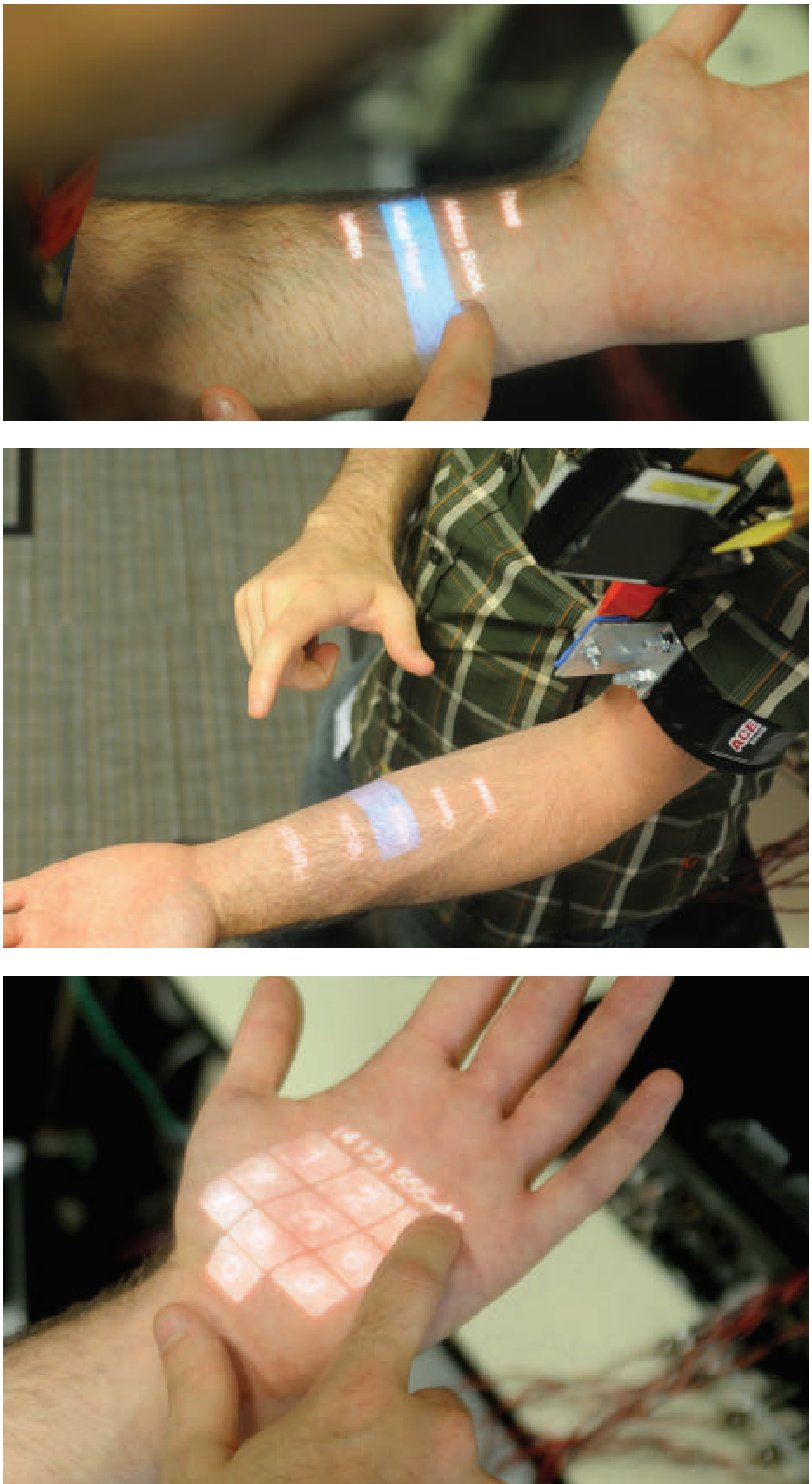
BIO-ACOUSTIC SENSING

To further expand the range of sensing modalities for always-available input systems, we developed Skinput (see Figure 3), a novel input technique that allows the skin to be used as a finger input surface. When a finger taps the skin, several distinct forms of acoustic energy are produced and transmitted through the body. We chose to focus on the arm, although the technique could be applied elsewhere. This is an attractive area to "steal" for input as it provides considerable surface area for interaction, including a contiguous and flat area for projection.

Using our prototype, we've conducted several experiments that demonstrate high classification accuracies even with a large number of tap locations. This remains true even when the sensing armband was placed above the elbow (where taps are both separated in distance and by numerous joints). For example, for a setup in which we cared to distinguish between taps on each of the five fingers, we attain an average accuracy of 88 percent across our 13 novice participants. If we spread the five locations out across the whole arm, the average accuracy goes up to 95 percent. The technique remains fairly accurate even when users are walking or jogging. Although classification is not perfect—nor will it likely ever be—we believe the accuracy of our proof-of-concept system clearly demonstrates that real-life interfaces could be developed on top of the technique.

While our bio-acoustic input approach is not strictly tethered to a particular output modality, we believe the sensor form factors we explored could be readily coupled with a small digital projector. There are two nice properties of wearing such a projection device on the arm: 1) the arm is a relatively rigid structure—the projector, when attached appropriately, will naturally track with the arm; 2) since we have fine-grained control of the arm, mak-

Figure 3: Augmented with a pico-projector, our sensing armband allows interactive elements to be rendered on the skin, potentially enabling a new class of mobile computing experiences



ing minute adjustments to align the projected image with the arm is trivial (e.g., projected horizontal stripes for alignment with the wrist and elbow).

CHALLENGES AND OPPORTUNITIES

Using the human body as the interaction platform has several obvious advantages. Foremost, it is great that we can assume a consistent, reliable, and always-available surface. We take our bodies everywhere we go (or rather it takes us). Furthermore, we are intimately familiar with our bodies, and proprioceptive senses allow us to interact even in harsh circumstances (like a moving bus). We can quickly and easily make finger gestures or tap on a part of our body, even when we cannot see it and are on the move.

That said, using the signals generated by or transmitted through the body as a means of intentional control comes with various new challenges and opportunities for innovation. From a technical perspective, building models of these signals that work across multiple users and multiple sessions with minimal calibration is often

"Who would have imagined when mobile data access became prevalent that we'd be able to price shop on-the-fly? Or resolve a bar debate on sports statistics with a quick Wikipedia search? Imagine what we could enable with seamless and even greater access to information and computing power."

ACRONYMS

AZERTY French version of the keyboard layout known in English as QWERTY

BCI Brain-Computer Interface: tech that reads your mind!

Bio-Acoustics Sounds produced by the body, such as touching one's arm,which computers can distinguish

EEG Electroencephalo-graphy: a BCI that measures electrical signals from sensors on your scalp

GUI Graphical user interface

HCI Human-computer interaction

QWERTU Eastern European version of QWERTY

QWERTZ Central European version of QWERTY

SSVEP Steady state visually evoked potentials: predictable brain responses to visuals that an EEG can detect

Super Pens Pens augmented with special hardware, such as cameras

TUI Tangible user interface

WIMP Windows, icons, menus, pointers—the typical way we interact with a GUI

WYSIMOLWYG What you see is more or less what you get

challenging. Most of our current work is calibrated and trained each time the user dons the device, and while these individual models work surprisingly well across different body types, we recognize that this overhead of training is not acceptable for real world use. Furthermore, regardless of universality of the models, processing the often-noisy signals coming from these sensors is not trivial and will likely never yield perfect results. This is true because of the complexity of the noise patterns as users move through different environments, perform different tasks, and as the physiological signals changes throughout the course of their normal activities. Hence, interaction techniques must be carefully designed to tolerate or even take advantage of imperfect interaction input.

On the interaction design front, there are many problems that must be addressed. For example, the system must provide enough affordances that the user can learn the new system. This is not specific to physiological sensing, though the level of indirect interpretation of signals can sometimes make end-user debugging difficult, especially when the system does not act as it is expected to. The interface must also be designed to handle the "midas touch" problem, in which interaction is unintentionally triggered when the user performs everyday tasks like turning a doorknob. We have purposely designed our gesture sets in order to minimize this, but we imagine there are more graceful solutions.

In fact, with many interaction modalities, our first instinct is often to emulate existing modalities (e.g., mouse and keyboard) and use it to control existing interfaces. However, the special affordances found in the mobile scenario bring with it enough deviations from our traditional assumptions that we must be diligent in designing for it. We should also emphasize the importance of designing these systems so that they operate seamlessly with other modalities and devices that the user carries with them.

Biographies

Desney Tan is a senior researcher at Microsoft Research, where he manages the Computational User Experiences group in Redmond, Washington and the Human-Computer Interaction group in Beijing, China. He has won awards for his work on physiological computing and healthcare, including a 2007 MIT TR35 Young Innovators award, SciFi Channel's Young Visionaries at TED 2009, and named to Forbes' Revolutionaries list in 2009. He will chair the CHI 2011 Conference, which will be held in Vancouver, BC.

Dan Morris is a researcher in the Computational User Experiences group in Microsoft Research. His research interests include computer support for musical composition, using physiological signals for input, and improving within-visit information accessibility for hospital patients. Dan received his PhD in Computer Science from Stanford University in 2006.

T. Scott Saponas is a PhD candidate in the Computer Science and Engineering department at the University of Washington. His research interests include Human-Computer Interaction (HCI), Ubiquitous Computing (UbiComp), and Physiological Computing. Scott received his B.S. in Computer Science from the Georgia Institute of Technology in 2004.

References

1. Ashbrook, D. L., Clawson, J. R., Lyons, K., Starner, T. E., and Patel, N. Quickdraw: the impact of mobility and on-body placement on device access time. In *Proceedings of CHI 2008*, 219-222.
2. Baudisch, P. and Chu, G. Back-of-device interaction allows creating very small touch devices. In *Proceedings of CHI 2009*, 1923-1932.
3. Benko, H., Saponas, T. S., Morris, D., and Tan, D. 2009. Enhancing input on and above the interactive surface with muscle sensing. In *Proceedings of Interactive Tabletops and Surfaces*.
4. Harrison, C., and Hudson, S.E. Scratch Input: Creating large, inexpensive, unpowered and mobile finger input surfaces. In *Proceedings of UIST 2008*, 205-208.
5. Harrison, C., Tan, D., and Morris, D. Skinput: Appropriating the body as an input surface. To appear in *Proceedings of CHI 2010*.
6. Hudson, S. E., Harrison, C., Harrson, B. L., LaMarca, A. 2010. Whack gestures: Inexact and inattentive interaction with mobile devices. In *Proceedings of the 4th International Conference on Tangible, Embedded and Embodied Interaction* (Cambridge, MA, January 25 - 27, 2010). TEI '10. ACM, New York, NY.
7. Ni, T. and Baudisch, P. 2009. Disappearing mobile devices. In *Proceedings of UIST 2009*, 101-110.
8. Oulasvirta, A., Tamminen, S., Roto, V., and Kuorelahti, J. Interaction in 4-second bursts: The fragmented nature of attentional resources in mobile HCI. In *Proceedings of CHI 2005*, 919-928.
9. Pierce, J.S., and Mahaney, H.E. 2004. Opportunistic annexing for handheld devices: Opportunities and challenges. *Human-Computer Interface Consortium*.
10. Saponas, T. S., Tan, D. S., Morris, D., and Balakrishnan, R. Demonstrating the feasibility of using forearm electromyography for muscle-computer interfaces. In *Proceedings of CHI 2008*, 515-524.
11. Saponas, T. S., Tan, D. S., Morris, D., Balakrishnan, R., Turner, J., and Landay, J. A. Enabling always-available input with muscle-computer interfaces. In *Proceedings of UIST 2009*, 167-176.
12. Saponas, T. S., Tan, D. S., Morris, D., Balakrishnan, R., Turner, J., and Landay, J. A. Making muscle-computer interfaces more practical. To appear in *Proceedings of CHI 2010*.

My New PC is a Mobile Phone

Techniques and devices are being developed to better suit what we think of as the new smallness.

By Patrick Baudisch and Christian Holz

DOI: 10.1145/1764848.1764857

The most popular computational device in the world is neither a desktop computer nor a netbook nor a hundred-dollar laptop—it is the mobile phone.

More than 4 billion mobile phones are in use worldwide today. Unlike any other computational device on the market, mobile phones have a very large user base, which includes people across the world, from developed countries to developing nations, and among both urban and rural populations. It is equally popular between people of different ages, both young and adult. The mobile phone's popularity creates a vast range of new use cases and scenarios that need to be designed for.

On the one hand, mobile devices allow PC users to undertake a broader and broader range of activities on the road, disconnected from the wired world. Most modern devices allow interactive web browsing, as well as the viewing and editing of documents, spreadsheets, images, and so on.

On the other hand, mobile devices are the most likely devices to keep people connected to the digital world. With widespread availability and low production costs, mobile phones are on their way to becoming the mass computation platform of the future, a task that neither desktop computers nor netbooks have succeeded in doing so far.

THE CHALLENGE

The role of mobile devices as desktop replacements and as terminals to the digital world requires new categories of mobile applications, ones that will allow users to not only view the data, but also analyze and manipulate it. This varies from editing simple text documents to complex processing of data.

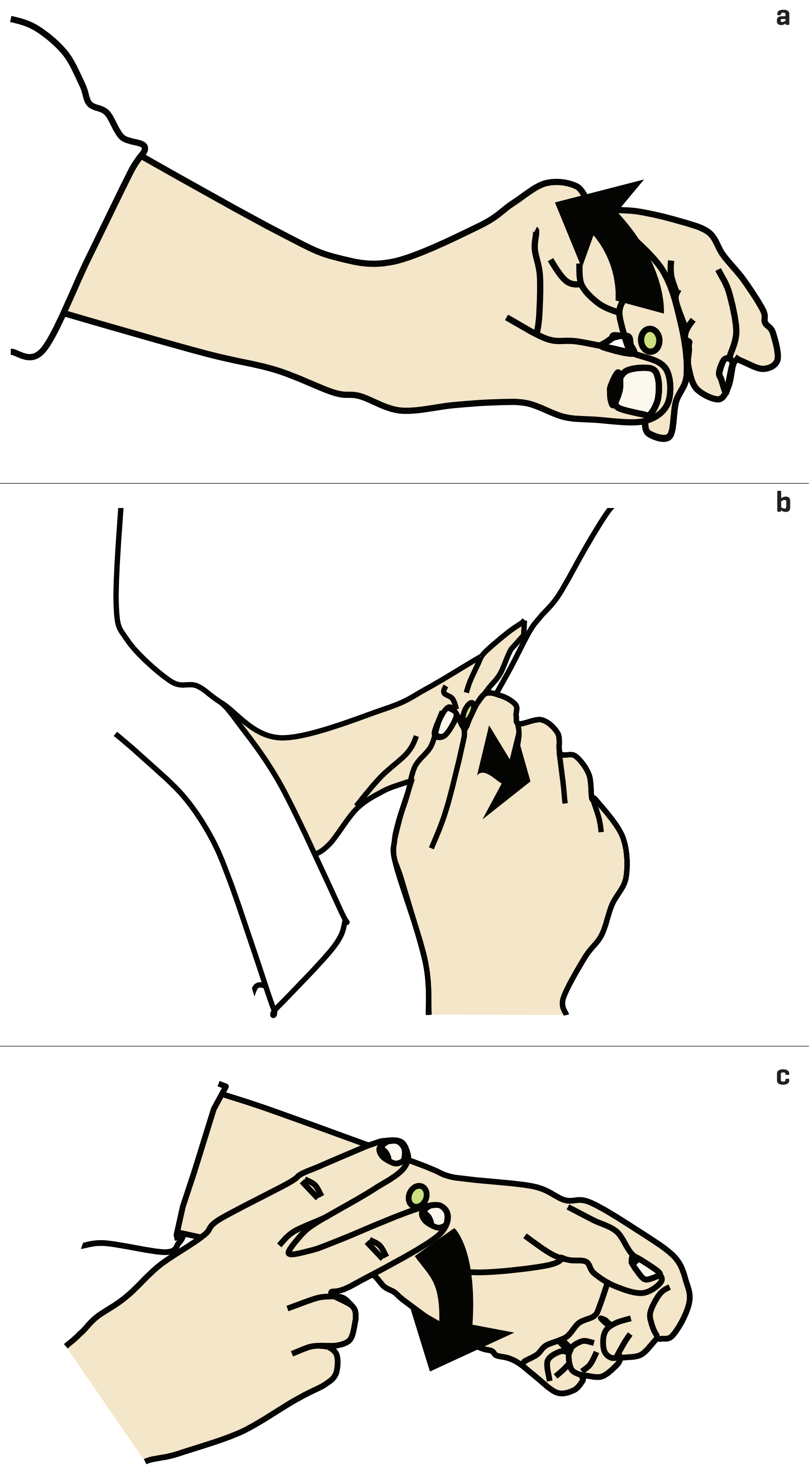
That these applications are still missing on today's mobile devices is the result of the limited size of these devices and the result of human factors. Because human eyesight is limited, a screen of a certain size can communicate only a certain amount of information. Because fingers have a certain size, a mobile keyboard or touch screen can offer only so many controls or buttons.

To eventually enable these complex applications, a lot of current research revolves around the very basics of the interaction: input and output. The eventual goal is to create interaction models that will evade the constraining human factors discussed above.

"Precise input on small devices opens up a large space of new device designs."

To overcome the limitations of displaying output to the users on tiny screens with limited screen size and resolution, much research has taken place. For example, researchers have created systems that provide display space on demand using projection, such as the Sixth Sense system. To keep the device truly mobile, this projection mechanism requires a flat surface at all times and at all places. Other researchers have instead built on zooming, panning, and scaling techniques. Ka Ping Yee's Peephole Display allows users to navigate a virtual world by moving a device in the physical world. The underlying concept allows users to leverage landmarks in the physical world to return to the associated locations in the virtual world. Summary Thumb-

Figure 1: Gesture input allows for input on the tiniest of mobile devices (a) on the finger, providing tactile feedback, (b) in the earlobe, providing auditory feedback, or (c) on the wrist, providing visual feedback. The user is entering a "2" by "scanning" two fingers (see Ni and Baudish's Disappearing Mobile Devices for more [5]).



nails are miniature views of a web page that keep font size readable by cropping text rather than scaling it. Off-screen visualization techniques, such as Halo and Wedge virtually extend the screen by leveraging off-screen space.

In this article, however, we focus on the other aspect of the challenge: input-related issues.

GESTURE INPUT

Gesture input bypasses the lack of input space by using the device as a whole. Users either move the device, which is tracked by an accelerometer present in the device (for example, Rekimoto's Tiltable Interfaces), or the users move their hands in front of the device, as in the Gesture Pendant by Thad Starner and colleagues. By performing gestures right on the surface of the device, gesture input can be brought to the tiniest form factors (**Figure 1**). Scratch Input by Harrison and Hudson [**3**] is basically a gesture interface—it allows users to enter commands by scratching on arbitrary surfaces, which is sensed using a microphone.

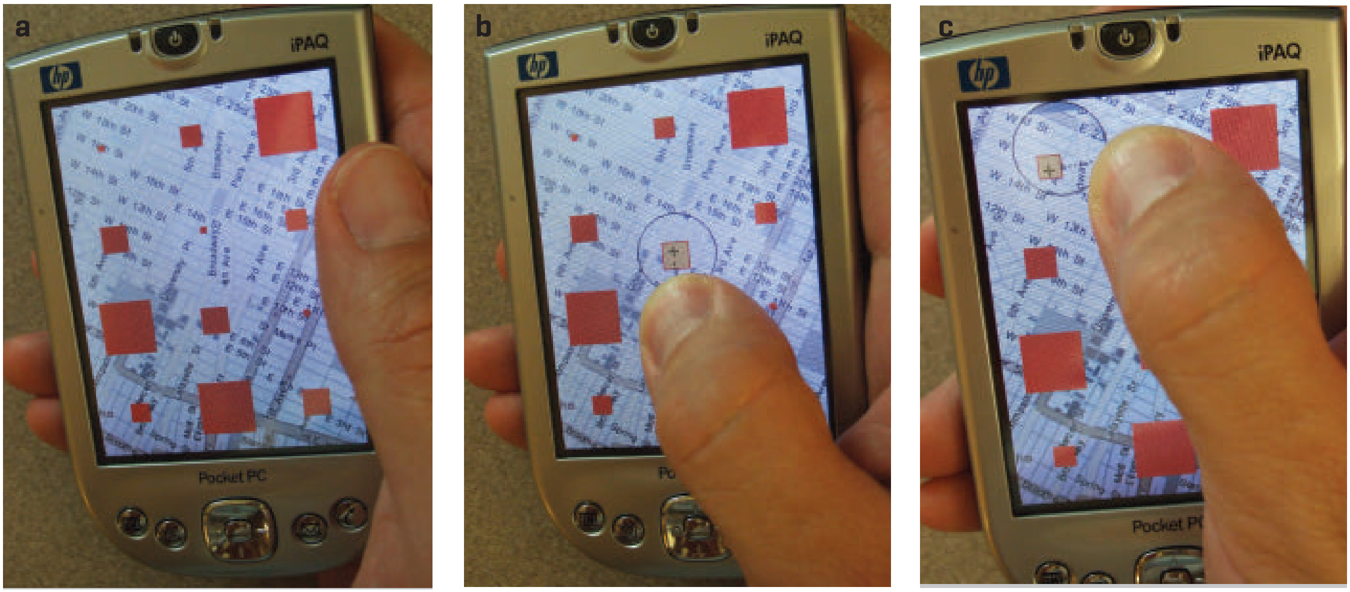
POINT, TOUCH, AND FAT FINGERS

On the flip side, gesture interaction is disjointed from the output space. Many complex desktop applications allow users to manipulate data, but only after having selected them, either through a pointing device, like a mouse, or through a keyboard. If we want to bring such applications to mobile devices, we need to be able to select these objects on the screen.

Miniature joysticks, still common on mobile phones, let the user select objects but with very limited pointing accuracy and abilities. Touch screens on the other hand offer much better performance. In addition to the ease of use, they are well-suited for mobile devices since they can integrate the input medium and the output screen into the same physical space, thus allowing for physical compactness of the device.

The opposite is true, however, because of the so-called fat finger problem. The softness of the user's skin causes the touch position to be sensed anywhere within the contact area between the user's fingertip and the device. Not only that, the relatively larger finger compared to that of the screen

Figure 2: (a) Small targets are occluded by a user's finger. (b) Shift reveals occluded screen content in a callout displayed above the finger. This allows users to fine tune with take-off selection. (c) By adjusting the relative callout location, Shift handles targets anywhere on the screen.



causes the finger to occlude the target. This prevents the target from providing visual feedback, preventing users from compensating for the randomness.

As a result of the fat finger problem, today's touch screen devices are therefore not smaller than their joystick-based predecessors, but actually larger.

Several researchers have proposed techniques and devices that resolve the fat finger problem by physically separating the input and the output space. The first technique based on this idea was the Offset Cursor designed by Potter and Shneiderman and published in 1988. In their design, when a user touches the screen, a pointer appears half an inch above the contact point. The user would move the pointer over the target and select it by lifting the finger off. Offset Cursor resolved the occlusion issue and was the first technique to allow for accurate input on touch screen devices that had previously been considered inherently inaccurate.

However, Offset Cursor has a number of limitations. For example, it does not allow selecting the contents at the very bottom of the screen. This becomes a particularly big issue when applied to the tiny screens of today's mobile devices.

We addressed this and other shortcomings with the Shift technique [**6**], shown in **Figures 2(a)** and **2(b).** While Offset Cursor requires users to aim below the target, Shift lets them aim at the target itself—a benefit in itself, as it reestablishes the direct touch affordance of touch screens. It then reveals the occluded screen content in a callout displayed above the finger. This allows Shift to handle targets anywhere on the screen by adjusting the relative callout location (**Figure 2 (c)**).

However, the Shift Cursor technique has its limitations as well. The callout mechanism limits the dragging of the pointer. The Shift technique also does not work well on very small devices—the smaller the device, the larger the finger in proportion and the harder it is to find space to place the callout.

"The role of mobile devices as desktop replacements requires new categories of mobile applications, ones that will allow users to not only view the data, but also analyze and manipulate it."

Researchers started exploring other options to use the user's finger as an input device. Sidesight by Butler et al. [**2**] allows users to move their finger next to the device. As a wrist-worn device it effectively creates a virtual touch pad on the user's arm. Sidesight thereby offers the functionality of an offset cursor that extends beyond the edge of the screen.

Abracadabra by Harrison and Hudson follows a similar approach. It magnifies the input space, allowing users to point in the space around the device.

On the flip side, all these techniques affect the registration of input and output space, thereby limiting the users' ability to simply "touch a target."

BACK-OF-DEVICE INTERACTION

To reestablish this registration, we proposed back-of-device interaction. **Figure 3** shows our second prototype called Nanotouch [**1**].

The main idea was to maintain the metaphor of direct touch but by touching the back of the device so that fingers never occlude the target. To allow users to accurately point and touch the target, Nanotouch always shows some representation of the finger on the front-side screen. To help the users learn the new input paradigm, we show an image of an actual finger to first-time users, as shown in Figure 3. For any real application, however, we remove the finger and replace it with a small dot—basically a pointer image—which minimizes occlusion and allows for precise targeting.

The key idea behind the design of any front-side touch or a back-of-device touch must be that the human interaction map to the same region from the user's perspective. Making the device appear transparent in the back-of-device design reinforced this correspondence nicely.

The new input design worked surprisingly well in our experiments. One of the reasons could be that the users are already familiar with this notion from activities that they perform using a mirror. When shaving or applying makeup, users perceive the "virtual person" in the mirror as facing them, yet interact backwards.

"Mobile phones are on their way to becoming the mass computation platform of the future."

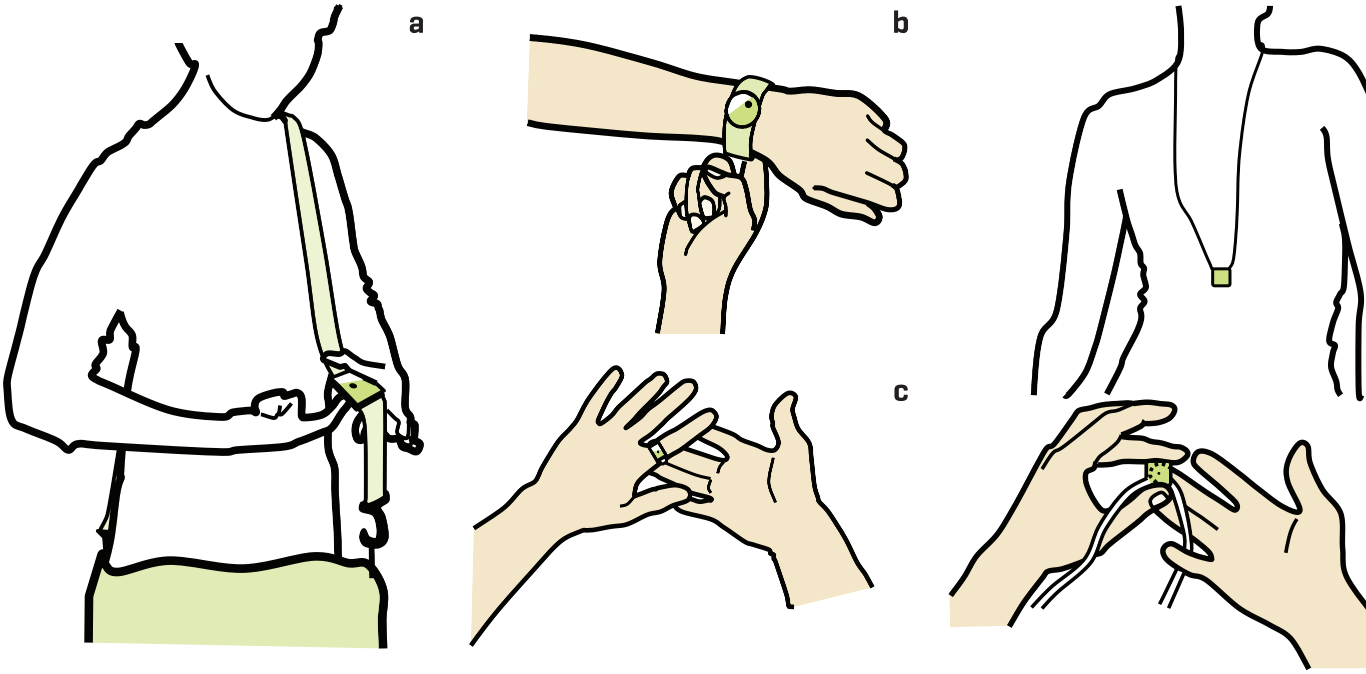
Interaction with Nanotouch also turned out to be highly precise and in a user study, participants were able to acquire a 2.8mm target with 98 percent accuracy. More importantly, back-of-device interaction works practically independent of the device size. In our user study, participants operated Nanotouch with a simulated screen with a diagonal size of only 8mm.

Precise input on small devices opens up a large space of new device designs, including the ones shown in **Figure 4**. All device concepts follow the same basic design: the front-side is for the display and the backside is for the touch input. The edges hold a set of buttons with specific functions. The notion of "back of the device" leaves us with some flexibility, such as in the case of the watch, where the back of wristband serves as the back of the device.

Still, the translation from front-side interaction to back-of-device leaves space for interpretation. In an informal survey, we asked people to write characters on the back of the device. About 80 percent of participants wrote left-to-right, which is consistent with the front-side experience of eyesight, shoulder motion, and elbow motion. The remaining 20 percent, however,

Figure 3: Users operate Nanotouch using pointing input on the back of the device.

Figure 4: Four of the back-of-device designs we envision: (a) a clip-on device with 2.4-inch screen, (b) a watch with 1.2-inch screen, (c) a ring with a screen diagonal of less than half an inch, and (d) a pendant.



wrote right-to-left, which is consistent with the familiar motion of the wrist.

Back-of-device touch input can enable pointing input on very small screens. However, this entire series of techniques and devices goes back to the fat finger problem, i.e., the understanding that the touch interface is inaccurate. Given that so much work has been done on this model, we felt it was time to go back and verify our underlying assumptions. Surprisingly, we found that the fat finger problem is largely a myth.

DISPROVING 'FAT FINGER' AND REDEEMING FRONT-SIDE TOUCH

We conducted some experiments to verify the existence of the fat finger problem. Subjects repeatedly acquired crosshairs with their index finger. For every trial we logged the resulting contact point, as re-ported by a capacitive touch pad. We expected the contact points to form a single large distribution.

Surprisingly, this was not the case. Contact point distributions turned out to be much smaller than ex-pected, about only a third of the expected size.

Instead, the error generally associated with the fat finger problem turned out to be the result of differences between users and variations in finger postures. During the experiment, we forced users to main-tain a constant finger posture, such as keeping a 45-degree tilt between the finger and the pad. We then varied the angle of the finger. As a result, the contact point distributions moved as shown in **Figure 5(a)**. Each of the five white ovals in the figure is the result of a different finger angle. We found similar shifts in the offsets across users, but the size of the distributions remained small.

This is a surprising observation. The smallness of each of the white ovals suggests that touch is not even close to as inaccurate as it is commonly assumed to be. Instead, the inaccuracy we observe with today's touch devices appears to be the result of overlaying many different contact point distributions, each of which is actually very small.

These observations suggest that the inaccuracy of touch devices can be resolved if a device can identify users and determine the angle between the finger and the pad. We created a series of devices that ex-ploit this insight in order to achieve very high touch accuracy.

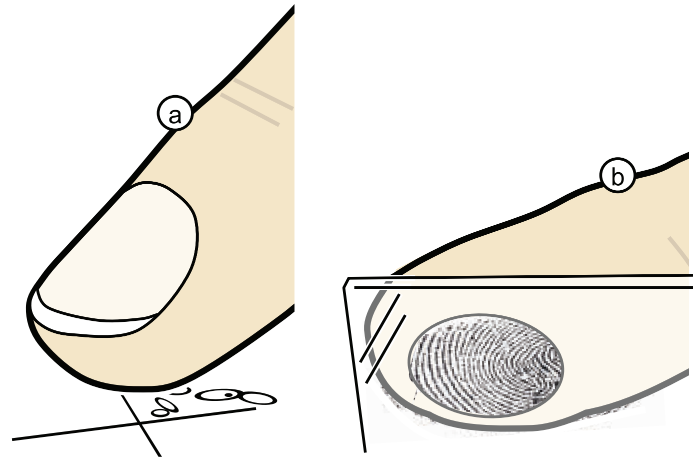
"We need to let go of the notion that the mobile devices are auxiliary devices that we use while on the road."

ACCURATE TOUCH FOR MOBILE DEVICES

Figure 6 shows a setup that implements two of these prototype devices. The cameras surrounding the finger belong to an Optitrack optical tracking system that determine finger angles by observing tiny markers glued to the user's fingernail. The resulting setup allows users to acquire targets of 4.3mm di-ameter with 95 percent accuracy, a 300 percent improvement over traditional touch screens.

However, this setup is hardly mobile. We therefore implemented a second method called RidgePad [**4**], also shown in **Figure 6**. This method is based on the fingerprint scanner in the center of the photo. Un-like a traditional touchpad, the device obtains not only the outline of the contact area between finger and device, but also the fingerprint within this area. By comparing the fingerprint's ridge pattern against samples in the database, the device first determines the user and looks up his or her personal calibration data. The device now determines where the observed part of the fingerprint is located on the user's finger, which allows RidgePad to reconstruct the finger's posture during the touch. By taking this angle into the account, RidgePad is 1.8 times more accurate than traditional touch pads.

Figure 5: (a) A touch input study found that contact points formed much more compact distributions than expected. (b) The RidgePad device exploits the effect that a fingerprint identifies not only a user, but also the angle between finger and device.

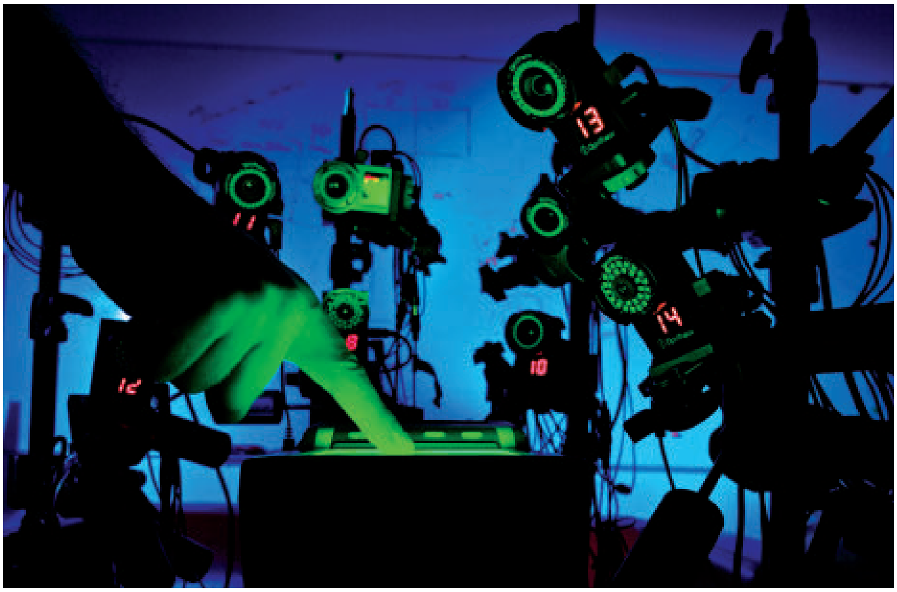


MOBILE PHONES AS PCS

Mobile devices are on the verge of becoming the computational platform of the world. In order to suc-ceed, a wide range of challenges needs to be tackled. We have discussed only on one particular facet: bringing accurate pointing and manipulation to tiny touch screens. This forms the basis for direct manipulation and thus has the potential to opens up mobile devices as a platform for more complex and more interactive applications.

But we have only scratched the surface. In order to tackle the new challenges, we need to make a major conceptual shift. We need to let go of the notion that the mobile devices are auxiliary devices that we use while on the road. Instead, we need to adopt a model in which the mobile devices are the main computational devices, if not the only computational device.

Figure 6: This experimental setup tracks finger angles using an optical tracker. It also implements the RidgePad prototype, which extracts user ID and finger angles from the user's fingerprint.



Biographies

Patrick Baudisch is a professor in Computer Science at Hasso Plattner Institute in Berlin/Potsdam and chair of the Human-Computer Interaction Lab. His research focuses on the miniaturization of mobile devices and touch input. Previously, he worked as a research scientist in the Adaptive Systems and Interaction Research Group at Microsoft Research and at Xerox PARC and served as an affiliate professor in computer science at the University of Washington. He holds a PhD in Computer Science from Darmstadt University of Technology, Germany.

Christian Holz is a PhD student in Human-Computer Interaction at Hasso Plattner Institute in Potsdam, Germany. Previously, he worked as a research scholar at Columbia University. His research focuses on understanding and modeling touch input on very small mobile devices.

Acknowledgements

The authors thank everyone in their lab group at Hasso Plattner Institute and former colleagues at Microsoft Research, in particular Ken Hinckley and Ed Cutrell. They also thank Dan Vogel who worked on Shift, and Gerry Chu who worked on NanoTouch. Finally, they thank the collaborators on back-of-device interaction, in particular, Daniel Wigdor and Cliff Forlines.

References

1. Baudisch, P. and Chu, G. Back-of-device interaction allows creating very small touch devices. In *Proc. CHI'09*. 1923-1932.
2. Butler, A., Izadi, S., and Hodges, S. SideSight: multi-"touch" interaction around small devices. *Proc. UIST '08*. 201-204.
3. Harrison, C. and Hudson, S. (2008). Scratch input: creating large, inexpensive unpowered and mobile finger input surfaces. *Proc. UIST '08*. 205-208.
4. Holz, C. and Baudisch, P. 2010. The Generalized Perceived Input Point Model and How to Double Touch Accuracy by Extracting Fingerprints. To appear in *Proceedings of CHI'10*. 10 pages.
5. Ni, T. and Baudisch, P. Disappearing Mobile Devices. In *Proc. UIST'09*. 101-110.
6. Vogel, D. and Baudisch, P. (2007). Shift: A Technique for Operating Pen-Based Interfaces Using Touch. *Proc. CHI'07*. 657-666.

From Brains to Bytes

Brain-computer interfaces have the potential to change the way we use devices, and there are at least four methods for implementation.

By Evan Peck, Krysta Chauncey, Audrey Girouard, Rebecca Gulotta, Francine Lalooses, Erin Treacy Solovey, Doug Weaver, Robert Jacob

DOI: 10.1145/1764848.17664858

Science fiction has long been fascinated by brain-computer interfaces (BCI)—the use of sensors to identify brain states. From Andre Maurois' 1938 story "The Thought-Reading Machine," in which a professor stumbles on a machine that reads people's thoughts, to the recent blockbuster *Avatar*, where humans control surrogate bodies with their minds, the public is captivated by the interaction between the human brain and the computers created by those brains.

Although most people are likely to conjure images of Neo's frightening "head port" from *The Matrix* before they dream of a university student wearing an elastic cap studded with electrodes, the media has closely followed less sinister, if also less all-powerful, BCI research. In the past year, University of Wisconsin-Madison's Brain-Twitter interface received *Time Magazine*'s honor as the no. 9 invention of the year. Furthermore, as brain-imaging technology has become more portable and less expensive, the human-computer interaction (HCI) community has begun to bring science fiction closer to reality.

MIND MATTERS

In the larger field of human-computer interaction, we are often concerned with the bandwidth of interaction between a user and the computer. How can we give the computer more information, and more relevant information? How can the computer give us more information without overloading our sensory systems?

Using a mouse in addition to a keyboard increases the bandwidth from the user to the computer by augmenting the type and number of commands the computer can recognize. An application that uses audio increases the bandwidth from the computer to the user, by adding to the type of information the computer can output. Seen in this context, brain-computer interfaces present an opportunity to expand the user-to-computer bandwidth in a unique and powerful way. Instead of identifying explicit actions, we can detect intent. Instead of evaluating action artifacts, we can recognize purpose. Even more interesting, we may be able to understand the user's needs before the user can articulate them.

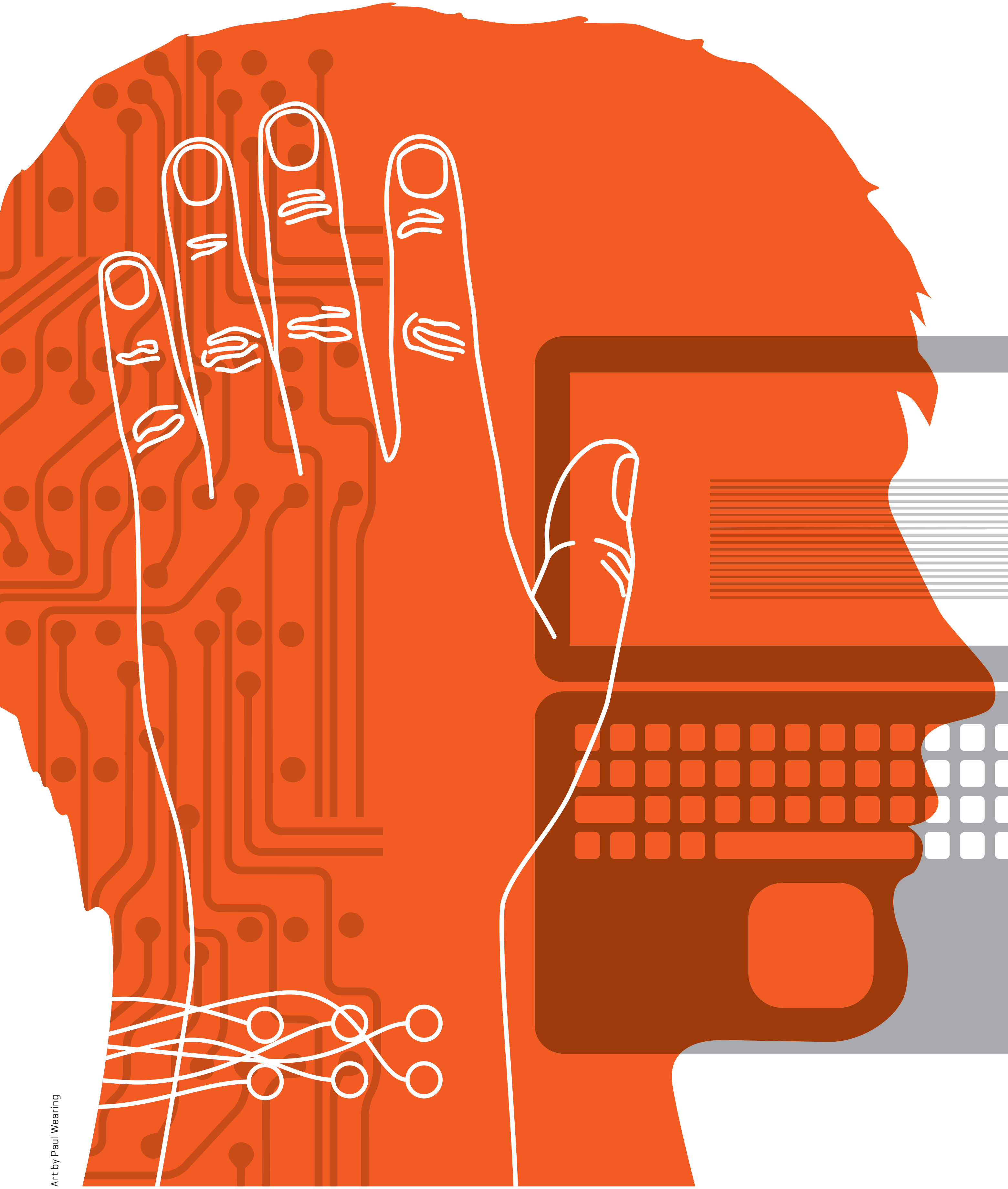
"As brain-imaging technology has become more portable and less expensive, the HCI community has begun to bring science fiction closer to reality."

But this is all far in the future. On the wide continuum between analyzing electroencephalo-graphs to *Avatar* mind-machines, where are we now? And perhaps more importantly, where are we going?

In this article, we will discuss several directions for research into brain-computer interaction, and the relative merits using these brain measurements to give the user direct or passive control of computer interfaces. We will also introduce projects across the world that offer a glimpse into the future of BCIs.

Imagine the following scenario:

It's 9:30 p.m., and you're driving in heavy traffic through Boston, unsure of where you're going. Your phone is alerting you of text messages that your sister is sending every 30 seconds. Your GPS is commanding you to turn in half a mile, then a tenth of a mile, but still, you cannot tell which of the six exits off the rotary to take. To make things worse, the radio commercials are blaring, but you are focused on the road, and



Art by Paul Wearing

uncomfortable taking your hand off the wheel to reach for the volume.

Unlike the head-ports in the fantastical world of The Matrix, this scenario is close to our current reality. Also unlike a fantastical movie world, this is the world in which any brain-computer interface will have to function. But how could a brain-computer interface deal with this situation without adding to the already full spectrum of sensory input? Researchers are taking steps toward answering this question and designing BCIs to meet our needs.

Several approaches to BCIs are being pursued using current brain-sensing technology. These systems detect biological changes occurring naturally during the operator's activity. Interface designers can use this information to deduce the operator's state and translate it into commands that adjust the computer accordingly. Changes to the interface can be the result of a direct—voluntary—input, or a passive measure.

Figure 1: Unlike some BCI systems that require inserting devices directly into the brain, EEG electrode caps are worn externally.

DIRECT CONTROL INTERFACES

Brain interfaces that allow direct control often replace a user's normal motor function (generally used to move mouse cursors or type on a keyboard), and are currently the dominant strain of research in brain-computer interaction. Direct control involves a structured mental activity that results in an explicit command to the computer. To move your mouse to the right, you might imagine moving your hand to the right.

These direct-control interfaces rely on the fact that the brain activity occurring when you move your hand to the right is very similar to the activity that occurs when you imagine moving your hand to the right. This consistency can be used to pair mental "movements" with commands: when participants imagine waving their arms up and down, for example, the volume on their phone might mute, or the zoom level on their screen might change.

Using this mechanism, we can imagine a world for our car scenario in which direct control interfaces are commonly available to everyone.

You have trained yourself to produce specific brain activity to control different technologies in the car (much like how we learn to touch type). Without taking your eyes off the road, you decide to silence your phone, and perform mental arithmetic, which the brain-computer interface recognizes as the command for muting the phone. You imagine swinging your right arm up and down, and the device also recognizes this as a command, turning the radio down. You imagine yourself moving your left pinky, and again, the device recognizes your brain state, redrawing the GPS map in more detail. Through the entire process, your eyes never leave the road, your hands never leave the steering wheel, and you never say a word.

If this sort of control seems contrived, it is because most direct-control interfaces are currently geared toward disabled users, such as paralyzed patients, or people with severe physical disabilities. In disabled users, familiar, physical mental commands can be repurposed to perform other tasks, as these motor skills are not available. However, if we look further into the future, users may not need to perform mental gymnastics, instead using the thought processes that occur in the brain anyway to control external devices.

To survey current direct control research, we further divide the topic into two brain-imaging techniques: invasive and non-invasive. Although brain sensing is not limited to direct control interfaces, we are not aware of any passive systems that use invasive techniques.

INVASIVE

Invasive BCIs involve implanting microelectrodes into the grey matter of the brain during neurosurgery in an effort to capture brain activity more accurately. While significant, successful work is being performed on invasive techniques at research facilities such at Brown University and University Wisconsin-Madison, there are many risks, difficulties and limitations involved. Aside from the difficulty of inserting sensors directly into the brain, there is the risk of scar tissue formation, infection to the brain, and the unknown long-term stability of implanted micro-electrodes. Depending on the task and the participant, accomplishing a task using brain states can require long training, and some users may never reach the desired level of reliability.

Despite this, any improvement in communication is worthwhile and rewarding for paralyzed users, who may not have voluntary control over any muscles at all. Invasive BCIs can provide remarkable new opportunities for disabled users; although currently such systems are limited to fairly rudimentary communication such as choosing pictures, letters, or cursor directions, some potential advantages for future applications include repairing damaged sight and providing new functionality to those with paralysis.

“One recent example of a BCI that uses EEG is a wheelchair that can be controlled through brain activity.”

NON-INVASIVE

Non-invasive direct control technologies, unlike their invasive counterparts, use external systems, such as electroencephalography (EEG) or functional near-infrared spectroscopy (fNIRS), to measure brain activity. See **Figure 1** for an example. Non-invasive imaging technologies are commonly used in a number of fields, including medicine and psychology, and provide valuable insight into brain activity without requiring surgery or implantation. As a result, they are an attractive option for researchers who want portable and flexible systems for measuring and interpreting this data.

EEG measures brain activity using electrodes placed on the scalp that detect electrical potential caused by neurons firing, providing researchers with information about activation in numerous regions of the brain. The complexity of these systems varies with the number of electrodes used and the techniques to process the data captured.

One advantage of this system is that the data recorded has a high temporal resolution. The system can detect brief changes in brain activity, in the millisecond realm. Additionally, it is possible to buy EEG systems that are small, lightweight, and portable. However, a number of limitations affect the utility of the data collected from EEG systems. For example, the EEG system is sensitive to muscle contraction, limiting the user’s physical movements during cognitive activity measurements. Additionally, these systems have low spatial resolution. It can be difficult or impossible to determine which precise region of the brain is being activated.

There are many successful direct control paradigms using EEG signal, which include visually evoked potentials (SSVEP), P300-based BCI, motor imagery, event-related synchronization and slow cortical potentials. A great example of work in this direction can be seen by reading Wolpaw et al.’s 2002 article in Clinical Neurophysiology [**3**]. The P300-based BCI allows selection by taking advantage of brain activity, called a P300, that occurs when your intended target is highlighted. This has been used successfully to create a spelling application.

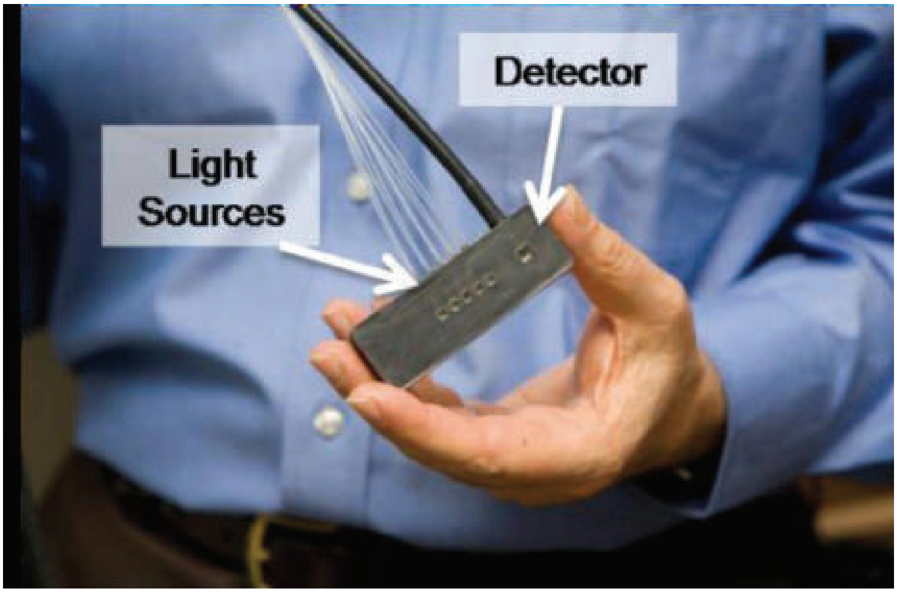
One recent example of a BCI that uses EEG is a wheelchair that can be controlled through brain activity, created by Brice Rebsamen et al. The researchers created a list of paths to locations in a small apartment and then presented those target locations to users. To select a target, the users were instructed to focus on that target when it was presented to them. After several minutes of training with a participant, the system could detect the desired location with almost perfect accuracy.

Clearly, there are limitations to this system, such as predetermining a list of routes and targets, which may not be possible in large or complex environments. However, this work is an example of the possibilities for EEG systems to be developed and incorporated into more sophisticated technologies.

Functional near-infrared spectroscopy (fNIRS) is a vastly different technology than EEG in that it measures blood flow changes instead of electrical activity. It uses optical wires to emit near-infrared light, which is then refracted from the tissue of the head, including the brain (see **Figure 2** for an example of the device). Sensors in the system detect changes in oxygenated and deoxygenated blood in that region [**1**]. This technology is marked by a high degree of spatial resolution but is less temporally sensitive to changes in brain activity than EEG.

The brain data recorded by fNIRS is less susceptible to movement artifacts and can be used in combination with computer usage. fNIRS has another notable advantage: it has a shorter setup time than EEG, which makes it a more practical option for use in research, government work, and commercial applications. Additionally, the part of the fNIRS system placed on the scalp or

Figure 2: An fNIRS sensor with five light sources and one detector.



forehead is typically small and therefore less bothersome to users than other brain measurement technologies.

Ruldolph L. Mappus IV and his colleagues at the Georgia Institute of Technology have used fNIRS to develop a technology for drawing letters using data collected from brain activity. The subjects were instructed to trace the letters "C" and "J" by performing different mental tasks to control the drawing direction. The researchers noted some difficulties with their design, including the slow fNIRS response time for changes in mental activity. However, the researchers indicated that, in the future, they believe that this work can be expanded to provide users with a broad range of tools to draw using fNIRS technologies.

"Brain-computer interfaces present an opportunity to expand the user-to-computer bandwidth... Instead of identifying explicit actions, we can detect intent. Instead of evaluating action artifacts, we can recognize purpose."

PASSIVE BCIS

The previous examples have helped illustrate direct control BCIs, which use brain activity as the primary input device, but which often require considerable user training to generate specific brain states. However, these technologies have a reduced relevance in the ordinary computer environment. A healthy student has little need to hook herself up to an EEG in order to move a cursor. It's easier, faster, and less error-prone to simply use a mouse. With this in mind, our research at Tufts University has turned to passive BCIs, interfaces that detect brain activity that occurs naturally during task performance [2]. Passive BCIs focus on the brain as a complementary source of information, an additional input used in conjunction with conventional computer inputs such as the mouse or keyboard.

Generally, passive BCIs use non-invasive measuring techniques such as EEG and fNIRS. fNIRS, as previously mentioned, requires little in the way of set up and imposes relatively few physical or behavioral restraints on the user. Because many passive BCIs aim to observe brain signals that can be used in a relatively ordinary computer task environment, these qualities make fNIRS particularly attractive as a measuring tool.

How would a future of passive BCIs impact our previous car example? Imagine now that you're wearing the headband shown in **Figure 3**:

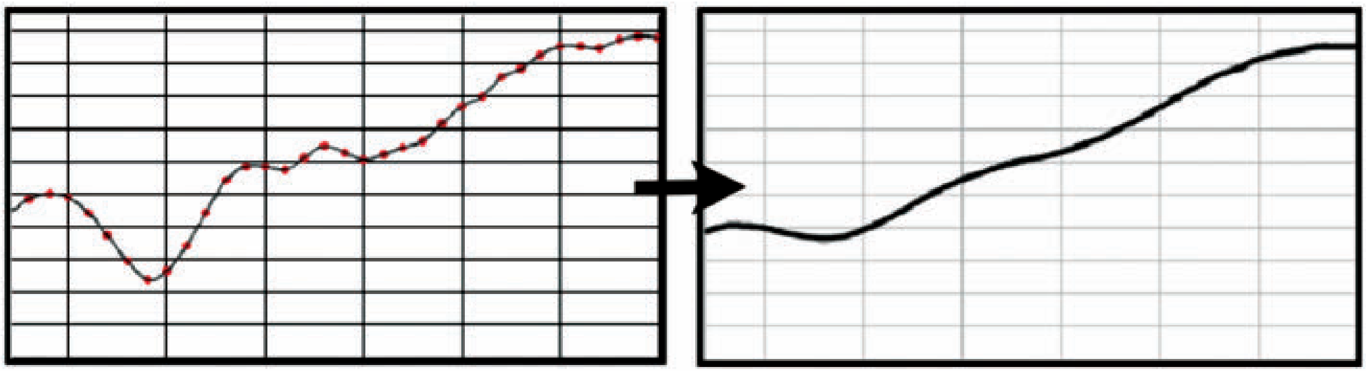
You drive, think, and behave normally, as you would before the BCI was introduced. Brain-sensing devices determine that you are mentally overloaded, juggling the phone, GPS, radio, and driving simultaneously. As a result, the system gently simplifies the map on your GPS. You may not have a clear understanding of the neighborhood, but you won't miss this turn. The system subtly turns down the interior dashboard lights to prevent glare on the windshield. Finally, since the sender of the text messages had previously been designated as low-priority, the system silences the phone, waiting for a better time to let you know about the messages.

The principal advantage of passive BCIs is that they operate using the brain activity that occurs naturally during task performance, and so they do not add to the user's task load (in this example, the driver does not have to think about waving her arms or any other extraneous mental process). Currently, we are working towards a similar class of BCIs to create adaptive environments.

In a recent experiment designed to lead to adaptive interfaces, participants played Pac-Man at two different levels of difficulty, while their brain activity was measured using fNIRS. Statistical analysis of the fNIRS measurements allowed the participants to be classified both by their playing state (playing versus rest) and difficulty level (easy versus hard) with high accuracy. Using this data, the interface could be subtly adapted to the task at hand. If the user is resting, play quiet, soothing

Figure 3: fNIRS sensors placed on the forehead can be non-intrusive when secured by a simple headband.

Figure 4: Stockbrokers might use BCIs to project appropriate market visualization based on how much distraction they can handle at the moment. The image on the left is what a broker might see when engaged in low amounts of work, while the more simplified version might be more appropriate when the mental workload is greater.



music. If the user is playing the game, speed up the pace and volume of music to make the experience more intense and enjoyable. Other research at Tufts has examined fNIRS use in an ordinary computer environment, as well as determined differences in semantic and syntactic workload.

Passive BCIs also lead to a completely different paradigm for interaction with the user. It is no longer acceptable to use bold, explicit reactions to brain states that we make in direct control interfaces. Instead, we find ourselves drawn toward gentle, more implicit changes to the interface. Fading, overlaying information, or changing an application's screen real estate are options that, if done slowly enough, may affect the user's focus in a positive way.

While this type of research is still in its infancy, there are countless examples of possible adaptive interfaces that could be used with fNIRS measurements. For example, at any given moment during the day, a stock broker could be overwhelmed with work or experiencing a lull in activity; either way, he must always be aware of the market. A display that gently changes visualizations of the market according to the user's workload could be invaluable. If the stockbroker is not currently exerting a high mental workload, the display can show his stock visualization as highly detailed, giving the stockbroker as much information as possible. See **Figure 4**. If the stockbroker is working exceptionally hard at his email and cannot afford to be distracted by complex stock information, the display can simply lower the level of detail. In this way, the broker will still recognize major changes in the data without getting bogged down in the details.

We can imagine a class of interfaces that work to benefit the general population, systems that dynamically filter streams of information (Twitter, RSS, email) to accommodate the workload of the user, systems that dynamically adjust teaching methods to best suit a child's learning style.

LOOKING AHEAD

BCIs currently in development have the potential to open worlds of communication and mobility to disabled users; further up the pipeline, BCIs have the potential to adjust our constantly-busy environments to provide us with the best possible chance of completing our tasks. Whether this is successful navigation to a new place, remotely commanding a robot, or buying all the items on a grocery list, BCIs hold the promise of performing tasks or changing environ-ments in the real world with no physical manipulation at all.

Biographies

Evan Peck is a PhD candidate in the Human-Computer Interaction lab at Tufts University. His research focuses on creating adaptive brain-computer interfaces to improve user experience with commonplace computer tasks.

Krysta Chauncey is currently a National Research Council Research Associate at the U.S. Army Natick Soldier Research, Development & Engineering Center in collaboration with the Brain-Computer Interface Project at the Tufts Human-Computer Interface Laboratory. Her primary interest is in applying the methods and knowledge of cognitive neuroscience to the development of brain-computer interfaces, particularly those which use electrophysiological or optical brain data as an auxiliary source of information to improve the user's task performance or experience.

Audrey Girouard is a PhD candidate in Computer Science at Tufts University, specializing in HCI. She is currently fascinated by passive brain computer interfaces, studying the use of brain imagery to enhance HCI for all. She is the laureate of the PostGratudate Scholarship from NSERC.

Rebecca Gulotta is a senior at Tufts University and a research assistant in the Tufts Human-Computer Interaction Lab. She studies Human Factors Engineering and Computer Science.

Francine Lalooses is a computer science doctoral candidate at Tufts University, where she is studying human computer interaction and specifically brain-computer interaction. She currently works at The MITRE Corporation while attending Tufts.

Erin Treacy Solovey is a computer science doctoral candidate at Tufts University, where she is interested in next-generation interaction techniques. She is currently investigating the use of brain sensors to detect signals that users naturally give off in order to augment standard input devices.

Doug Weaver is a master's student at Tufts University. Currently, he is participating in research developing adaptive brain-computer interfaces.

Robert Jacob is a professor of computer science at Tufts University, where his research interests are new interaction media and techniques and user interface software. He was also a visiting professor at the Universite Paris-Sud and at the MIT Media Laboratory, in the Tangible Media Group. He is a member of the editorial board of Human-Computer Interaction and the ACM *Transactions on Computer-Human Interaction*. He was elected to the ACM CHI Academy in 2007.

References

1. Chance, B., Anday, E., Nioka, S., Zhou, S., Hong, L., Worden, K., Li, C., Murray, T., Ovetsky, Y., and Thomas, R. 1998. A novel method for fast imaging of brain function, non-invasively, with light. *Optics Express*, 411-423.
2. Cutrell, E., and Tan, D. S. 2008. BCI for passive input in HCI. Paper presented at the ACM CHI'08 Conference on Human Factors in Computing Systems Workshop on Brain-Computer Interfaces for HCI and Games.
3. Wolpaw, J.R., Birbaumer, N., Mcfarland, D.J., Pfurtscheller, G., and Vaughan, T.M. Brain-computer interfaces for communication and control. *Clinical Neurophysiology, 113* (6). 2002, 767-791.

PROFILE

Photo of ClearBoard by NTT ICC; Inset photo by Webb Chappell.

"By complementing physical objects with digital information, we can improve and augment the way we perform tasks."

Hiroshi Ishii Tangible Bits

BY DAVID CHIU

DOI: 10.1145/1764848.1764859

Hiroshi Ishii sees the world differently. The Massachusetts Institute of Technology professor of media arts and sciences, widely regarded as the pioneer of tangible user interfaces (TUI), is changing the way we interact with our surroundings by integrating computing and physical objects. Specifically, within his Tangible Media Group at MIT, Ishii and his students are looking for ways to tie physical objects to digital information in a vision they call Tangible Bits.

Their vision, which departs from the pervasive "painted bits" within current graphical user interfaces, is led by the observation that humans have developed a lifetime of intuition manipulating objects in the physical world. By complementing physical objects with digital information, we can improve and augment the way we perform tasks.

Before joining MIT Media Labs in 1995, Ishii worked for NTT Human Interface Labs in Japan, where he led a research group toward developing two critical projects: TeamWorkStation and ClearBoard. TeamWorkStation was designed in 1990 to provide real-time sharing of drawing space between geographically disparate collaborators. It was enabled through a translucent video overlay of the collaborators' workspaces. ClearBoard, developed in 1992, allowed for vis-à-vis interaction between two collaborators, and for the first time supported gaze awareness (so that the partner's focus of attention was communicated) over a large, clear screen for drawing. These seminal efforts have since been succeeded by a cornucopia of interface projects under Ishii's lead.

Ishii, who received his PhD in computer engineering in 1992 from Hokkaido University in Japan, recalls the circumstances that led him to his current work. "My father was a programmer of the IBM 360 mainframe when I was a kid, [which] is why I chose computer science." He added that his initial "shock" when he first saw the Xerox Alto (hailed as the first computer with a GUI) back in 1973 was what prompted his interests in HCI. Years later, Ishii is now a leader in tangible user interface research and development. In 2006, Ishii was elected by ACM SIGCHI into the prestigious CHI Academy for his significant contributions to the field.

Certainly, Ishii's success did not come without some initial roadblocks. One of the great challenges he has faced is discovering "compelling applications" to convince people of their vision in well-established HCI conferences, which have traditionally been more focused on user-centered designs. Another ongoing challenge involves the fact that tangible user interfaces often require proprietary and non-standard hardware platforms, but Ishii says he is optimistic about their acceptance in the future.

The growing number of researchers, designers, and artists who are contributing to the field of tangible user interfaces share his optimism. In fact, Ishii refers to the success of the International Conference in Embedded and Embodied Interaction series (TEI), most recently held in January 2010 at MIT Media Labs, as an encouraging sign for the community.

With these challenges currently being addressed and novel high-level concepts coming to fruition, Ishii is prepared to invoke the next big idea. He believes that in the next five to ten years, we can expect to see an integration of manipulatory and ambulatory interfaces as well as "a departure from a table [interface] to an entire room, building, and city." As tangible user interfaces continue to emerge and mature, we can surely expect Ishii to lead this movement.

HELLO WORLD

Real-Time Detection with Webcam

by Dmitry Batenkov

OpenCV is an open-source, cross-platform library for real-time computer vision. Originally developed by Intel, the library will use Intel's Integrated Performance Primitives, if it is found on the system. It is very well-documented with a full reference manual, many examples and tutorials, and a book (which is also a good introduction to computer vision). Interfaces for C, C++, and Python are also available in OpenCV.

Example applications of the OpenCV library include human-computer interaction; object identification, segmentation and recognition; face recognition; gesture recognition; motion tracking, ego motion, motion understanding; structure from motion (SFM); stereo and multi-camera calibration and depth computation; and mobile robotics.

In this tutorial, we will learn how to do real-time face detection using a webcam. We will utilize a machine-learning object detection algorithm known as the Viola-Jones detector. It's a fast classification mechanism using Haar-like wavelet features. OpenCV ships with a very good "classifier file" for faces, but one can also train the classifier to recognize any kind of objects.

Instructions

First, download the latest OpenCV release for your platform from http://opencv.willowgarage.com and install it.

Next, copy the attached program to a file named facedetect.py. You can also download it from http://XRDS.acm.org.

In the downloaded source archive, locate the classifier file data/haarcascades/haarcascade_frontalface_alt_tree.xml and replace the placeholder in the code with this original location.

Make sure that the Python interpreter knows the location for the OpenCV Python bindings. In Linux, it should be set automatically. In Windows, set the environment variable

set pythonpath = <opencvdir>\Python2.6\Lib\site-package.

Now, connect your webcam and run the program: python facedetect.py

To exit, press Esc. Have fun!

Improvements

Once an object is detected, we can start tracking it. OpenCV has an implementation for CamShift tracking algorithm. (See the example on http://XRDS.acm.org.)

Add detection of the eyes, mouth, and so on. (OpenCV ships with corresponding classifiers.) You can recognize emotions! See the video: www.youtube.com/watch?v=V7UdYzCMKvw.

If you replace the face classifier with hands classifier, and add tracking, you can now recognize gestures!

—Dmitry Batenkov

"In this tutorial, we will learn how to do real-time face detection using a webcam. We will utilize a machine-learning object detection algorithm known as the Viola-Jones detector."

RESOURCES

Object identification links
Viola-Jones algorithm
http://www.face-rec.org/algorithms/Boosting-Ensemble/16981346.pdf

Haar training tutorial
http://note.sonots.com/SciSoftware/haartraining.html

Haar cascades repository
http://alereimondo.no-ip.org/OpenCV/34

HCI PROJECTS USING OPENCV

HandVu
Gesture recognition
www.movesinstitute.org/~kolsch/HandVu/HandVu.html

EHCI Head tracking
http://code.google.com/p/ehci

PyEyes Eyes tracking
http://eclecti.cc/olpc/pyeyes-xeyes-in-python-with-face-tracking

CCV/touchlib Multi-touch library
http://nuigroup.com

OTHER HCI/CV TOOLKITS

TUIO Common API for tangible multitouch surfaces
www.tuio.org/?software (list of implementations)

Trackmate Do-it-yourself tangible tracking system
http://trackmate.media.mit.edu

Sphinx Speech recognition toolkit
www.speech.cs.cmu.edu/sphinx/tutorial.html

VXL versatile computer vision libraries
http://vxl.sourceforge.net

Integrating Vision Toolkit
http://ivt.sourceforge.net

```
import sys
import cv

storage=cv.CreateMemStorage(0)
image _ scale=1.3
haar _ scale=1.2
min _ neighbors=1
haar _ flags=0

def detect_and_draw(img):
  # allocate temporary images
  gray=cv.CreateImage((img.width,img.height),8,1)
  small _ img=cv.CreateImage((cv.Round(img.width/
  image _ scale),
    cv.Round(img.height/image _ scale)), 8, 1 )

  # convert color input image to grayscale
  cv.CvtColor( img, gray, cv.CV _ BGR2GRAY )

  # scale input image for faster processing
  cv.Resize( gray, small _ img, cv.CV _ INTER _ NN )
  cv.EqualizeHist( small _ img, small _ img )

  # start detection
  if( cascade ):
    faces=cv.HaarDetectObjects( small _ img,
    cascade, storage,
      haar _ scale, min _ neighbors, haar _ flags )
  if faces:
    for (x,y,w,h),n in faces:
    # the input to cvHaarDetectObjects was resized, so scale the
    # bounding box of each face and convert it to two CvPoints
    pt1=(int(x*image _ scale),int(y*image _ scale))
    pt2=(int((x+w)*image _ scale),
    int((y+h)*image _ scale))
    # Draw the rectangle on the image
    cv.Rectangle(img,pt1,pt2,cv.CV _ RGB(255,0,0),3,8,0)
    cv.ShowImage( "result", img )


if _ _ name _ _ ==' _ _ main _ _ ':
  # Load the Haar cascade
  cascade _ name="./haarcascade _ frontalface
  alt _ tree.xml"
  cascade=cv.Load(cascade _ name)


  # Start capturing.Can change index if more than one
  camera present
  capture=cv.CaptureFromCAM(0)


  # Create the output window
  cv.NamedWindow("result",1)


  frame _ copy=None
  while True:
    frame=cv.QueryFrame( capture )
    # make a copy of the captured frame
    if not frame _ copy:
      frame _ copy=cv.CreateImage((frame.
      width,frame.height),
      cv.IPL _ DEPTH _ 8U, frame.nChannels )
      cv.Copy( frame, frame _ copy )

    detect _ and _ draw(frame _ copy)

    c=cv.WaitKey(7)
    if c==27: # Escape pressed
            break
```

Interns and employees alike gather and swap ideas in the atrium of Microsoft Research's building 99.

LABZ

Microsoft Research
Redmond, Washington

At 9:00 a.m., the Microsoft shuttle bus pulls up to an inconspicuous office building in a leafy suburb of Seattle. Known as building 99, this structure provides the homebase for Microsoft Research. Inside, under the glass roofed atrium, groups of researchers are hard at work discussing their projects over coffee, available free from Starbucks machines on every floor.

The atrium serves as a hub for collaboration and discussion, with residents gathering throughout the day to meet and socialize. The 300 people at Microsoft Research are a mix of permanent researchers, visiting collaborators, and interns. Although interns are only there for three months at a time, they get all the same privileges, and many of the same responsibilities, as regular employees.

Internships are one of the best ways to get a foot in the door with mainstream industry research. Internships give students a taste of the research community in a specific field, but also let them start networking with experienced and likeminded people.

THE INTERN EXPERIENCE

In the social computing lab on the ground floor, it's hard not to get drawn into discussions on the benefits of Twitter with interns who are working on ways to visualize and analyze social networking data. In June 2009, when Michael Jackson died, news of his death hit Twitter and caused fascinating social grouping patterns, which the Microsoft Research interns visualized in new ways using tools they had developed. Projects such as Bing Collaborative Search, Kodu, and Microsoft Portrait all have roots in this group.

The Windows Mobile building next door provides a huge selection of fresh food cooked on demand. During lunch, it's easy to get wrapped up chatting about projects, but it also often motivates lucrative collaboration between colleagues, which is encouraged by the company. Researchers typically spend half a work day per week on something aside from their primary projects.

All interns are assigned a mentor in a field related to their project to guide them and act as their first point of contact. Additionally, Microsoft Research hosts weekly talks from other Microsoft divisions about current projects, general topics, or even to get feedback on released products, opening up even more venues for networking and learning.

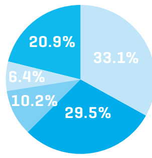
If prototypes need to be constructed, the fully equipped on-site hardware lab, kitted out with a laser cutter, a CNC milling machine, an optics bench, and more, is a perfect place to test out ideas in practice. It's common to find groups huddling over half-built prototypes late at night, frantically preparing for demos the next day.

Amazingly, many of the cool demos seen at events like TechFest are actually built by a couple of dedicated guys in building 99 using everyday materials. Although these projects rarely become full products, some do—such as Surface, Project Natal (for Xbox 360), and Azure. Many projects contribute toward future releases in core technologies, such as .NET and Windows, as well as collaborations with other institutions.

Walking around building 99, it's apparent that people enjoy working here, and not just for the free coffee.

1965

Douglas Engelbart of Stanford Research Laboratory (now SRI) develops the computer mouse



Apple	33.1%
Nokia	29.5%
HTC	10.2%
Samsung	6.4%
Others	20.9%

WORLDWIDE TOUCH SCREEN SMARTPHONE MARKET

564

Average number of mouse clicks per day by an average employee

Source: Remedy Interactive

Research is driven by the need to contribute to the scientific community, by both improving existing ideas and systems, and introducing new ones.

If interns have any energy left after a full day in the lab, they can join some of the local trips that Microsoft organizes to see the sights of Seattle and Washington state. Some excursions have included the nearby Olympic Mountains, downtown Seattle, the Boeing factory, and local arts festivals.

HOW TO GET INVOLVED

Microsoft Research accepts applications for interns on a rolling basis via an online submission form (http://research.microsoft.com/en-us/jobs/intern/apply.aspx), though it is restricted to masters-level and PhD students only. Internships, which are paid, usually last 12 weeks and are offered at any of Microsoft's eight labs worldwide, though the majority are in Redmond.

The review process can take up to three months. Selected candidates are then invited to interview. If accepted, the company provides some support in finding accommodation, making travel arrangements, and securing visas.

—Tom Bartindale

OTHER TOP INDUSTRY LABS

Intel Labs has three labs in three locations: Berkeley, California; Seattle, Washington; and Philadelphia, Pennsylvania. *www.intel-research.net*

IBM Research has eight labs spread all over the world, offering numerous opportunities for students outside North America. *www.research.ibm.com*

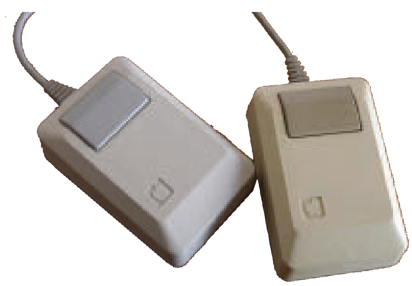
INRIA is a French national research institution, focused on three areas: computer science, control theory, and applied mathematics.*www.inria.fr*

PARC, the Palo Alto Research Center is business-focused and to some extent client-driven. *www.parc.com*

BACK

Apple's Mouse
25 Years Later

Few input devices revolutionized interaction as much as the mouse. Much of the world was first exposed to the mouse when Apple released the Macintosh 128k in 1984 and the boxy mouse that came with it. In 2009, Apple integrated multi-touch gestures with the Magic Mouse. Users now have a range of motions tied directly to their fingertips. Such drastic improvements raise just one question: Will the mouse ever disappear from our desks? *—James Stanier*



1984 Apple Macintosh Mouse	**2009** Apple Magic Mouse
Released January 24, 1984	Released October 20, 2009
Price Included with Macintosh 128k	Price $69, or incl. with desktop computer
Interactivity One physical button	Interactivity Gestural multi-touch interface with 360° scroll and two-finger swipe
Tracking Mechanical ball	Tracking Infrared laser diode
Slogan "If you can point, you can use a Macintosh."	Slogan "Suddenly, everything clicks. And swipes. And scrolls."
Criticism "The Macintosh uses an experimental pointing device called a 'mouse.' There is no evidence that people want to use these things." —John C. Dvorak	Criticism "We struggled through a difficult learning curve due to its uniformly narrow profile that sits too low for comfort."—CNET

EVENT DATES

CONFERENCES & JOURNALS

Computer Graphics International 2010
Singapore, Singapore
June 8-11,2010
Cost: SGD 960, SGD 580 (student)
http://cgi2010.miralab.unige.ch

IDC 2010- The 9th International Conference on Interaction Design and Children
Pompeu Fabra Universit
Barcelona, Spain
June 9-12
Cost: €410, €370, €130 (student)
www.iua.upf.edu/idc2010

EuroVis 2010 - Eurographics/IEEE Symposium on Visualization
Bordeaux 1 campus
Bordeaux, France
June 9-11
Cost: €240 (student)
http://eurovis.org

EuroITV 2010 - 8th European Conference on Interactive TV & Video
Tampere University of Applied Sciences and the hotel Holiday Club Tampere
Tampere, Finland
June 9-11
Cost: unavailable at press time
www.euroitv2010.org

ACM Hypertext 2010
Emmanuel College, University of Toronto
Toronto, Canada
June 13-16
Cost: $450, $350 (ACM/SIGWEB), $195 (student)
www.ht2010.org

ACM SIGCHI Symposium on Engineering Interactive Computing Systems
Ernst-Reuter-Haus
Berlin, Germany
June 21-23
Cost: €450 (student)
http://research.edm.uhasselt.be/˜ep-muti2010

UMAP 2010 - 18th International Conference on User Modeling, Adaptation and Personalization
Hilton Waikoloa Village
Waikoloa, Hawaii, U.S.
June 20-24
Cost: $250-$600
www.hawaii.edu/UMAP2010

Afrigraph 2010
Rickety Bridge Wine Estate
Franschhoek, South Africa
June 21-23
Cost: 5200 ZAR (international students)
www.afrigraph.org/conf2010

SG '10: 10th International Symposium on SMART GRAPHICS
Banff Centre
Banff, Canada
June 24-26, 2010
Cost: unavailable at press time
www.smartgraphics.org

Create10 - Innovative Interactions
Edinburgh Napier University
Edinburgh, U.K.
June 30-July 2
Cost: £290, £80 (Students)
www.create-conference.org

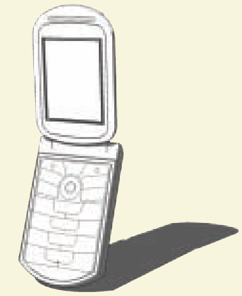
EuroHaptics 2010
Vrije Universiteit
Amsterdam, The Netherlands
July 8-10
Cost: unavailable at press time
www.eurohaptics2010.org

eNTERFACE 10 - 6th International Summer Workshop on Multimodal Interfaces
UvA Building at Science Park Amsterdam
Amsterdam, The Netherlands
July 12 - August 6
Cost: free
http://enterface10.science.uva.nl

SIGGRAPH 2010
Los Angeles Convention Center
Los Angeles, U.S.
July 25-29
Cost: $900, $850(ACM SIGGRAPH member), $350 (student member)
www.siggraph.org/s2010

DIS 2010: Designing Interactive Systems
Aarhus School of Architecture
Aarhus, Denmark
August 16-20
Cost: unavailable as of press time
www.dis2010.org

HCI 2010 - Play is a serious business
University of Abertay Dundee,



FEATURED EVENT

MobileHCI International Conference on Human-Computer Interaction with Mobile Devices and Services
Lisboa, Portugal
September 7-10,2010
http://mobilehci2010.di.fc.ul.pt/index.html
Technology is shrinking and becoming more and more mobile, and the opportunities to work with small, mobile devices are constantly growing. MobileHCI provides a unique forum for academics and practitioners to come together and discuss innovations, challenges, solutions, and ideas for the many aspects of human-computer interaction in mobile computing. The relaxed environment of this conference encourages open discussion among attendees and provides an opportunity to interact with a wide variety of people with common interests and many different perspectives.

MobileHCI 2010 will be held in Lisbon, Portugal, and promises an engaging program covering design, evaluation, and application for mobile and wearable computing. The conference will include full and short research papers; workshops to foster discussion based on selected themes; tutorials; interactive panels; practically-motivated industrial case studies; a design competition to encourage innovative problem solving; a future innovations track to encourage fresh concepts; a doctoral consortium to encourage student feedback and interaction; posters that provide the opportunity to informally discuss current work; and hands-on demonstrations.

Keynote speakers include Patrick Baudisch, Scott Jenson, and Josh Ulm.

—Jason Wiese

Dundee, Scotland
September 6-10
Cost: unavailable as of press time
http://hci2010.abertay.ac.uk

23rd ACM Symposium on User Interface Software and Technology
New York Hall of Science
New York, New York, U.S.
Deadline to submit posters, doctoral symposium: June
Conference date: October 3-6, 2010
Cost: unavailable as of press time
www.uist2010.org

CHIMIT 2010: Computer Human Interaction for the Management of Information Technology
San Jose, California, U.S.
Deadline to submit: July 3
Conference date: November 7-8
Cost: $75-$100 (student)
www.chimit10.org

TEI'11: Fifth International Conference on Tangible, Embedded, and Embodied Interaction
Funchal, Portugal
Deadline to submit: August 1
Conference date: January 22-26, 2011
Cost: unavailable as of press time
www.tei-conf.org/11

HCI International 2011
Hilton Orlando Bonnet Creek
Orlando, Florida, U.S.
Deadline to submit papers: October 15
Conference date: July 9-14, 2011
Cost: unavailable as of press time
www.hcii2011.org/index.php

SCHOLARSHIPS, FELLOWSHIPS & GRANTS

Korean American Scholarship Fund
Deadlines: May 31 (schools in the western, mid-western, mid-eastern, and southern regions); June 15, 2010 (eastern schools); June 26 (northeastern schools)
Eligibility: Full-time undergraduate and graduate students, as well as high school juniors, of Korean-American background, studying in the U.S. may apply. Scholarships are awarded based on financial need, academic achievement, school activities, and community services.
Benefits: Award amounts and distribution timelines vary.
www.kasf.org

AISES Google Scholarship
Deadline: June 15
Eligibility: American Indian, Alaska Native and Native Hawaiian AISES members pursuing degrees in computer science or computer engineering, enrolled full-time as undergraduate or graduate students (or in their second year of a two-year college with a plan to move to a four-year institution) with 3.5/4.0GPA.
Benefits: $10,000 scholarship
www.aises.org

Hertz Graduate Fellowship Award
Deadline: October 2010
Eligibility: Graduate students working toward a PhD degree in the applied physical, biological, and engineering sciences. Must study at one of the Foundations tenable schools (see www.hertzfoundation.org/dx/fellowships/schools.aspx) or be willing to petition for your school to be accepted.
Benefits: Merit-based scholarships vary but can consist of cost-of-education support and personal support, up to $31,000.
www.hertzfoundation.org

CONTESTS & EVENTS

ACM Student Research Competition at ASSETS 2010
Held in conjunction with ACM SIGACCESS ASSETS 2010
Orlando, FL
Deadline to submit: July 2
Event date: October 25-27
www.sigaccess.org/assets10/competition.html

International Olympiad in Informatics
University of Waterloo
Waterloo, Ontario, Canada
Deadline to submit: Competitions take place August 16 and 18 during week-long event
Event dates: August 14-20
www.ioi2010.org

SMART Multitouch Application Contest
Design a software application for the SMART Table
Deadline to submit: July 1
Winners Announced: September 1
www2.smarttech.com/

POINTERS

WHERE TO FIND HCI EVENTS

ACM Special Interest Group on Computer Human Interaction's calendar
www.sigchi.org/conferences/calendarofevents.html

ACM Conferences
www.acm.org/conferences

Interaction-Design.org's calendar
www.interaction-design.org/calendar

HCI GROUPS

Special Interest Group on Computer Human Interaction
www.sigchi.org

HCI RESOURCES

HCI Resources & Bibliography
www.hcibib.org

Scientometric Analysis of the CHI Proceedings
www.bartneck.de/publications/2009/scientometricAnalysisOfTheCHI

ACM *interactions* magazine
http://interactions.acm.org

GRADUATE PROGRAMS

Carnegie Mellon University, Human-Computer Interaction Institute
www.hcii.cmu.edu

Georgia Institute of Technology,
Human-Computer Interaction
www.cc.gatech.edu/hci

Massachusetts Institute of Technology,
MIT Media Laboratory
www.media.mit.edu

Stanford University, HCI Group
http://hci.stanford.edu

University of California—Irvine,
Human Computer Interaction
www.ics.uci.edu/faculty/area/area_hci.php

University of Maryland,
Human-Computer Interaction Lab
www.cs.umd.edu/hcil

University of Michigan, Human-Computer Interaction Specialization
www.si.umich.edu/msi/hci.htm

BEMUSEMENT

Can Oleg Beat Erdos?

A puzzle to get started, from our friends at CMU

Oleg and (the ghost of) Erdös play the following game. Oleg chooses a non-negative integer a_1 with at most 1,000 digits. In Round i the following happens:

Oleg tells the number a_i to Erdös, who then chooses a non-negative integer b_i, and then Oleg defines $a_{i+1} = |a_i - b_i|$ or $a_{i+1} = a_i + b_i$. Erdös wins if a_{20} is a power of 10, otherwise Oleg wins.

Who is the winner, Oleg or Erdös?

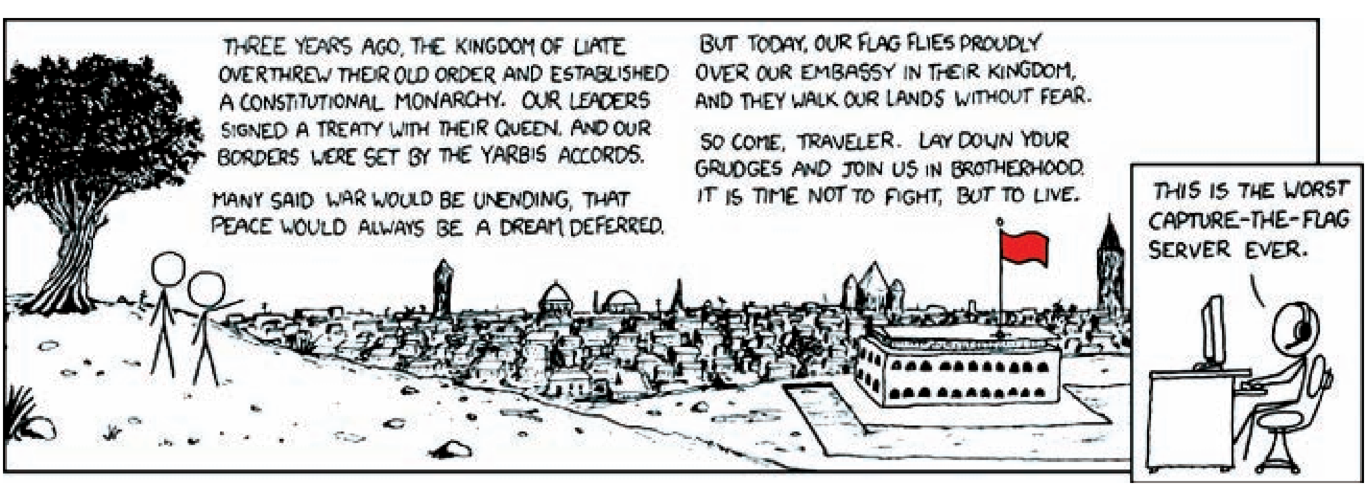
—Tom Bohman, Oleg Pikhurko, Alan Frieze, and Danny Sleator at The Puzzle Toad.

Find the solution at: http://www.cs.cmu.edu/puzzle/Solution22.pdf

Debugging

Borders

xkcd.com



SUBMIT A PUZZLE

Can you do better? Bemusements would like your puzzles and mathematical games (but not Sudoku). Contact **xrds@acm.org** to submit yours!

acm STUDENT MEMBERSHIP APPLICATION

Join ACM online: www.acm.org/joinacm

CODE: CRSRDS

Name *Please print clearly*

Address

City | State/Province | Postal code/Zip

Country | E-mail address

Area code & Daytime phone | Mobile Phone | Member number, *if applicable*

MEMBERSHIP BENEFITS AND OPTIONS

- Print & electronic subscription to *XRDS: The ACM Magazine for Students*
- Electronic subscription to *Communications of the ACM* magazine
- Free software and courseware through the ACM Student Academic Initiative
- 3,200 online courses in multiple languages, 1,000 virtual labs and 500 online books
- ACM's *CareerNews* (twice monthly)
- ACM e-news digest *TechNews* (tri-weekly)
- Free e-mentoring service from MentorNet
- ACM online newsletter *MemberNet* (twice monthly)
- *Student Quick Takes*, ACM student e-newsletter (quarterly)
- ACM's Online Guide to Computing Literature
- Free "acm.org" email forwarding address plus filtering through Postini
- Option to subscribe to the full ACM Digital Library
- Discounts on ACM publications and conferences, valuable products and services, and more

PLEASE CHOOSE ONE:

❑ **Student Membership: $19 (USD)**
❑ **Student Membership PLUS Digital Library: $42 (USD)**
❑ **Student Membership PLUS Print *CACM* Magazine: $42 (USD)**
❑ **Student Membership w/Digital Library PLUS Print *CACM* Magazine: $62 (USD)**

PUBLICATIONS

Please check one

Check the appropriate box and calculate amount due.	Issues per year	Code	Member Rate	Member Rate PLUS Air*
• ACM Inroads	4	247	$13 ❑	$65 ❑
• *acmqueue* website www.acmqueue.com	N/A	N/A	N/A	N/A
• Computers in Entertainment (online only)	4	247	$40 ❑	N/A
Computing Reviews	12	104	$52 ❑	$85 ❑
• Computing Surveys	4	103	$33 ❑	$59 ❑
• XRDS: Crossroads, The ACM Magazine for Students (included with student membership)	4	XRoads	N/A	N/A
• interactions (included in SIGCHI membership)	6	123	$19 ❑	$48 ❑
Int'l Journal of Network Management (Wiley)	6	136	$90 ❑	$116 ❑
• Int'l Journal on Very Large Databases	4	148	$81 ❑	$107 ❑
• Journal of Experimental Algorithmics (online only)	12	129	$29 ❑	N/A
• Journal of Personal and Ubiquitous Computing	6	144	$63 ❑	$89 ❑
• Journal of the ACM	6	102	$52 ❑	$104 ❑
• Journal on Computing and Cultural Heritage	4	173	$46 ❑	$63 ❑
• Journal on Data and Information Quality	4	171	$46 ❑	$65 ❑
• Journal on Emerging Technologies in Computing Systems	4	154	$39 ❑	$56 ❑
• Linux Journal (SSC)	12	137	$29 ❑	$58 ❑
• Mobile Networks & Applications	4	130	$70 ❑	$95 ❑
• Wireless Networks	4	125	$70 ❑	$95 ❑
Transactions on:				
• Accessible Computing	4	174	$46 ❑	$64 ❑
• Algorithms	4	151	$49 ❑	$66 ❑
• Applied Perception	4	145	$40 ❑	$57 ❑
• Architecture & Code Optimization	4	146	$40 ❑	$57 ❑
• Asian Language Information Processing	4	138	$36 ❑	$53 ❑
• Autonomous and Adaptive Systems	4	158	$38 ❑	$55 ❑
• Computational Biology and Bio Informatics	4	149	$18 ❑	$63 ❑
• Computer-Human Interaction	4	119	$40 ❑	$59 ❑
• Computational Logic	4	135	$41 ❑	$60 ❑
• Computation Theory		176	$46 ❑	$72 ❑
• Computer Systems	4	114	$44 ❑	$63 ❑
• Computing Education (online only)		277	$38 ❑	N/A
• Database Systems	4	109	$43 ❑	$62 ❑
• Design Automation of Electronic Systems	4	128	$40 ❑	$59 ❑
• Embedded Computing Systems	4	142	$41 ❑	$58 ❑
• Graphics	4	112	$48 ❑	$67 ❑
• Information & System Security	4	134	$41 ❑	$58 ❑
• Information Systems	4	113	$44 ❑	$63 ❑
• Internet Technology	4	140	$39 ❑	$56 ❑
• Knowledge Discovery From Data	4	170	$47 ❑	$64 ❑
• Mathematical Software	4	108	$44 ❑	$63 ❑
• Modeling and Computer Simulation	4	116	$48 ❑	$67 ❑
• Multimedia Computing, Communications, and Applications	4	156	$39 ❑	$56 ❑
• Networking	6	118	$28 ❑	$76 ❑
• Programming Languages & Systems	6	110	$56 ❑	$82 ❑
• Reconfigurable Technology & Systems	4	172	$46 ❑	$64 ❑
• Sensor Networks	4	155	$39 ❑	$56 ❑
• Software Engineering and Methodology	4	115	$40 ❑	$59 ❑
• Speech and Language Processing (online only)	4	153	$39 ❑	N/A
• Storage	4	157	$39 ❑	$56 ❑
• Web	4	159	$38 ❑	$55 ❑

marked • are available in the ACM Digital Library
**Check here to have publications delivered via Expedited Air Service. For residents outside North America only.*

PUBLICATION SUBTOTAL: ______

INSTRUCTIONS

Carefully complete this application and return with payment by mail or fax to ACM. You must be a **full-time** student to qualify for student rates.

CONTACT ACM

phone: 800-342-6626 (US & Canada)
+1-212-626-0500 (Global)
hours: 8:30am–4:30pm US Eastern Time
fax: +1-212-944-1318
email: acmhelp@acm.org
mail: Association for Computing Machinery, Inc.
General Post Office
P.O. Box 30777
New York, NY 10087-0777

For immediate processing, FAX this application to +1-212-944-1318.

PAYMENT INFORMATION

Payment must accompany application

Member dues ($19, $42, or $62) $ ______

To have *Communications of the ACM* sent to you via Expedited Air Service, add $50 here (for residents outside of North America only). $ ______

Publications $ ______

Total amount due $ ______

Check or money order (make payable to ACM, Inc. in U.S. dollars or equivalent in foreign currency)

❑ **Visa/Mastercard** ❑ **American Express**

Card number | Exp. date

Signature

Member dues, subscriptions, and optional contributions are tax deductible under certain circumstances. Please consult with your tax advisor.

EDUCATION

Name of School

Please check one: ❑ **High School** (Pre-college, Secondary School) **College:** ❑ Freshman/1st yr. ❑ Sophomore/2nd yr. ❑ Junior/3rd yr. ❑ Senior/4th yr. **Graduate Student:** ❑ Masters Program ❑ Doctorate Program ❑ Postdoctoral Program ❑ **Non-Traditional Student**

Major | Expected mo./yr. of grad.

Age Range: ❑ 17 & under ❑ 18-21 ❑ 22-25 ❑ 26-30 ❑ 31-35 ❑ 36-40 ❑ 41-45 ❑ 46-50 ❑ 51-55 ❑ 56-59 ❑ 60+

Do you belong to an ACM Student Chapter? ❑ Yes ❑ No

I attest that the information given is correct and that I will abide by the ACM Code of Ethics. I understand that my membership is non transferable.

Signature