

The Multiple-Touch User Interface Revolution

Philip I.S. Lei and Angus K.Y. Wong, *Macau Polytechnic Institute*

Because of their intuitive interaction styles, multiple-touch devices could trigger a revolution in user interfaces. But how can designers advance this technology?

The window, icon, menu, and pointer (WIMP) interface built for single-touch pointing devices has long been established as the standard GUI in desktop operating systems and is also prevalent in mobile devices, such as personal digital assistants and smartphones.

However, this approach offers limited actions for GUI designers, a fact that becomes more obvious when using an application with a sophisticated interface, such as Adobe Photoshop. To accommodate complex operations with single-touch pointing devices, we must use widgets (such as a scroll bar or popup menu), modifier keys (such as control + click or double-click), or mouse actions (such as drag and drop). Although they're usable, such traditional interfaces are

less intuitive and impose a heavier cognitive load on the user.

In contrast, multiple-touch technology has attracted much attention in recent years for its ease of use. A multiple-touch device can track more than one touch simultaneously, providing more sophisticated and intuitive interaction in user interface design. For example, to zoom into a photo using a multiple-touch interface, a user can first touch the photo with two fingers and then spread the fingers apart, as Figure 1a shows. The interface reacts as if the two fingers were stretching a printout on a rubber sheet. This pinching gesture is more intuitive and direct than using a slider widget for resizing in the traditional interface, as Figure 1b shows.

Advances in multiple-touch hardware and user interface design have resulted in recent high-profile

products such as Apple's iPhone (www.apple.com/iphone) and Microsoft Surface (www.microsoft.com/surface). At the 2008 All Things Digital Conference, Microsoft demonstrated how multi-touch technology will enhance the next version of Windows (Windows 7; http://d6.allthingsd.com/20080527/gates_ballmer/). Consumers are increasingly intrigued with this new style of interaction and its ease of use. Therefore, we believe that this promising technology could trigger a revolution in GUI design.

Multiple-Touch Detection Mechanisms

Most people aren't aware that multitouch technology has been around for quite some time; the sidebar "Overview of Multiple-Touch Products" provides a brief history.

Part of the fascination with this type of tool is understanding how it works. Multitouch products use detection mechanisms, which we can classify into three categories: sensor array, capacitive sensing, and vision based. Figure 2 shows a schematic comparison of these mechanisms.

Sensor Array Detection

A sensor array touch surface consists of a grid of touch sensors (see Figure 2a). When a user exerts a touch on

the surface, one or more touch sensors activate and the surface uses the physical positions of these sensors to determine the touch's position. Because the touch sensors work independently, the surface can track multiple touches simultaneously.

Tactex Controls (www.tactex.com) has produced a series of touchpads featuring sensor array detection of pressure-sensitive touch sensors. Timothy Chen and his colleagues¹ used one of these, called MTC Express, to build a simulation of a liquid flow field in which fingers assert the obstacles. The touchpad contained 12×6 sensors on a 19×17 -cm opaque surface, with each sensor measuring 8-bit pressure. The



Figure 1. Pinching gesture versus slider widget. (a) Photo zooming using a pinching gesture (in Apple's iPhone). (b) Photo zooming using a slider widget.

MULTIPLE-TOUCH TECHNOLOGY

Overview of Multiple-Touch Products

Although the general public isn't aware of it, multiple-touch technologies have a long history. Capacitive multiple-touch tablets¹ appeared as early as 1985, and the pinching gesture for scaling was first demonstrated in 1991.² A chronology of various multitouch devices and related work is available at www.billbuxton.com/multitouchOverview.html.

Multiple-touch technology also enables a wide spectrum of user interfaces, including mouse emulation, gesture, dual-hand input, and multiuser interaction. Generally speaking, we classify multiple-touch products into two categories: a separated input device, and a touch surface with an integrated display.

Separated Input Devices

These multiple-touch products work as the input devices that connect to ordinary computers and provide enhanced input capability to desktop applications. As an example, JazzMutant's Lemur (www.jazzmutant.com), released in 2003, is considered the first commercial product to use multitouch technology (see Figure A1). It's a configurable music controller with a distinctive 12-inch multitouch-enabled screen. Users interact with the virtual controller on the touchscreen as if it were a real counterpart—a user, for example, can move several virtual knobs simultaneously in the equalizer interface. This prod-

uct can serve as a controller for MIDI applications running on a connected computer.

FingerWorks (which Apple recently acquired; www.fingerworks.com) produced several touchpads that can recognize a rich vocabulary of multitouch hand gestures. One touchpad is called iGesture Numpad (see Figure A2), on which users can perform gestures to issue commands to the supported desktop applications. For example, to zoom within a Web browser, a user can spread all five fingers on the touchpad. The touchpad can also emulate the mouse and keyboard.

Tactiva's TactaPad (www.tactiva.com) incorporates both a camera and a pressure-sensitive touchpad with tactile response to give an immersive experience of multifinger control. As Figure A3 shows, the camera captures images of the hands on the touchpad, and the application software overlays a translucent shadow of them on the GUI. This shadow functions as a multifinger cursor. To emulate a mouse click, the user presses any finger on the touchpad, which in turn reduces hand motion because the user can intuitively use the nearest finger to tap a certain point on the screen. A demo video on the company's Web site shows how a multitouch drawing program takes advantage of concurrent inputs from both hands.

Touch Surfaces with Integrated Displays

In contrast to the aforementioned input devices, some products combine a multitouch-enabled surface with a display and become an integrated input/output device.

The Mitsubishi Electric Research Laboratories released a multitouch table called DiamondTouch³ in 2001. This product supports multiuser interaction and can identify which user makes a certain touch. The application screen is front-projected on the opaque touch surface (see Figure B1). Several studies have used DiamondTouch⁴⁻⁸ to explore interaction styles in the multiple-touch interface. For example, Mike Wu and Ravin Balakrishnan⁴ presented several multifinger and whole-hand gestures for room planning in a multiuser environment, and Butler and Amant⁸ studied the bimanual operation of drawing tools in graphics-editing software.

In 2005, Jeff Han demonstrated⁹ how to use frustrated total internal reflection (FTIR) to construct a high-resolution, multiple-touch surface. Since

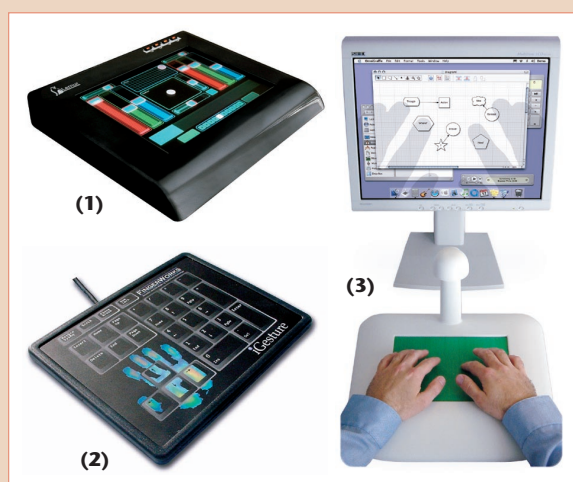


Figure A. Examples of separated multitouch input devices. (1) JazzMutant's Lemur, (2) FingerWorks's iGesture Numpad, and (3) Tactiva's TactaPad.

then, Perceptive Pixel (www.perceptivepixel.com) has commercialized the technology to manufacture large interactive walls. Users manipulate objects projected on an interactive wall by gestures—for example, they can resize a map on the interactive wall by pinching. They can also pan the map by dragging a finger on it (see Figure B2).

Apple's iPhone is considered the first portable consumer product with a multitouch interface. This mobile phone features a multitouch-enabled screen as the primary user interface and supports intuitive gestures for various commands—for example, a user can wipe his or her finger on the contact list to scroll it and tap a telephone number in a contact to make a call. The user can also drag a finger on the screen to pan and pinch two fingers apart to zoom in on a map application.

In June 2007, Microsoft announced a multitouch tabletop called Microsoft Surface, which can detect both finger touches and physical objects resting on the tabletop. It allows multiple users to manipulate photos on the tabletop simultaneously by using the dragging and pinching gestures (see Figure B3). In addition to fingertips, the product can also recognize physical objects placed on the surface, which lets users work with a painting brush, for example, to draw in a drawing application.

References

1. S. Lee, W. Buxton, and K.C. Smith, "A Multiple-Touch Three-Dimensional Touch-Sensitive Tablet," *Proc. Conf. Human Factors in Computer Systems (CHI 85)*, ACM Press, 1985, pp. 21–26.
2. P. Wellner, "The DigitalDesk Calculator: Tactile Manipulation on a Desktop Display," *Proc. Symp. User Interface Software and Technology (UIST 91)*, ACM Press, 1991, pp. 27–33.
3. P. Dietz and D. Leigh, "DiamondTouch: A Multi-User Touch Technology," *Proc. ACM Symp. User Interface Software and Technology (UIST 01)*, ACM Press, 2001, pp. 219–226.
4. M. Wu and R. Balakrishnan, "Multi-Finger and Whole Hand Gestural Interaction Techniques for Multi-User Tabletop Displays," *Proc. ACM Symp. User Interface Software and Technology (UIST 03)*, ACM Press, 2003, pp. 192–202.
5. H. Benko and S. Feiner, "Balloon Selection: A Multi-Finger Technique for Accurate Low-Fatigue 3D Selections," *Proc. IEEE Symp. 3D User Interface*, IEEE CS Press, 2007, pp. 79–86.

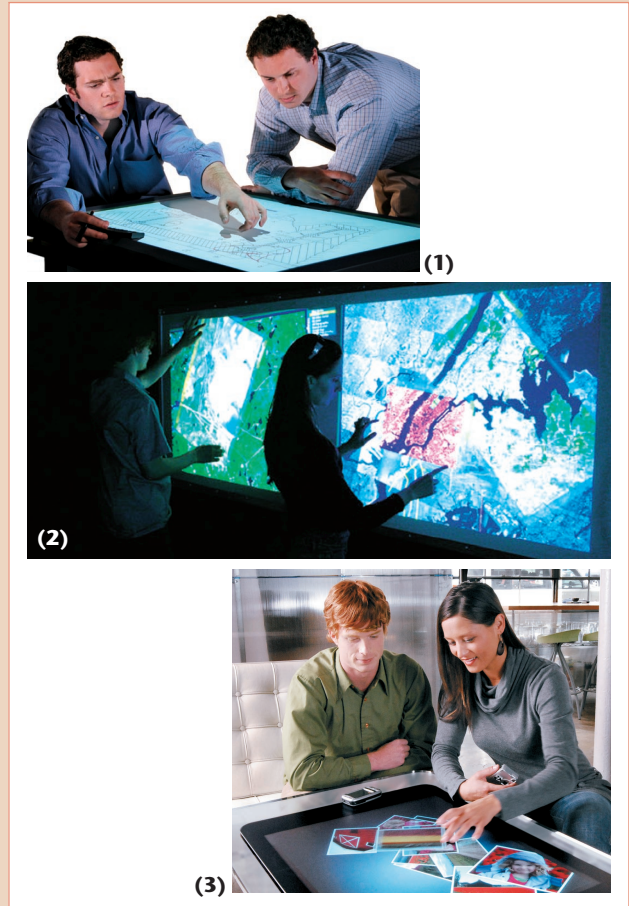


Figure B. Examples of touch surfaces with integrated displays. (1) The DiamondTouch table, (2) Perceptive Pixel's interactive wall, and (3) Microsoft Surface.

6. T. Moscovich et al., "A Multi-Finger Interface for Performance Animation of Deformable Drawings," *Proc. ACM Symp. User Interface Software and Technology (UIST 05)*, ACM Press, 2005; www.acm.org/uist/archive/adjunct/2005/pdf/demos/d8-moscovich.pdf.
7. H. Benko, E.W. Ishak, and S. Feiner, "Cross-Dimensional Gestural Interaction Techniques for Hybrid Immersive Environments," *Proc. IEEE Virtual Reality 2005*, IEEE CS Press, 2005, pp. 209–216.
8. C.G. Butler and R.S. Amant, "HabilisDraw DT: A Bimanual Tool-Based Direct Manipulation Drawing Environment," *Proc. Conf. Human Factors in Computing Systems (CHI 04)*, ACM Press, 2004, pp. 1301–1304.
9. J.Y. Han, "Low-Cost Multi-Touch Sensing through Frustrated Total Internal Reflection," *Proc. ACM Symp. User Interface Software and Technology (UIST 05)*, ACM Press, 2005, pp. 115–118.

MULTIPLE-TOUCH TECHNOLOGY

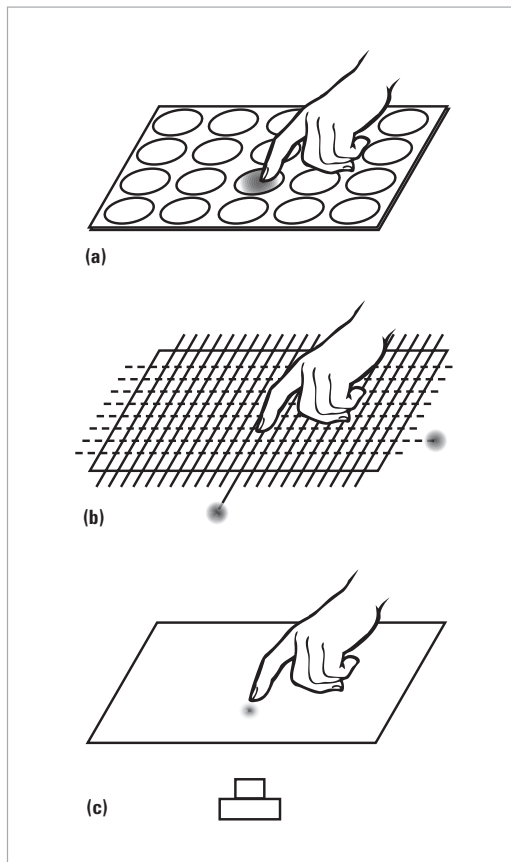


Figure 2. Multiple-touch detection mechanisms. (a) In the sensor array, each sensor in a grid can detect the press of a touch. (b) With capacitive sensing, when a finger is near the cross point of a pair of horizontal and vertical electrodes, the receivers connected to the electrodes will detect a signal change. (c) During vision-based detection, when a finger touches the touch surface, light is emitted or reflected from the fingertip. A digital camera then detects this light.

device used interpolation to achieve a higher effective input resolution of roughly 20 mm.

Capacitive-Sensing Detection

A capacitive touch surface uses the capacitive coupling between two conductors to sense a touch. Typically, the touch surface contains a mesh of horizontal and vertical antennas (see Figure 2b), which function as either a transmit-

ter or a receiver of electric signals. Because capacitive sensing requires the touching finger to be a conductor, the user must touch the surface with an ungloved hand.

Both Sony CSL's SmartSkin² and the Diamond-Touch table from Circle Twelve³ implement a multi-touch tabletop using a capacitive-sensing touch surface. The surface is opaque and the application screen is front-projected on it, making it look much like a touchscreen.

The SmartSkin touch surface is embedded with a grid of vertical transmitter electrodes and horizontal receiver electrodes. The vertical electrodes take turns transmitting a reference wave signal, which injects a signal in each horizontal electrode via capacitive coupling at the cross points of the grid. A receiver monitors the signals for each horizontal electrode. When a finger, which is conductive and grounded, is near a cross point (within 5 to 10 cm), the signals received by the horizontal electrodes will weaken because of the capacitive coupling between the finger and the electrodes. This effect helps the surface locate the fingertip's touch and estimate the hand's proximity. The device can increase position accuracy to 1 cm for a grid size of 10 cm through bicubic interpolation.

The DiamondTouch surface consists of a mesh of horizontal and vertical transmitter antennas. The touch surface looks like a woven fabric in which each horizontal (vertical) antenna is revealed on the top intermittently and a sharp contact touches exactly one antenna. The user is capacitively coupled to the signal receiver through its chair. When the user touches the sensing surface with a finger, the fingertip will cover a few antennas, including both vertical and horizontal ones. The receiver is capacitively coupled with the antennas under the fingertip and detects a mix of the antennas' identifying signals. The transmitter antennas' unique signals are designed to be orthogonal, so that the receiver can easily separate the mix of signals received. Through interpolation, the accuracy can be as small as 0.3 mm (www.circletwelve.com/circle12/images/file/DTflier-web.pdf).

Designers can make capacitive touch surfaces transparent using an indium-tin oxide (ITO) electrode mesh. By overlaying such a transparent touch surface on a flat panel display, we can manufacture portable devices with multitouch-

screens such as Apple's iPhone. The US patent application filed for the iPhone suggests a possible detection mechanism used by the device, but no official information is revealed.

Vision-Based Detection

Several research prototypes and products use digital cameras and computer vision techniques to track multiple touches. In these models, one or more digital cameras monitor the touch surface continuously—a user's touch on the surface causes the change in the image stream captured by the cameras, and the device then applies computer vision techniques to analyze the image stream to detect and track the touch. Vision-based touch surfaces excel in higher accuracy and support bigger form factors, but they're susceptible to harsh lighting conditions.

Existing research prototypes use numerous kinds of captured images to determine a touch. PlayAnywhere⁴ captures shadows cast by hands touching a front-projected touch surface and detects a fingertip's touch by the change of the shadow shape. TouchLight⁵ utilizes a binocular vision technique to distinguish touches on the surface by comparing images from two infrared cameras behind the surface.

Tactiva's TactaPad (www.tactiva.com) uses a camera above a touch surface to take pictures of both hands on the surface. A translucent image of the hands is overlaid on the screen and acts as a multifinger cursor for locating objects and widgets on the screen. The pressure-sensitive touch surface detects the start and end of a touch.

Microsoft Surface's multiple-touch tabletop uses several infrared cameras under the touch surface to capture images of physical objects resting on the surface. The device illuminates these objects, including fingertips, using an infrared light source beside the cameras. In addition, it back-projects the application screen on the touch surface. The input resolution is 1,280 × 960 pixels, and the output resolution is 1,024 × 768 pixels on a 30-inch display. A distinct feature of Microsoft Surface is the versatility of objects that it can detect—for example, a user can touch the surface using a paintbrush instead of a finger because infrared imaging doesn't rely on electrical properties or a refractive index of objects.

Jeff Han⁶ presented the use of frustrated total internal reflection (FTIR) to build a vision-based touch surface. Unlike the approaches previously discussed, FTIR doesn't directly capture the images of hands and then indirectly recognize the effects of touches. The device emits strong infrared light sideways, and the total internal reflection traps the light within the acrylic touch surface. When an object (such as a finger) touches the surface, it can frustrate the reflection and cause infrared to emit at the contact point. A camera underneath captures these touches continuously, and a connected computer uses motion detection techniques to track them. The application screen back-projects via a projector behind the touch surface. The research prototype Han used⁶ supports

Because vision-based detection mechanisms support bigger form factors, they're suitable for large displays such as tabletops and interactive walls.

an input resolution of 640 × 480 on a 406- × 305-mm surface, which implies an accuracy of 0.6 mm. However, the occurrence of FTIR depends on the touching material's refractive index, and as with capacitive sensing, an FTIR sensing surface might not detect a hand in a glove.

Comparison

The products we've mentioned have different input precision and support a wide spectrum of form factors. Table 1 offers a detailed comparison. As you can see, because sensor array and capacitive-sensing detection mechanisms support compact and portable form factors, and because they can be used in various lighting conditions, they're suitable for input devices for desktop or mobile devices. However, because vision-based detection mechanisms support bigger form factors, they're suitable for large displays such as tabletops and interactive walls.

Design Considerations

Now that we have a better understanding of how these devices work, what can we do to improve

MULTIPLE-TOUCH TECHNOLOGY

Table 1. Recent multiple-touch products and their characteristics.

Mechanism	Multiple-touch device	Features/resolution	Screen projection	Form factor
Sensor array	Tactex's EMC Express	Low resolution: ~20 mm	None	Input device for desktop
Capacitive sensing	FingerWorks touchpads	Focuses on gesture recognition	None	Input device for desktop
	SmartSkin	Proximity sensitive Medium resolution: 10 mm	Front projection	Tabletop
	DiamondTouch	Distinguishes touch from a person Medium resolution: 3 mm	Front projection	Tabletop
	Apple's iPhone	Small, transparent touch surface	Touch surface overlaid on LCD screen	Mobile phone
Vision based	Tactiva's TactaPad	Multifinger cursor Tactile feedback	None	Input device for desktop
	PlayAnywhere	Shadow-based touch detection	Front projection	Tabletop
	TouchLight	Binocular vision Camera can see through the touch surface	Rear projection	Interactive wall
	Microsoft Surface	Captures image of objects Accepts many kinds of paintbrushes	Rear projection	Tabletop
	Frustrated total internal reflection (FTIR)	High resolution: 0.6 mm	Rear projection	Tabletop and large interactive wall

and advance current designs? Here, we discuss future considerations for designing multitouch products.

Low Precision

In most multiple-touch products, users touch the surface with their fingers instead of using a tool like a stylus. This is necessary because human dexterity in coordinating finger movement is essential to the multiple-touch interface's usability. Nonetheless, a fingertip's relatively large contact area limits the precision.

However, the nature of the touch detection hardware also restricts input precision. Sensor array touch surfaces inherently have lower resolution because of each touch sensor's size. Vision-based touch surfaces might fail to recognize an individual fingertip from the image or shadow of the hands when the fingers are close together.

To mitigate this problem, current products generally avoid small screen objects that are close together. Recent work⁷ attempts to improve the effective precision of finger pointing by changing the interface's zoom level via the other hand. A user could also specify a parameter precisely via a widget instead of using a gesture's continuous action.

Hand-Eye Coordination

With multitouch devices, users can competently perform gestures to issue commands and specify parameters without looking at their hands. Hence, users can perform these gestures on indirect-control touchpads or an unattended region of direct-control devices. However, it isn't a simple task to point fingers to two desired positions on the screen at the same time. It takes even more hand-eye coordination to locate two screen points using an indirect-control touchpad. Therefore, existing systems usually have users select objects one by one.

Low Affordance of Gestures

Existing products and prototypes generally exhibit only a few onscreen hints of acceptable gestures. To cope with this low affordance, we must carefully design gestures that are simple, intuitive, and easy to learn—yet not easily confused. To facilitate this, we could limit the number of available gestures in any instance or possibly implement context-sensitive gesture recognition with visible hints. For example, Katherine Everitt and her colleagues⁸ reuse a small set of gestures for different commands performed in a separate region of the tabletop.

Hand Fatigue

The problem of hand fatigue becomes more serious in multitouch interfaces because the multiple interactions generally involve more hand motion than you would find in a traditional interface. Ergonomic issues could favor form factors that let users rest their palms on the touch surface.

Screen Occlusion

The user's hands obscure a larger amount of the screen when performing multifinger gestures. Existing systems usually display large objects on a large screen, so for systems with a smaller screen, the user interface should show only a few objects on the screen and favor gestures where the palm can be slanted on one side (for pinching instead of spreading five fingers). Although front projection alleviates the occlusion problem—because the application screen is projected on the user's hand—this might not be feasible in some environments.

Hardware Operating Requirements

Capacitive-sensing touch surfaces require the hand to be conductive and grounded for the sensors to work. Therefore, gloved hands won't be detected properly in this environment.

However, vision-based touch surfaces are generally sensitive to ambient lighting conditions, so most researchers test prototypes using FTIR in a dark room.⁶ Nonetheless, Microsoft Surface seems to work normally in indoor lighting.

As with any new technology, there are still many exciting opportunities for multitouch devices in how they can develop and evolve. We believe that multitouch interfaces will yield entirely new applications in many domains (including IT shops) by taking advantage of the novel and intuitive interaction styles the interface provides. Hopefully we've provided a basic overview of this technology, along with food for thought on how to improve future designs—particularly for desktop or laptop models using vertical screens. Regardless, we anticipate witnessing new computing modalities (in particular, modalities that don't use a vertical screen) as they emerge and become prevalent in the near future. 

References

1. T.T.H. Chen, S. Pels, and S.S. Min, "Flowfield and Beyond: Applying Pressure-Sensitive Multi-Point Touchpad Interaction," *Proc. Conf. Multimedia and Expo (ICME 03)*, IEEE CS Press, 2003, pp. 49–52.
2. J. Rekimoto, "SmartSkin: An Infrastructure for Free-hand Manipulation on Interactive Surfaces," *Proc. Conf. Human Factors in Computing Systems*, ACM Press, 2002, pp. 113–120.
3. P. Dietz and D. Leigh, "DiamondTouch: A Multi-User Touch Technology," *Proc. ACM Symp. User Interface Software and Technology (UIST 01)*, ACM Press, 2001, pp. 219–226.
4. A.D. Wilson, "PlayAnywhere: A Compact Interactive Tabletop Projection-Vision System," *Proc. ACM Symp. User Interface Software and Technology (UIST 05)*, ACM Press, 2005, pp. 83–92.
5. A.D. Wilson, "TouchLight: An Imaging Touchscreen and Display for Gesture-Based Interaction," *Proc. Conf. Multimodal Interfaces*, ACM Press, 2004, pp. 69–76.
6. J.Y. Han, "Low-Cost Multi-Touch Sensing through Frustrated Total Internal Reflection," *Proc. ACM Symp. User Interface Software and Technology (UIST)*, ACM Press, 2005, pp. 115–118.
7. H. Benko, A.D. Wilson, and P. Baudisch, "Precise Selection Techniques for Multi-Touchscreens," *Proc. Conf. Human Factors in Computing Systems*, ACM Press, 2006, pp. 1263–1272.
8. K. Everitt et al., "Modal Spaces: Spatial Multiplexing to Mediate Direct-Touch Input on Large Displays," *Proc. Human Factors in Computing Systems*, ACM Press, 2005, pp. 1359–1362.

Philip I.S. Lei is an associate professor at Macau Polytechnic Institute and a PhD candidate in computer science at Sun Yat-Sen University. His research interests include user interface design, social network analysis, and data mining. Contact him at philiplei@ipm.edu.mo.

Angus K.Y. Wong is an associate professor at Macao Polytechnic Institute. His research interests include Internet caching systems, network infrastructure security, and human-computer interactions. Wong has a BSc and a PhD in information technology from the City University of Hong Kong. Contact him at kywong@ipm.edu.mo.

Join the IEEE Computer Society
www.computer.org

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.