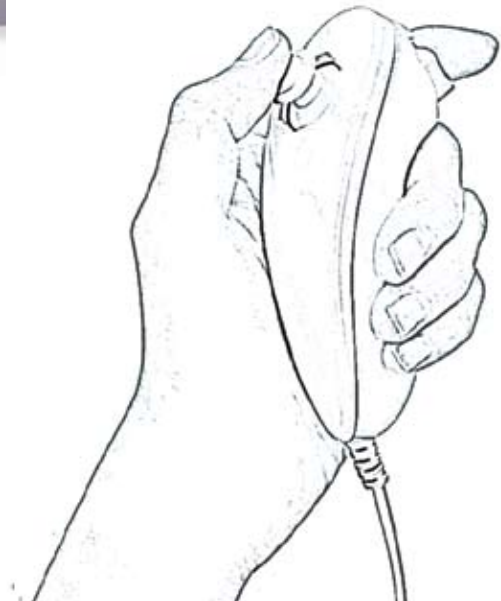




BY ROBERT CRAVOTTA • TECHNICAL EDITOR

# RECOGNIZING GESTURES

BLURRING THE LINE BETWEEN  
HUMANS AND MACHINES





he most basic and simplest gesture is pointing, and it is an effective method for most people to communicate with each other, even in the presence of language barriers. However, pointing quickly fails as a way to communicate when the object or concept that a person is trying to convey is not in sight to point at. Taking gesture recognition beyond simple pointing greatly increases the type of information that two people can communicate

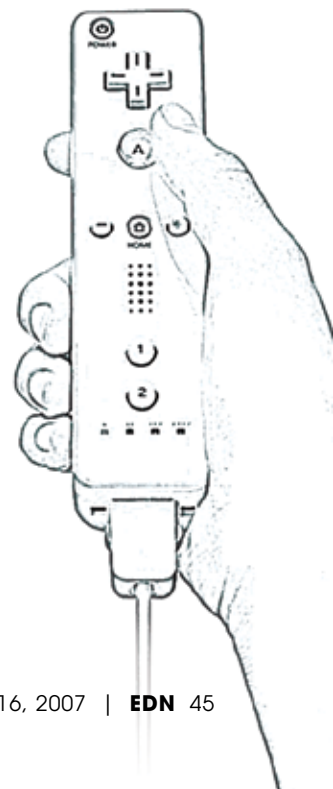
with each other. Gesture communication is so natural and powerful that parents are increasingly using it to enable their babies to engage in direct, two-way communication with their care givers, through baby sign language, long before the babies can clearly speak (**Reference 1**).

The level of communication between users and their electronic devices has been largely limited to a pointing interface. To date, a few common extensions to the pointing interface exist. They include single- versus double-click or tap devices and devices that allow users to hold down a button while moving the pointing focus, such as mice, trackballs, and touchscreens. A user's ability to naturally communicate with a computing device through a gesture interface and a speech-recognition interface, such as a multitouch display or an optical-input system, is still largely an emerging capability. Consider the new and revolutionary mobile phone that relies on a touchscreen-driven user interface instead of physical buttons and uses a predictive engine that helps users with typing on the flat panel. This description could apply to Apple's iPhone, which the company launched in June, but it can also apply to the IBM Simon, which the company launched with Bell South in 1993, 14 years earlier than the iPhone. Differences exist between the two touch interfaces. For example, the newer units support multitouch gestures, such as "pinching" an image to size it and flicking the display to scroll the content. This article touches on the nature of how gesture interfaces are evolving and what they mean for future interfaces.

Much of the technology driving many of today's latest and innovative gesture-like interfaces is not exactly new: Most of these interfaces can trace their heritage in products or projects from the

past few decades. According to **Reference 2**, multitouch panel interfaces have existed for at least 25 years, and that length of time is on par with the 30 years that elapsed between the invention of the mouse in 1965 and the mouse's reaching its tipping point as a ubiquitous pointing device, which happened with the release of Microsoft Windows 95. Improvements in the hardware for these types of interfaces enable designers to shrink and lower the cost of end systems. More important, however, these improved interfaces enable designers to leverage additional low-cost software-processing capacity to use it to better identify more contexts so they can better interpret what a user is trying to tell the system to do. In other words, most of the advances in emerging gesture interfaces will come not so much from new hardware as from more complex software algorithms that best use the strengths and compensate for the weaknesses of each type of input interface. **Reference 3** provides a work-in-progress directory of sources for input technologies.

In addition to the commercial launch of the iPhone, this year has borne witness to the Korean and European launch of the LG Electronics-manufactured, Prada-designed LG Prada phone, the successful commercial launch of Nintendo's Wii gesture-interface console, and the pending launch of the multitouch Microsoft Surface Platform (see sidebar "Multitouch surfaces"). Are the lessons designers learned from previous iterations of gesture interfaces sufficient





to give today's latest innovative products the legs they need to survive more than a year or two and finally usher in the promising age of more natural communication between humans and machines? These platforms have access to large amounts of memory and worldwide connectivity through the Internet for software updates. So, perhaps the more relevant question is: Can the flexible, programmable nature of these platforms enable the gesture interfaces to adjust to the set of as-yet-unlearned lessons without going back to the drawing board?

Gesture-recognition interfaces are not limited to just gaming and infotainment products. Users of Segway's PTs (personal transporters) intuitively command their transporters by leaning in the appropriate direction to move forward, stop, and turn left or right (Figure 1). Some interfaces focus on capturing a rich range of subtle gestures to emulate using a real-world tool rather than issuing abstract commands to a computer. For example, Wacom's Intuos and Cintiq tablets coupled with tablet-enhanced paint- and graphics-software programs can faithfully capture an artist's hand and tool motions in the six dimensions of up and down, left and right, downward pressure on the tablet surface, stylus-tilt angle, stylus-tilt direction, and stylus rotation. This feature enables the software to recreate not only the gross motions, but also the fine motions, such as twisting a user's hand to more realistically emulate the behavior of complex objects, such as paint and drawing tools.

Another example of capturing subtle motions to enable the emulation of the direct manipulation of real-world tools is Intuitive Surgical's da Vinci Surgical System. This system employs a proprietary 3-D-vision system and two sets of robots—the masters and the EndoWrist instruments—to faithfully translate the intent of a surgeon's hand and finger motions on the masters to control the EndoWrist instruments during robotic laparoscopic surgery (Figure 2). Decoupling the surgeon's hand motions from the on-site surgical instruments through the masters not only allows the surgery to require only a few small cuts to insert the surgical tools into the patient, but also affords the surgeon a better posture to delay the onset of fatigue when per-

## AT A GLANCE

- Many of the gesture interfaces we see in innovative products can trace their roots back several decades.
- Gesture interfaces find more use than just in games and infotainment devices; they also control systems in industrial and medical environments.
- Much of what makes a gesture interface reliable and useful, such as inferring or predicting intent, is not obvious to the user.
- The success of an interface is in how well it handles uncertainty with the user.
- Devices with modern interfaces must consider how to manage wireless and network connectivity between systems so that they appear as one system to the user.

forming long procedures. It also enables greater surgical precision, an increased range of motion, and improved dexter-



Figure 1 The Segway PT interface translates the leaning direction of the user into commands to move the PT in a direction (courtesy Segway).

ity through digital filtering than if the surgeon directly manipulates the surgical tools, such as in traditional laparoscopic surgery.

The 3-D-vision system is a critical feedback interface that enables surgeons to effectively use the da Vinci Surgical System and avoid mistakes. Additionally, the system complements the visual-feedback interface with some simple haptics or force feedback such as that to detect when internal and external collisions occur during a motion. Research organizations, such as at Johns Hopkins University, are using the da Vinci Surgical System to study technologies that support a "sense of touch." "The da Vinci is a perfect 'laboratory,' as it provides high-quality motion and video data of a focused and stylized set of goal-directed tasks," says Gregory D Hager, professor of computer science at Johns Hopkins. "We envision using the statistical models we develop as a way of making the device more 'intelligent' by allowing it to recognize what is happening in the surgical field."

## UNSEEN POTENTIAL

"Great experiences don't happen by accident," says Bill Buxton, principal researcher at Microsoft. "They are the result of deep thought and deliberation." His decidedly low-tech example involves two manual juicers that look similar and have the same user interface (Reference 4). If you can use one, you can use the other. The juice tastes the same from each, and each takes the same amount of time to make the juice. However, they differ in the method and the timing of a user's applying the maximum force. The juicer with the "constant-gear-ratio" effect requires the user to apply the maximum force at the end of the lever pull, whereas the other juicer delivers a "variable-gear-ratio" effect that reduces the pressure the user needs to apply at the end of the lever pull. In essence, the qualitative difference between the juicers is the result of nonobvious mechanisms hidden in the interface.

These examples of gesture-recognition interfaces are direct-control interfaces, in which users explicitly tell or direct the system to do what they want. However, the emerging trend for embedded or "invisible" human-machine



interfaces is an area of even greater potential. Embedded processing, which is usually invisible to the end user, continues to enable designers to make their products perform more functions at lower cost and with better energy efficien-

cy. As the cost of sensors and processing capacity continue to drop and the processors are able to optimize the essential functions of the systems they control, an opportunity arises for the extra available processing to provide an im-

plicit or embedded human-machine interface between the user and the system. In other words, users may imply their intent with the system without consciously being aware they are doing just that. This emerging capability is essential to

## MULTITOUCH SURFACES

Multitouch interfaces have existed in some form for the last 25 years, and the time of their ubiquitous adoption is either fast approaching or upon us with this year's commercial offerings, such as the Apple iPhone and the Microsoft Surface (reference A and B). These multitouch displays enable users to operate directly on the displayed objects with their hands and fingers rather than mentally correlate the position of an on-screen cursor with the motion of a pointer, such as from a mouse. The multitouch interfaces offer a richer array of interactions than single-touch or single-focus interfaces that are common today. Apple based the iPhone's multitouch interface on a capacitive-touch technology that limits interactions to only those from the user's fingers; it supports gestures such as flicking the screen

to scroll content and "pinching" the screen to zoom in and out on content.

Perceptive Pixel has released a demonstration video showcasing the company's work with a large multitouch display. It shows a variety of gestures and contexts that could benefit from a multitouch interface (Reference C). At press time, no additional information other than the video was available; however, a few items are worth noting. The display-and-touch-panel system is on a wall and is much larger than a typical display available to consumers today. In many of the scenes, more than one person is operating the touch display at once; sometimes, they are working together, and, at other times, they are working independently on different objects. The operator is using both of his hands at the same time during most of

the video, and he effects a tremendous number of actions in a short period. The examples of manipulating 3-D virtual objects are probably harbingers of things to come. Finally, the room is dark, which suggests that the sensor implementation is not appropriate for all environments; however, other sensor implementations could deliver similar sensitivity in different environments.

Microsoft announced the table-top-like Surface multitouch-display interface in May, and the company expects production equipment to be available in November. The platform works by shining a near-infrared, 850-nm-wavelength light source on the bottom of the table's surface and using multiple infrared cameras to detect reflections of that light when objects and fingers touch the surface of the display (Figure A). The use of the near-infrared light allows users to employ the table in ambient light. A textured diffuser over the display causes the near-infrared light to reflect back to the cameras under the table in a way that allows the software to meaningfully identify fingers, hands, motions, and other real-world objects. The platform supports many of the same types of direct-interaction gestures as the other multitouch examples, such as flicking and pinching, but the interface adds

a new twist: It can interact with dozens of real-world objects in addition to the hands and fingers of many users at once. The table-top form factor is natural for supporting face-to-face collaboration among multiple people, electronic content, and real-world objects.

The ability of the platform to bridge real-world objects with virtual objects is profound for gesture-interface actions. Users can place wireless devices on the display, and the platform recognizes, invisibly establishes communications links with, and identifies them with a circle around them on the surface display. The user can drag content, such as photographs, from one device to the surface interface of another device on the table. The data transfer can eliminate the need for cables between the devices, and the transfer of virtual objects between the real-world devices can consist of natural dragging and dropping gestures.

### REFERENCE

- A Microsoft Surface, [www.microsoft.com/surface](http://www.microsoft.com/surface).
- B "Microsoft Surface: Behind-the-Scenes First Look (with Video)," *Popular Mechanics*, July 2007, [www.popularmechanics.com/technology/industry/4217348.html](http://www.popularmechanics.com/technology/industry/4217348.html).
- C Perceptive Pixel, "Multitouch Demonstration Video," [www.perceptivepixel.com](http://www.perceptivepixel.com).

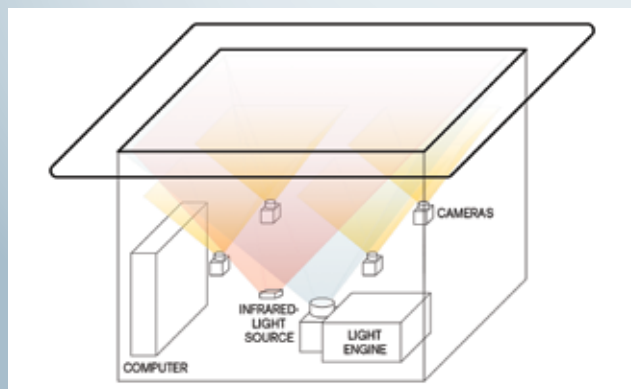


Figure A A conceptual (not accurate for intellectual-property reasons) artist's rendition shows the Microsoft Surface components (courtesy Microsoft).



enabling systems to use predictive compensation to better accommodate a user's inexperience or errors and allowing the system to still perform what the user intended.

The Simon's PredictaKey keyboard explicitly listed to the user its top six predicted-letter candidates and allowed the user to explicitly select from that list. To take advantage of the prediction engine, the user had to explicitly engage with the engine's suggestions and choose from them. In contrast, the iPhone's typing interface manifests itself in several obvious and hidden ways to improve typing speed and accuracy. First, it presents specialized key layouts for each application so that only keys that are relevant are available for input. As the user types, the system may predict the word and present it to the user while they are typing; if the word is correct, the user can select it by pressing the space "key" on the display or just continue typing. Likewise, the system tries to identify potentially misspelled words and presents the word with the correct spelling in a similar fashion to allow the user to accept or ignore the proposed correction.

However, the new and invisible magic in the iPhone typing interface is that it compensates for the possibility of the user's pressing the wrong letter on the display panel by dynamically resizing the target area or tap zone assigned to each letter without changing the display size of any of the letters, based on its typing engine's predictions of what letter the user will select next (**Reference 5**). The letters that the prediction engine believes the user may press next receive a larger tap zone that can overlap with the display area of nearby, lower probability letters, which receive a smaller tap zone as a result. This feature increases the chances of selecting the predicted letter and decreases the chances of selecting an unpredicted letter that is adjacent to the predicted letter.

Although not considered a strict user interface, such as that between a user and a computer, some automobile-safety features implement an early form of implicit communication interfaces for predictive-safety features. As an example, to determine whether to warn the driver of an imminent lane departure, the system can examine the turn signal to de-



**Figure 2** The da Vinci Surgical System combines two robotic systems, the masters and the EndoWrist Instruments, with a 3-D-vision system to enable surgeons to better perform complex laparoscopic surgical procedures (courtesy Intuitive Surgical).

termine whether the impending lane departure is intentional or accidental. People unintentionally and implicitly communicate their presence to passenger-detection systems that may control whether safety systems should deploy in the event of an accident. For example, the automobile may adjust how the air bag deploys to avoid certain types of injuries for passengers of different sizes. Electronic stability-control systems can compare the driver's implied intention, by examining the steering and braking inputs, with the vehicle's actual motion; they can then appropriately apply the brakes on each wheel and reduce engine power to help correct understeer (plowing), oversteer (fishtailing), and drive-wheel slippage to help the driver maintain some control of the vehicle.

The control systems for the highest maneuverable fighter aircraft offer some insight into the possible future of consumer-level control of complex systems. Because these aircraft employ high levels of instability to realize their maneuverability, the pilot can no longer explicitly and directly control the aircraft

subsystems; rather, the embedded-processing system handles those details and enables the pilot to focus on higher level tasks. As automobile-control systems can better predict a driver's intentions and correlate those intentions with the state of the vehicle and the surrounding environment, they may be able to deliver even higher levels of energy efficiency by reducing energy loads in situations in which they are currently unnecessary—without sacrificing safety. In each case, the ability of the system to better understand the user's intention and act appropriately correlates to the system's ability to invisibly and accurately predict what the user can and might do next.

No matter how rich and intuitive an interface is, its ultimate success and adoption depend on how well the user and the system can signal each other and compensate for the possible range of misunderstandings. Uncertainty or unpredictability between how to command a system and its resultant behavior can kill the immediate usefulness and delay the adoption of the gesture interface. Merely repetitively informing the



## I'LL COMPENSATE FOR YOU

Modern interfaces are often more complex than they appear because they embody years of lessons designers have learned to compensate for how users view and interact with a system. The iPhone's predictive-typing engine couples with the dynamically resized tap zones to provide an example of the system's compensating for the user to reduce the frequency of unintentional inputs. This technology builds on and extends the techniques of past systems employed to improve key- or tap-based communication between user and system. One such older technology is keyboard-debounce filtering, which eliminates scenarios that occur when the system improperly interprets a single key press as multiple key presses because of transient properties of the input device.

The history of keyboard debounce illustrates a possible life cycle for an error-compensation mechanism. Errors were issues for early systems with electronic keyboards or touch displays that did not filter for debouncing. In these systems, the user is responsible for determining when the system misinterpreted a single key press as multiple presses. This situation places a low-value cognitive load on the user that can add to a user's frustration when using the interface. By filtering away debounce conditions, the user is free to focus on higher level tasks. Solving the debounce problem was once a differentiating feature, but it is now an assumed and normal capability.

The history of the delete-confirmation mechanism that many systems use today exemplifies how an interface capability can evolve to accommodate a better understanding of how users view the system. The delete-confirmation mechanism evolved from users' accidental deletion of data, such as files. With a command-line interface, a user could accidentally delete a file

by using wild cards to specify an unintended file name for deletion. A pointing interface, such as a keyboard or a mouse cursor, allows accidental file deletion because the file name is based on where the cursor is pointing at the moment of the delete command.

An early change to user interfaces to compensate for this type of error was to ask the user to confirm the deletion; the user could verify the file name and spot an error before it occurred. A problem with this mechanism is that the confirmation applied to every deletion, and it easily became a mindless and automated key press or pointer click that lost its effectiveness as a safety step over a short time. The next compensation was to provide an undelete command to "fix" the failings of the deletion confirmation, and many systems now allow the user to skip or avoid the deletion-confirmation notice because it is often considered a low-value and high-noise way to protect data. A "trash-can" or "recycle bin" icon has replaced the undelete command; these features allow users to recover many deleted files. An analogous evolution occurred for in-application deletion of data with the introduction of the undo command and the eventual improvement of a multistep undo capability common in modern applications.

With each new compensation mechanism, the system took on more responsibility for understanding what users needed, even when that meant allowing users to reverse a previously irreversible action. At each stage of this evolution, the interface supported mechanisms that users could misunderstand or misuse. Each new compensation incorporated the lessons designers had learned about how the user might interact with the system to avoid future unwanted outcomes.



user that there is an error is insufficient in modern electronic equipment. These devices often guide the user about the nature of the error or misunderstanding and how they might correct the condition. Modern interfaces employ a combination of sensors, improved processing algorithms, and user feedback. This combination provides a variety of mechanisms to reduce ambiguity and uncertainty between the user and the system so that each can more quickly and meaningfully compensate for unexpected behavior of the other (see sidebar “I’ll compensate for you”).

One way to compensate for potential misunderstandings is for the system to control and to reduce the set of possible inputs to only those with a valid context, such as with the iPhone’s specialized key layouts. Applications that can segment and isolate narrow contexts and apply strong goal-defined tasks in each one are good candidates for this type of compensation. Handwriting systems based on the Graffiti recognition system, such as Palm PDAs, improved the usability of a handwriting interface by narrowing the possibility for erroneous inputs, but doing so involved a significant learning curve for users before they could reliably use the system. Speech-recognition systems that require no training from a speaker increase their success rate by significantly limiting the number of words the systems can recognize, such as the 10 digits, or by presenting the user with a short menu of responses.

Another method of compensating for misunderstandings is to eliminate or move translations from the user to the system. HP Labs India is working with a pen-based device, the GKB (gesture keyboard), which allows users to enter phonetic scripts, such as Devanagari and Tamil scripts, as text input without the benefit of a language-specific keyboard. Another example is the Segway PT that once required a user to translate a forward and backward twist to correspond to a signal to turn left or right. Now, it instead allows the user to indicate left or right by leaning in the desired direction. In this case, the newer interface control removes the ambiguity of which twist direction aligns with which turn direction, and it aligns the control with the natural center-of-grav-

## MORE AT EDN.COM



⊕ [Go to \*\*www.edn.com/070816cs\*\* for a sign-language alphabet. At this link, you can also click on \*\*Feedback Loop\*\* to post a comment on this article.](http://www.edn.com/070816cs)

⊕ [For a related article about natural-interface input devices, go to \*\*www.edn.com/article/CA263121\*\*.](http://www.edn.com/article/CA263121)

⊕ [For a related article about interfacing electronics to people, go to \*\*www.edn.com/article/CA6309109\*\*.](http://www.edn.com/article/CA6309109)

⊕ [For a related article about smarter vehicles, go to \*\*www.edn.com/article/CA6339246\*\*.](http://www.edn.com/article/CA6339246)

ity use scenario for the system, which greatly increases its chances as a useful and sustainable interface.

Another important way to compensate for potential errors or misunderstandings is to give users enough relevant feedback so that they can appropriately change their expectations or behavior. Visual feedback is a commonly used mechanism. The mouse cursor on most systems performs more functions than just acting as a pointing focus; it also acts as a primary feedback to the user about when the system is busy and why. The success of the gesture interface with the Wii remote hinges in part on how well the system software improves over time to provide better sensitivity to player gestures. It also depends on how well it provides feedback, such as a visual cue on the display, that points out how users can make small adjustments to their motions so that the system properly interprets their intended gestures.

Haptic or tactile feedback engages the user’s sense of touch; it is a growing area for feedback, especially as a component of multimodal feedback involving more than a single sense. Game consoles have employed rumble features for years in their handheld controllers. The Segway PT signals error conditions to the user through force feedback in the control stick. The da Vinci Surgical System uses force feedback to signal boundary collisions, such as when the EndoWrist instrument makes contact with the surface of the cutting target. Haptic feedback can compensate for the weaknesses of other feedback methods, such as audio sounds in noisy environments.

Haptic feedback can also help offload



the visual sensory overload by freeing the user's eyes from seeking visual confirmation that the system has received an input to focus his eyes on other details. For example, the iPhone keypad does not implement haptic feedback to signal the user which key was pressed and when, so the user must visually confirm each key press the system processes. One company, Immersion, offers a way to simulate a tactile sensation for mobile devices by issuing precise pulse control over a device's vibration actuator within a 5-msec window of the input event.

When all other compensation methods fail to eliminate a misunderstanding, designers can employ a context-relevant response to address the uncertainty of a given input. A common response type is to issue a warning and to ask the user to repeat the input, but this situation risks frustrating the user if the system repeatedly requests the input with no additional guidance about what it needs. The system can make a best guess as to what the input was and ask the user to confirm that the guess is correct; this scenario also can cause frustration to the user if no method is available to refine the guess on a second try or if the system must too often confirm an input. A possible strategy for minimizing the use of these types of responses is for the system to profile the user's behavior and develop statistical models to better correlate guesses with what the user requests most frequently.

Gene Frantz, principal fellow at Texas Instruments, observes that the size of a system is scalable when you consider that networks can tie systems together. This consideration is increasingly important for modern devices. Consider that the iPhone, Wii, and Microsoft Surface include wireless-communication links with other systems. How these devices interact with other external systems correlates with how well they meet the needs of their users. Even as the world of gesture interfaces begins to stand on its electronic feet, we are increasing our expectations for our devices to apply the lessons we have learned for interacting with a single device to multiple devices seamlessly interacting with the user and each other. Those systems that can best predict the user's intent to minimize and avoid uncertainty and seamlessly pull

together other systems in a connected world will likely drive the future of gesture interfaces. **EDN**

## REFERENCES

- 1 "Signing with your baby," [www.signingbaby.com](http://www.signingbaby.com).
- 2 Buxton, Bill, "An Incomplete Roughly Annotated Chronology of Multi-Touch and Related Work," from *Multi-Touch Systems That I Have Known and Loved*, [www.billbuxton.com/multitouch/Overview.html](http://www.billbuxton.com/multitouch/Overview.html).
- 3 Buxton, Bill, "A directory of sources for input technologies," [www.billbuxton.com/InputSources.html](http://www.billbuxton.com/InputSources.html).
- 4 Buxton, Bill, "Experience Design vs Interface Design," *Rotman Magazine*, Winter 2005, pg 47, [www.billbuxton.com/experienceDesign.pdf](http://www.billbuxton.com/experienceDesign.pdf).
- 5 "iPhone Keyboard," [www.apple.com/iphone/usingiphone/keyboard.html](http://www.apple.com/iphone/usingiphone/keyboard.html).

## FOR MORE INFORMATION

**Analog Devices**  
[www.analog.com](http://www.analog.com)

**Apple**  
[www.apple.com](http://www.apple.com)

**Celestron**  
[www.celestron.com](http://www.celestron.com)

**Cypress Semiconductor**  
[www.cypress.com](http://www.cypress.com)

**Freescale**  
[www.freescale.com](http://www.freescale.com)

**GestureTek**  
[www.gesturetek.com](http://www.gesturetek.com)

**Gyration**  
[www.gyration.com](http://www.gyration.com)

**HP Labs India**  
[www.hpl.hp.com/india/](http://www.hpl.hp.com/india/)

**IBM**  
[www.ibm.com](http://www.ibm.com)

**Immersion**  
[www.immersion.com](http://www.immersion.com)

**Intuitive Surgical**  
[www.intuitivesurgical.com](http://www.intuitivesurgical.com)

**Johns Hopkins University**  
[wse.jhu.edu](http://wse.jhu.edu)

**LG Electronics**  
[www.lge.com](http://www.lge.com)

**Microchip**  
[www.microchip.com](http://www.microchip.com)

**Microsoft**  
[www.microsoft.com](http://www.microsoft.com)

**Nintendo**  
[www.nintendo.com](http://www.nintendo.com)

**NXP Semiconductors**  
[www.nxp.com](http://www.nxp.com)

**Palm**  
[www.palm.com](http://www.palm.com)

**Perceptive Pixel**  
[www.perceptivepixel.com](http://www.perceptivepixel.com)

**Prada**  
[www.prada.com](http://www.prada.com)

**Segway**  
[www.segway.com](http://www.segway.com)

**Sony**  
[www.sony.com](http://www.sony.com)

**STMicroelectronics**  
[www.st.com](http://www.st.com)

**Texas Instruments**  
[www.ti.com](http://www.ti.com)

**Wacom**  
[www.wacom.com](http://www.wacom.com)

**Xsens Technologies**  
[www.xsens.com](http://www.xsens.com)

You can reach  
Technical Editor  
**Robert Cravotta**  
at 1-661-296-5096  
and [rcravotta@edn.com](mailto:rcravotta@edn.com).

