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-

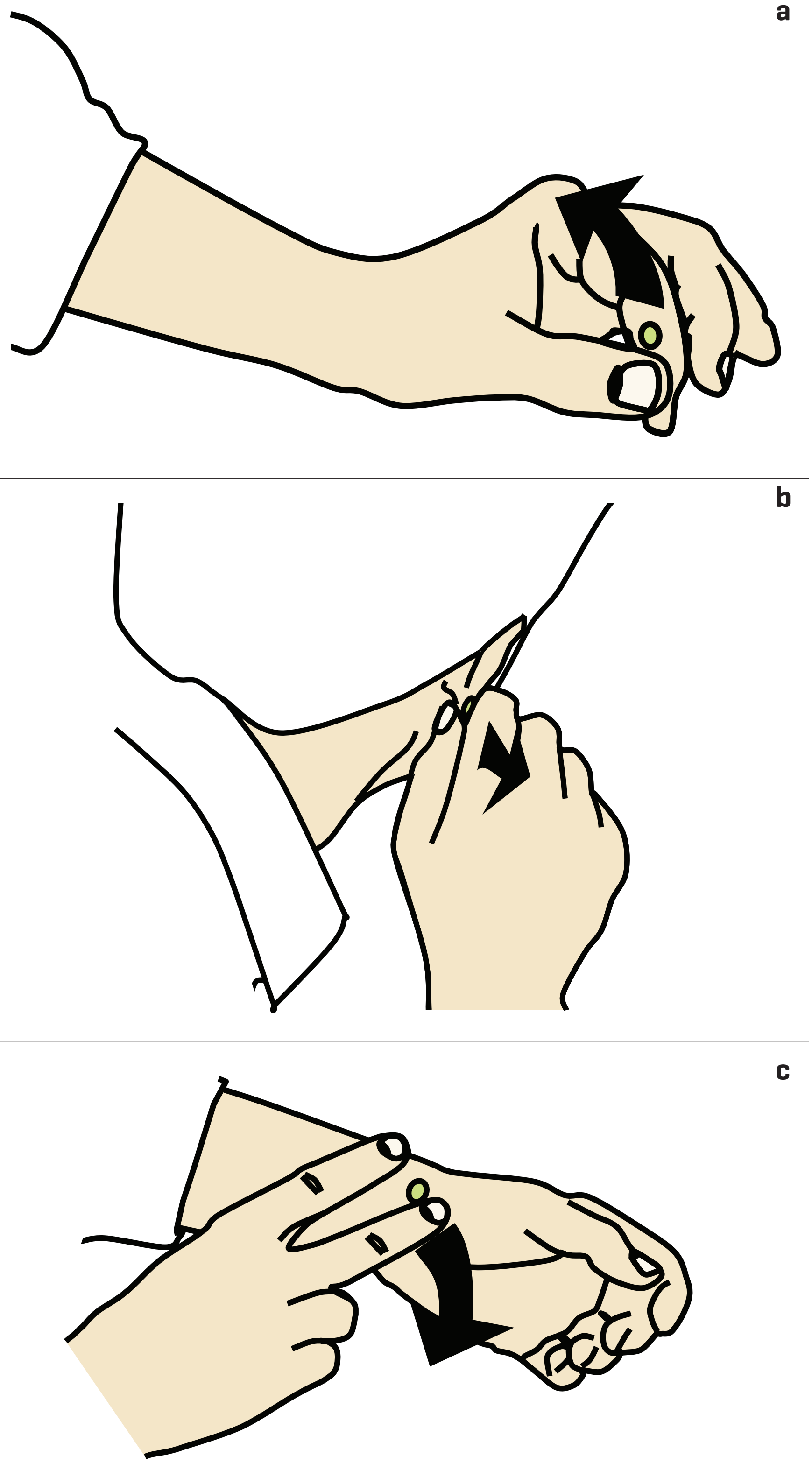


Figure 1: Gesture input allows for input on the tiniest of mobile devices (a) on the finger, providing tactile feedback, (b) in the earlobe, providing auditory feedback, or (c) on the wrist, providing visual feedback. The user is entering a "2" by "scanning" two fingers (see Ni and Baudish's Disappearing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

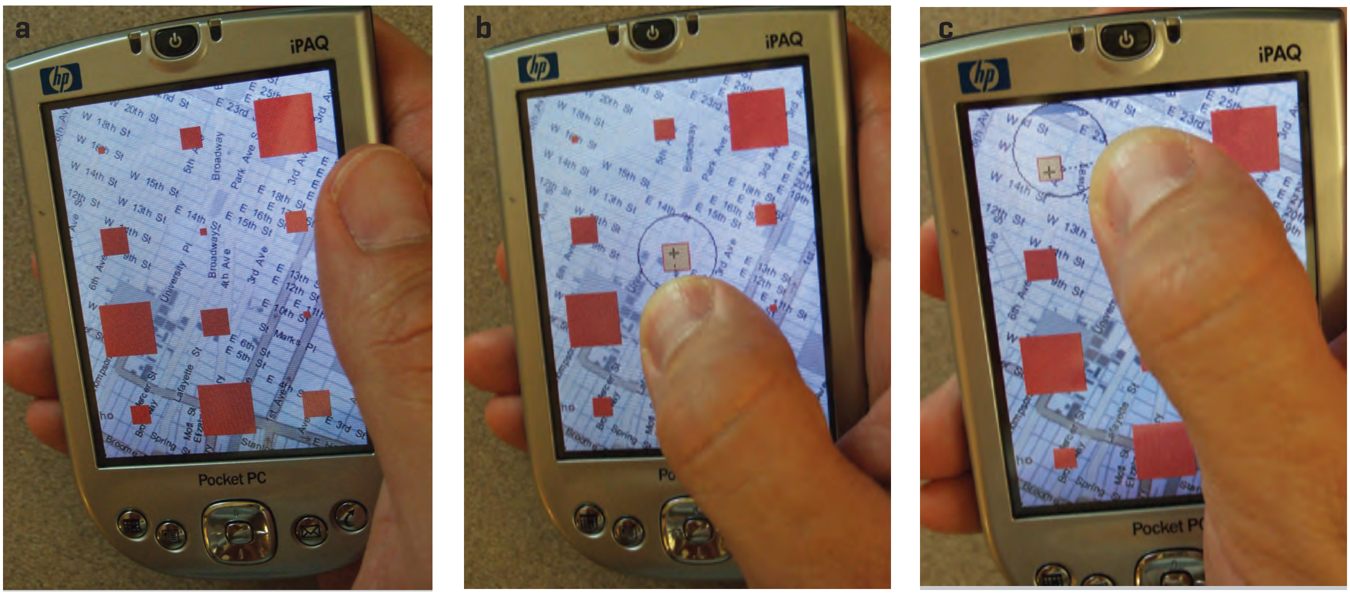
POINT, TOUCH, AND FAT FINGERS

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."

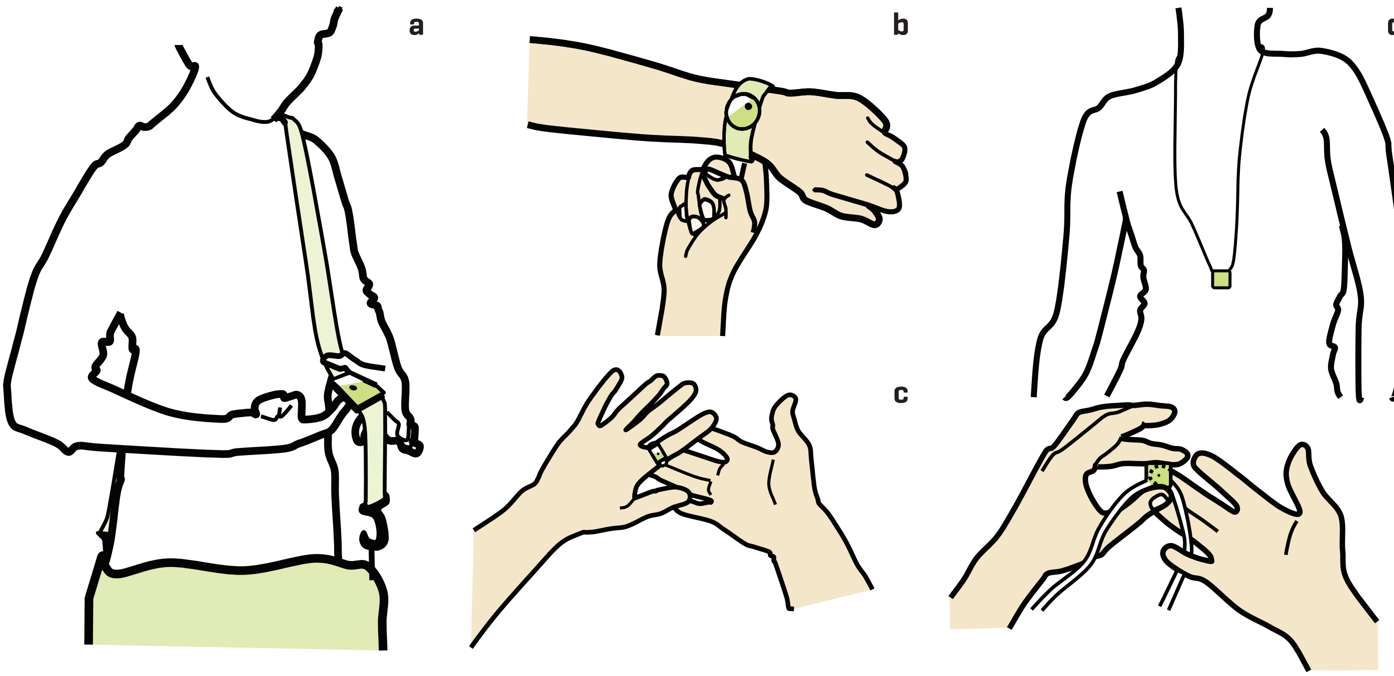
Interaction with Nanotouch also turned out to be highly precise and in a user study, participants were able to acquire a 2.8mm target with 98 percent accuracy. More importantly, back-of-device interaction works practically independent of the device size. In our user study, participants operated Nanotouch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

ACCURATE TOUCH FOR MOBILE DEVICES

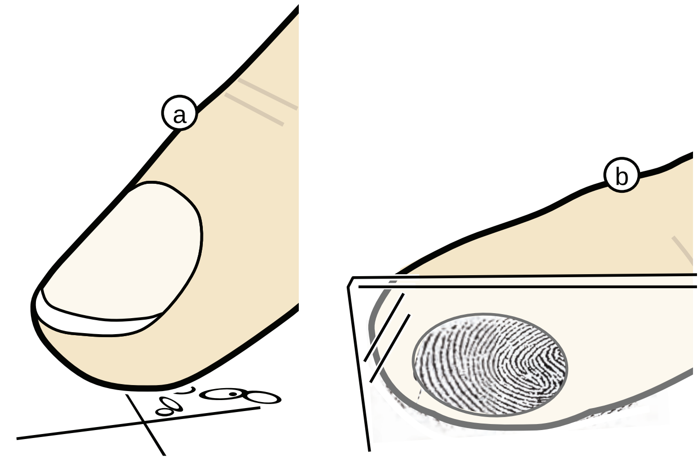
Figure 6 shows a setup that implements two of these prototype devices. The cameras surrounding the finger belong to an Optitrack optical tracking system that determine finger angles by observing tiny markers glued to the user's fingernail. The resulting setup allows users to acquire targets of 4.3mm 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**],

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

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.

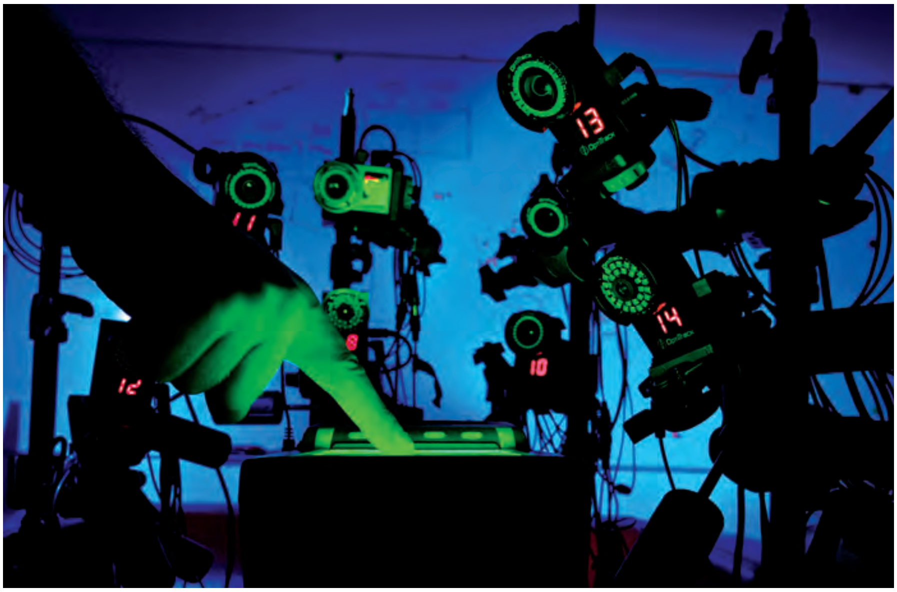


MOBILE PHONES AS PCS

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.