SmartSkin: An Infrastructure for Freehand Manipulation on Interactive Surfaces

Jun Rekimoto
Interaction Laboratory
Sony Computer Science Laboratories, Inc.
3-14-13 Higashigotanda
Shinagawa-ku, Tokyo 141-0022, Japan
Phone: +81 3 5448 4380
Fax: +81 3 5448 4273
Mail: rekimoto@acm.org
http://www.csl.sony.co.jp/person/rekimoto.html

ABSTRACT
This paper introduces a new sensor architecture for making interactive surfaces that are sensitive to human hand and finger gestures. This sensor recognizes multiple hand positions and shapes and calculates the distance between the hand and the surface by using capacitive sensing and a mesh-shaped antenna. In contrast to camera-based gesture recognition systems, all sensing elements can be integrated within the surface, and this method does not suffer from lighting and occlusion problems. This paper describes the sensor architecture, as well as two working prototype systems: a table-size system and a tablet-size system. It also describes several interaction techniques that would be difficult to perform without using this architecture.

Keywords
Interactive surfaces, gesture recognition, augmented tables, two-handed interfaces, touch-sensitive interfaces.

INTRODUCTION
Many methods for extending computerized workspace beyond the computer screen have been developed. One goal of this research has been to turn real-world surfaces, such as tabletops or walls, into interactive surfaces [23, 21, 16, 20, 9]. The user of such a system can manipulate, share, and transfer digital information in situations not associated with PCs. For these systems to work, the user's hand positions often must be tracked and the user's gestures must be recognizable to the system. Hand-based interaction offers several advantages over traditional mouse-based interfaces, especially when it is used in conjunction with physical interactive surfaces.

While camera-based gesture recognition methods are the most commonly used (such as [24, 13, 9]), they often suffer from occlusion and lighting condition problems. To correctly capture hand images on a surface, a camera must be mounted above the table or in front of the wall. As a result, the system configuration becomes complex, making it difficult to implement the system as a portable (integrated) unit. The use of magneto-electric sensors (e.g., Polhemus [15]) is another possible sensing method, but it requires attaching a tethered magneto-electric sensor to each object being tracked.

CHI 2002, April 20-25, 2002, Minneapolis, Minnesota, USA.

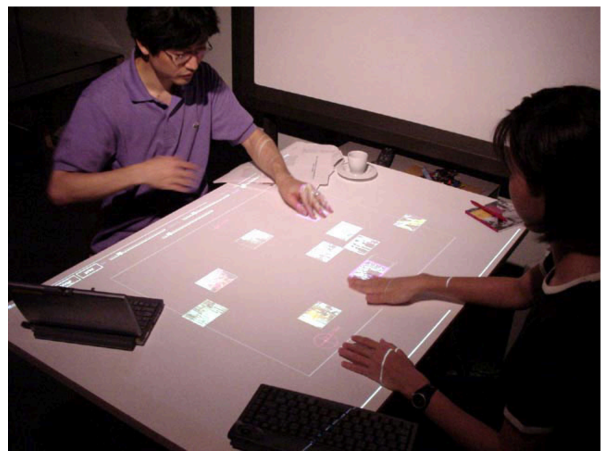

Figure 1: An interactive surface system based on the SmartSkin sensor.

This paper introduces a new sensing architecture, called SmartSkin, which is based on capacitive sensing (Figure 1). Our sensor accurately tracks the position of the user's hands (in two dimensions) and also calculates the distance from the hands to the surface. It is constructed by laying a mesh of transmitter/receiver electrodes (such as copper wires) on the surface. As a result, the interactive surface can be large, thin, or even flexible. The surface does not need to be flat – i.e., virtually any physical surface can interactive. By increasing the density of the sensor mesh, we can accurately determine the shape of the hand and detect the different positions of the fingers. These features enable interaction techniques that are beyond the scope of normal mouse-based interactions.

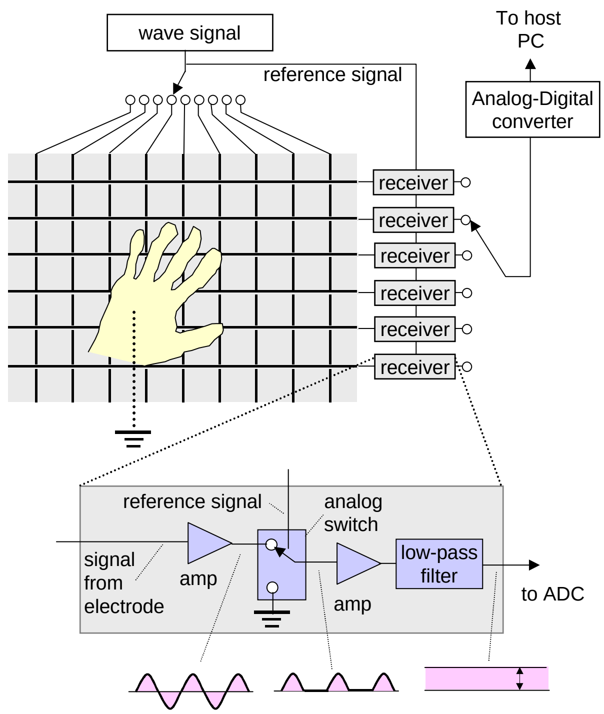

Figure 2: The SmartSkin sensor configuration: A mesh-shaped sensor grid is used to determine the hand's position and shape.

We describe the sensing principle of SmartSkin and two working systems: an interactive table system and a hand-gesture sensing tablet. We also describe new interaction techniques of these systems.

SMARTSKIN SENSOR ARCHITECTURE

Figure 2 shows the principle of operation of the SmartSkin sensor. The sensor consists of grid-shaped transmitter and receiver electrodes (copper wires). The vertical wires are transmitter electrodes, and the horizontal wires are receiver electrodes. When one of the transmitters is excited by a wave signal (of typically several hundred kilohertz), the receiver receives this wave signal because each crossing point (transmitter/receiver pairs) acts as a (very weak) capacitor. The magnitude of the received signal is proportional to the frequency and voltage of the transmitted signal, as well as to the capacitance between the two electrodes. When a conductive and grounded object approaches a crossing point, it capacitively couples to the electrodes, and drains the wave signal. As a result, the received signal amplitude becomes weak. By measuring this effect, it is possible to detect the proximity of a conductive object, such as a human hand.

The system time-dividing transmitting signal sent to each of vertical electrodes and the system independently measures values from each of receiver electrodes. These values are integrated to form two-dimensional sensor values, which we called "proximity pixels". Once these values are obtained, algorithms similar to those used in image processing, such

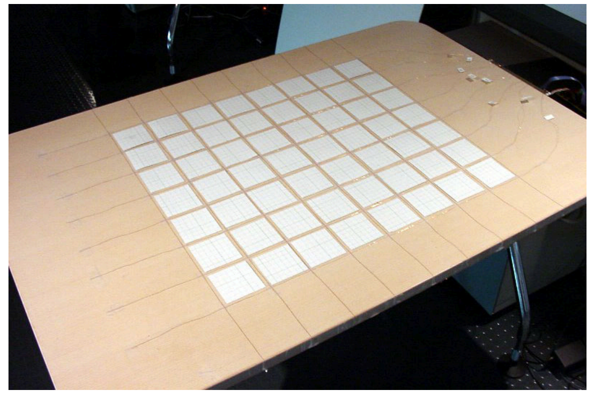

Figure 3: Interactive table with an 8 $\times$ 9 SmartSkin sensor: A sheet of plywood covers the antennas. The white squares are spacers to protect the wires from the weight of the plywood cover.

as peak detection, connected region analysis, and template matching, can be applied to recognize gestures. As a result, the system can recognize multiple objects (e.g., hands). If the granularity of the mesh is dense, the system can recognize the shape of the objects.

The received signal may contain noise from nearby electric circuits. To accurately measure signals only from the transmitter electrode, a technique called "lock-in amplifier" is used. This technique uses an analogue switch as a phase-sensitive detector. The transmitter signal is used as a reference signal for switching this analog switch, to enable the system to select signals that have the synchronized frequency and the phase of the transmitted signal. Normally, a control signal needs to be created by phase-locking the incoming signal, but in our case, the system can simply use the transmitted signal, because the transmitter and the receiver are both on the same circuit board. This feature greatly simplifies the entire sensor design.

We chose a mesh-shaped electrode design for our initial experiment because of its simplicity and suitability for sensing hand shapes as pixel patterns. Other layouts are possible, depending on the application requirements. For example, the density of the mesh can be adjusted. In addition, since the electrodes are simply thin copper wires, it is possible to create a very thin interactive surface such as interactive paper, which can even be flexible.

PROTOTYPE 1: AN INTERACTIVE TABLE

Based on the principle described above, we developed two interactive surfaces: a table-size system that can track multiple hand positions, and a smaller (and more accurate) system that uses a finer electrode layout.

The table system is constructed by attaching sensor elements to a wooden table. A mesh-like antenna, made of polyurethane-coated 0.5 mm-thick copper wire, is laid on the tabletop. The number of grid cells is 8$\times$9, and each grid cell is 10$\times$10 cm. The entire mesh covers an 80$\times$90 cm area of the tabletop

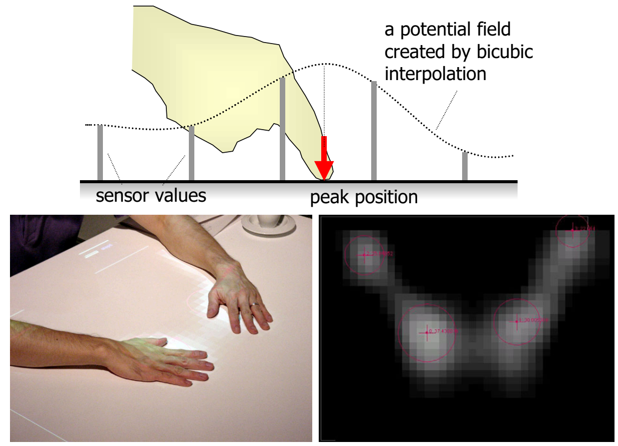

Figure 4: top: A bicubic interpolation method is used to detect the peak of the potential field created by hand proximity. bottom: arms on a table and a corresponding potential field.

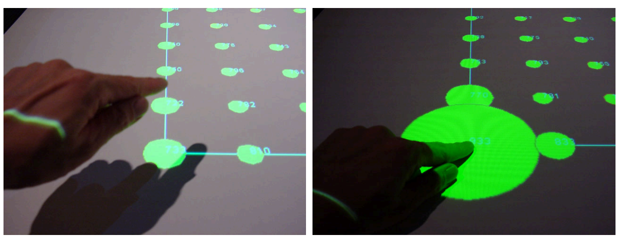

Figure 5: Relationship between distance between hand and sensor and sensed values. The diameter of the circle represents the amplitude of the sensed value.

(Figure 3). A plywood board covers the antennas. A signal transmitter / receiver circuit is attached to the side of the table. Two Atmel AVR microprocessors control this circuit. One microprocessor generates square-wave signals (400 KHz) with firmware that directly controls the I/O port, and the other microprocessor with a built-in A/D converter measures the values of the received signals and transmits them to the host computer. A projector is used to display information on the table. The current implementation is capable of processing 8×9 sensor values 30 times per second.

When the user's hand is placed within 5-10 cm from the table, the system recognizes the effect of the capacitance change. A potential field is created when the hand is in the proximity to the table surface. To accurately determine the hand position, which is the peak of the potential field, a bicubic interpolation method is used to analyze the sensed data (Figure 4). By using this interpolation, the position of the hand can be determined by finding the peak on the interpolated curve. The precision of this calculated position is much finer than the size of a grid cell. The current implementation has an accuracy of 1 cm, while the size of a grid cell is 10 cm.

As for the distance estimation, although there is no way to directly measure the precise distance between the hand and the table surface, we can estimate relative distance. Figure 5 shows the relationship between the hand position and obtained A/D-converted values. Our system enables detecting various levels of hand proximity, which is difficult to do with other technologies such as computer vision.

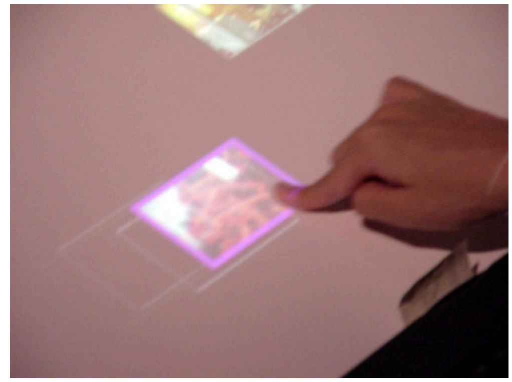

Figure 6: Mouse emulation by using calculated hand position. The distance between the hand and the surface is used to determine button-press and button-release states.

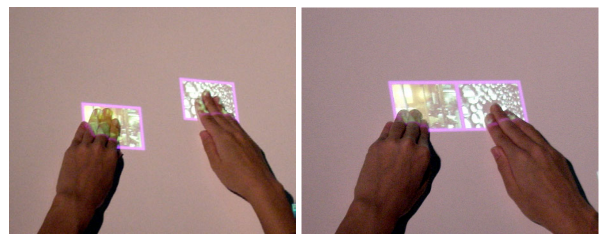

Figure 7: Two-handed operation is used to concatenate two objects.

Since each point on the grid can independently measure the proximity of an object, the system can simultaneously track more than one hand. This feature is important because many table-based applications are used by more than one user.

Interaction techniques

We studied two types of basic interaction techniques for this platform. One is 2D-position control with distance measurement, and the other uses a sensor potential field as input.

Mouse emulation with distance measurement The first interaction technique is the simple emulation of a mouse-like interface. The estimated 2D position is used to emulate moving the mouse cursor, and the hand-surface distance is used to emulate pressing the mouse button. A threshold value of the distance is used to distinguish between pressed and released states that the user can activate "mouse press" by touching the table surface with the palm, and move the cursor without pressing the mouse button by touching the table surface with the fingers. Normally, touch-sensitive panels cannot distinguish between these two states, and many interaction techniques developed for the mouse (such as "mouse over") cannot be used. In contrast, an interactive table with a SmartSkin sensor can emulate most mouse-based interfaces. Figure 6 shows how the user "drags" a displayed object.

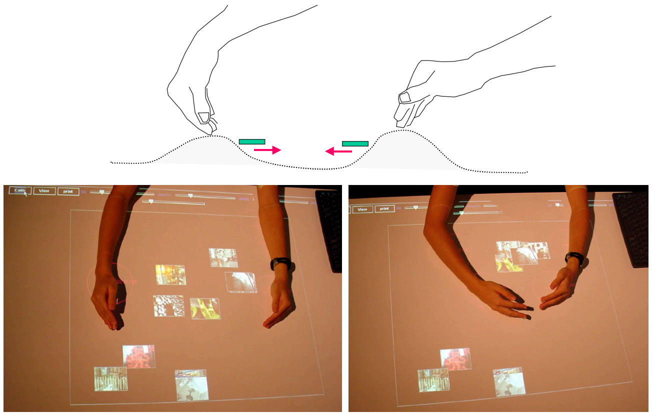

Figure 8: Shape-based object manipulation. The potential field created by the hand's proximity to the table is used to move objects. The user can use both hands or even entire arms to manipulate objects.

A notable advantage of SmartSkin over traditional mouse-based systems is its natural support for multiple-hand, multiple-user operations. Two or more users can simultaneously interact with the surface at the same time. The multiple-hand capability can also be used to enhance object manipulation. For example, a user can independently move objects with one hand. He or she can also "concatenate" two objects by using both hands, as shown in Figure 7, or can take objects apart in the same manner.

Shape-based manipulation The other interaction technique, which we call "shape-based manipulatio", does not explicitly use the 2-D position of the hand. Instead, a potential field created by sensor inputs is used to move objects. As the hand approaches the table surface, each intersection of the sensor grid measures the capacitance between itself and the hand. By using this field, various rules of object manipulation can be defined. For example, an object that "descend" to a lower potential area repels from the human hand. By changing the hand's position around the object, the direction and speed of the object's motion can be controlled.

We implemented this interface and observed how users tried to control objects. Overall, the reaction to the interface was quite encouraging. The people were quickly able to use this interface even though they did not fully understand the underlying dynamics. Many users naturally used two hands, or even arms. For example, to move a group of objects, one can sweep the table surface with one's arm. Two arms can be used to "trap" and move objects (Figure 8).

PROTOTYPE 2: A GESTURE-RECOGNITION PAD

The table prototype demonstrates that this sensor configuration can be used to create interactive surfaces for manipulating virtual objects. Using a sensor with a finer grid pitch we should be able to determine the position and shape of the hand more accurately. In addition, if the sensor can sense more than one finger position, several new interaction techniques are possible. For example, a 3D-modeling system often requires manipulation of multiple control points such as curve control points. Normally, a user of traditional mouse-based interfaces has to sequentially change these control points one by one. However, it would be more efficient and more intuitive if the user could control many points simultaneously.

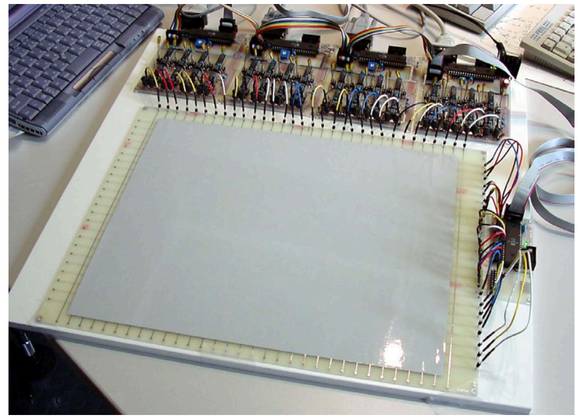

Figure 9: A gesture-recognition pad made up of a 32$\times$24 grid mesh. A sheet of plastic insulating film covers Sensor electrodes.

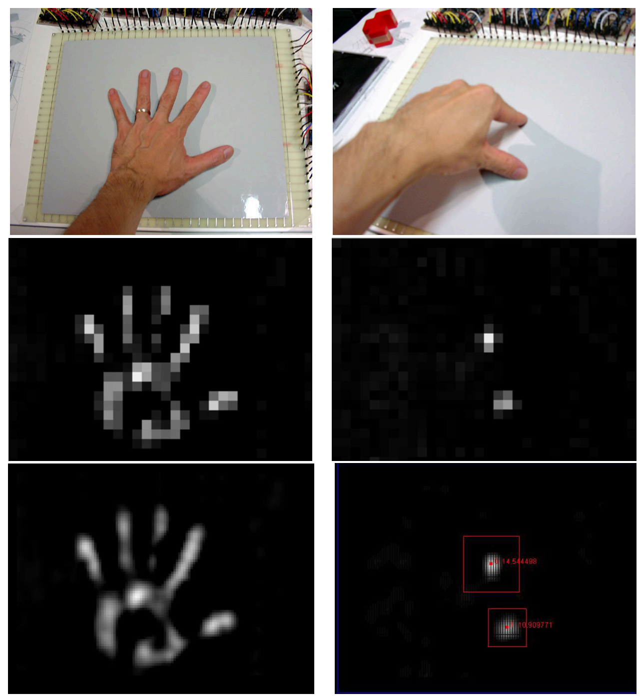

Figure 10: Gestures and corresponding sensor values. (top: a hand on the sensor mesh, middle: raw input values, bottom: after bicubic interpolation)

The second prototype uses a finer mesh pitch compared to that of the table version (the number of grid cells is 32 $\times$ 24, and each grid is 1 $\times$ 1 cm). A printed circuit board is used for the grid electrodes (Figure 9). The prototype uses the bicubic interpolation algorithm of the interactive table sys-

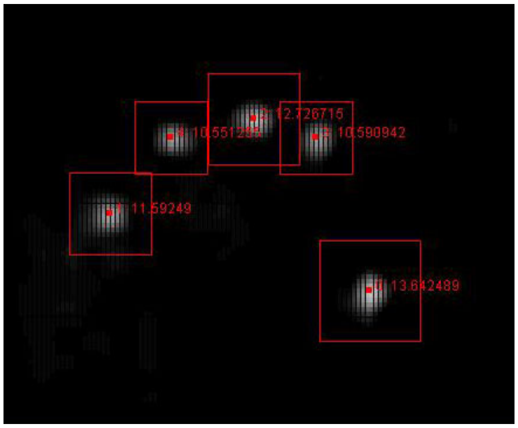

Figure 11: Fingertip detection.

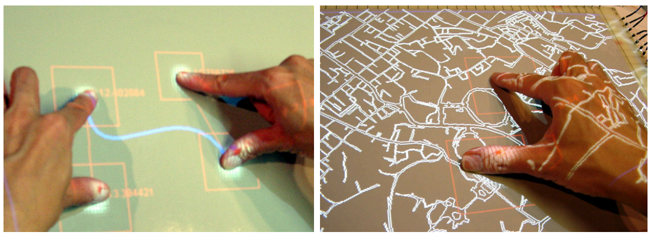

Figure 12: Examples of uses of multiple-finger interfaces: left: curve editing. right: a map browsing system. The user can use one finger for panning, or two or more fingers for simultaneous panning and scaling.

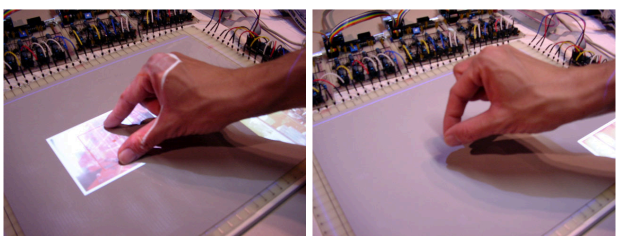

Figure 13: Two-finger gestures can be used to "pick-up" objects.

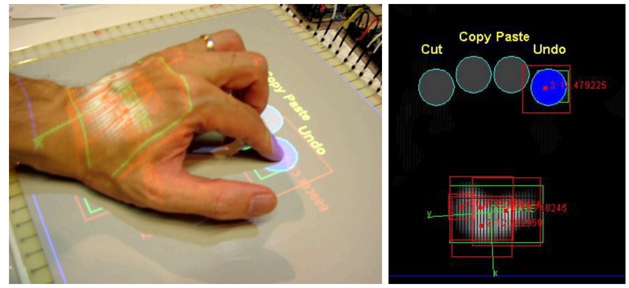

Figure 14: A palm is used to trigger a corresponding action (opening menu items). The user then taps on one of these menu items.

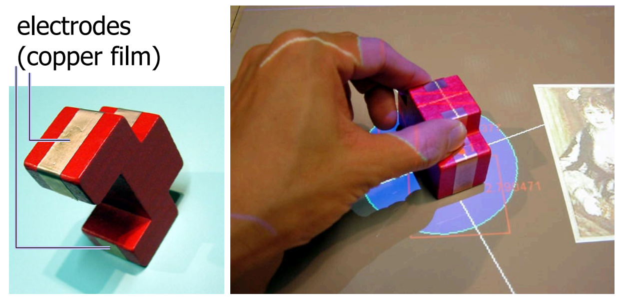

Figure 15: The "capacitance tag": a conductive pattern attached at the bottom of an object is used to identify this object.

tem, and it can determine the human hand shape as shown in Figure 10. The peak detection algorithm can also be used, and in this case, the algorithm can track multiple positions of the fingertips, not just one position of the hand (Figure 11).

Interactions by using fingers and hand gestures

We studied three possible types of interaction for this platform. The first one is (multiple) finger tracking. Here, the user simultaneously controls several points by moving his or her fingertips. The second is using hand or finger shape as input, and the third is identifying and tracking physical objects other than the user's hands.

A typical example of a situation in which the multi-finger interface is useful is diagram manipulation. A user can simultaneously move and rotate a displayed pictogram on the surface with two fingers. If three or more fingers are used, the system automatically uses a least-squares method to find the motion (consisting of moving, rotating, and scaling) that best satisfies the constraints given by the fingers.

Another simple example is the expression of attributes during manipulation. For example, the user normally drags a scroll bar with one finger, but to increase the scrolling ratio, he or she could use two or more fingers.

Figure 12 shows a map browsing system. The user scrolls the map by sliding a finger along the sensor surface. The scrolling speed increases with the number of fingers in contact with the surface. If the user touches the surface with two or more fingers, by changing the distance from the fingers to the surface, he/she can control the map scale. Simultaneous control of scrolling and zooming is intuitive, because the user feels as if his or her fingers are fixed to the map's surface.

Other possibilities we have explored include gesture commands. For example, two fingers moving toward the center of an object represent a "picking up" action (Figure 13), while a similar outward motion represents a "dropping" action. There are probably many other actions or functions representable by multi-finger gestures, for example, those based on the geographical relations between tapped fingers.

An example of using a hand shape as input is shown in Figure 14. In this example, the user places a hand on the surface, its shape is recognized by the system, and a corresponding action, in this case, "showing menu item", is triggered. The action is selected by template matching. The system first lists up connected regions (a group of sensor values that are connected), and then calculates the values of correlation between the stored templates. The system selects the region with the highest correlation value, and if this value exceeds a predetermined threshold value, the corresponding action is activated. In Figure 14, the user first touches the surface with his/her palm, then selects one of the displayed menu items.

Capacitance tags

While exploring several hand-based interaction techniques, we also found a way to make the SmartSkin sensor interact with objects besides than the hand. This feature can support graspable / tangible user interfaces [2, 8].

The principle of this method, called "capacitance tags", is shown in Figure 15. The capacitance tag block is made of a dielectric material such as wood or plastic. Some parts of this tag block are coated with a conductive material such as copper film. These conductive areas are connected to each other (by a copper wire, for example). This wire also connects the conductive areas at the bottom and at the top of the block.

When this block is placed on the SmartSkin surface, the sensor does not detect the capacitance change because the block is ungrounded. However, when a user grasps it (and touches the conductive area at the top), all the conductive areas become grounded, and areas corresponding to the conductive parts coated at the bottom of the block can be detected. Since the geometrical relationship (e.g., the distance between conductive areas) is predetermined, the system can distinguish these patterns from other patterns created when the user moves his/her hands or fingers. Essentially, the combination of conductive areas works like a barcode. In addition, the geometry of the patterns indicates the position and orientation of the tag block. Simultaneous object identification and position tracking is a key technology for many post-GUI user interface systems (such as [21, 22, 16]), and this method should be a new solution for such systems.

Another advantage of this capacitance tag method is its ability to support simultaneous hand gestures. For example, a user places a capacitance tag block on an interactive surface, and then issues a "data transfer" command by hand-dragging the displayed object toward the block.

DISCUSSIONS

Design issues

Most computers now use mice as input devices. With a mouse, the user controls one 2D position on the screen, and uses various interaction techniques, such as clicking or dragging. Although the mouse is a popular input device, its 'way' of interaction is different from the way manipulate objects in our daily lives. In the real world, we often use multiple fingers or both hands to manipulate a physical object. We control several points on the object's surface by touching, not by using "one position of the cursor" as in GUI systems. Consequently, with mouse-based interfaces, we have to unnaturally decompose some tasks into primitive operations.

In addition, our ability to interact with the physical environment is not limited to the control of multiple points. Hands and fingers can also create various phenomena, such as pressure. As a result, interaction becomes more subtle and analogue.

Related work

Capacitive sensing for human-computer interaction The idea of using capacitive sensing in the field of human-computer interfaces has a long history. Probably the earliest example is a musical instrument invented by Theremin in the early 20th century, on which a player can control the pitch and volume by changing the distance between the hand and the antenna. Other examples include a "radio drum" [11], which is also an electric musical instrument, and Lee et al.'s multi-finger touch panel, which has a sub-divided touch-sensitive surface [10].

Zimerrman et al.'s work [26] pioneered the sensing of an electric field as a method for hand tracking and data communication (e.g., "personal area network" [25]). Although there has been a lot of research in this area, interaction techniques, like the ones described in this paper, have not been studied extensively. Our other contributions to this field are the new electrode design that enables accurate and scalable interactive surfaces, and the creation of tagged physical objects that can be used in combination with hand gestures.

Hinkely et al. showed how a simple touch sensor (which is also based on a simple capacitive sensor) can enhance existing input devices such as a mouse or a trackball [6].

Vision-based gesture recognition There have been a number of studies on using computer vision for human gesture recognition [7]. However, achieving robust and accurate gesture recognition in unconditioned environments, such as the home or office, is still difficult. The EnhancedDesk [9] uses a thermo-infrared camera mounted above the table to extract the shape of the hand from the background. In contrast to these vision-based approaches, our solution does not rely on the use of external cameras, and all the necessary sensors are embedded in the surface. As a result, our technology offers more design flexibility when we implement systems.

Other types of vision-based systems include HoloWall [13] and Motion Processor [14]. Both systems use a video camera with an optical infrared filter for recognition, and infrared lights are used to illuminate objects in front of the camera. While Motion Processor directly uses this infrared reflection, HoloWall uses a diffuser surface to eliminate the background image. "Barehand" [19] is an interaction technique for a large interactive wall. It enables recognizing hand shapes by using a sensor similar to that of HoloWall, and it uses the shapes to trigger corresponding actions. Using infrared reflection, the system can detect not only the shape of the hand, but also its distance from the camera. As a result, gestures that cannot be recognized by other vision-based systems, such as moving a finger vertically over a surface (i.e., tapping), can be detected. However, like other vision-based systems, these systems also require the use of external cameras and lights, and thus they cannot be integrated into a single unit.

Bimanual interfaces Various types of bimanual (two-handed) interfaces (for example, see [1, 5, 17] and [4] for physiological analysis of these interfaces) have been studied. With such an interface, the user normally holds two input devices (e.g., a trackball and a mouse), and controls two positions on the screen. For example, the user of ToolGlasses [1] controls the tool-palette location with his/her non-dominant hand, while the cursor position is controlled by the user's dominant hand. Some bimanual systems [5, 17] provide higher-degree-of-freedom control by using motion- or rotation-sensitive input devices. With the SmartSkin sensor, the user can also control more than two points at the same time, and the shape of the arm or hand can be used as input. This is another approach to achieving higher-degree-of-freedom manipulation.

In contrast to two-handed interfaces, interaction techniques that are based on the use of multiple fingers have not been well explored. DualTouch [12] uses a normal touch panel to detect the position of tow fingers. Its resistive touch panel gives the middle position between two fingers when two positions are pressed, and assuming that the position of one finger is known (i.e., fixed to the initial position), the position of the other finger can be calculated. DualTouch can perform various interaction techniques such as "tapping and dragging", but due to this assumption of the initial position, most multiple-finger interfaces described in this paper are not possible.

CONCLUSION AND DIRECTIONS FOR FUTURE WORK

Our new sensing architecture can turn a wide variety of physical surfaces into interactive surfaces. It can track the position and shape of hands and fingers, as well as measure their distance from the surface. We have developed two working interactive surface systems based on this technology: a table and a tablet, and have studied various interaction techniques for them.

This work is still at an early stage and may develop in several directions. For example, interaction using multiple fingers and shapes is a very new area of human-computer interaction, and the interaction techniques described in this paper are just a few examples. More research is needed, in particular, focusing on careful usability evaluation.

Apart from investigating different types of interaction techniques, we are also interested in the following research directions.

Using a non-flat surface as an interaction medium: Places of interaction are not limited to a tabletop. Armrests or table edges, for example, can be good places for interaction, but have not been studied well as places for input devices. Placing SmartSkin sensors on the surface of 'pet' robots, such as Sony's AIBO, is another possibility. The robot would behave more naturally when interacting with humans. Similarly, if a game pad were "aware" of how the user grasps it, the game software could infer the user's emotions from this information.

Combination with tactile feedback: Currently, a SmartSkin user can receive only visual feedback, but if SmartSkin could make the surface vibrate by using a transducer or a piezo actuator, the user could "feel" as if he/she were manipulating a real object (the combination of a touch panel and tactile feedback is also described by Fukumoto [3]). [1]

Use of transparent electrodes: A transparent SmartSkin sensor can be obtained by using Indium-Tin Oxide (ITO) or a conductive polymer. This sensor can be mounted in front of a flat panel display or on a rear-projection screen. Because most of today's flat panel displays rely on active-matrix and transparent electrodes, they can be integrated with SmartSkin electrodes. This possibility suggests that in the future, display devices that will be interactive from the beginning, and will not require "retrofitting" sensor elements into them.

We also want to make transparent tagged objects by combining transparent conductive materials with the use of capacitance tags as shown in Figure 15. This technology will enable creating interface systems such as "DataTiles" [18], a user can interact with the computer via the use of tagged physical objects and hand gestures.

Data communication between the sensor surface and other objects: Because the SmartSkin sensor uses a wave signal controlled by software, it is possible to encode this signal with data. For example, location information can be transmitted from a SmartSkin table, and a digital device such as a PDA or a cellular phone on the table can recognize this information and trigger various context-aware applications. The table could also encode and transmit a "secret key" to mobile devices on the table, and these devices can establish a secure network with this key.

ACKNOWLEDGEMENTS

We thank our colleagues at Sony Computer Science Laboratories for the initial exploration of ideas described in this paper. We also thank Shigeru Tajima for the valuable technical advice, Takaaki Ishizawa and Asako Toda for their contribution to the implementation of the prototype system. We also would like to thank Toshi Doi and Mario Tokoro for their continuing support of our research.

REFERENCES

1. Eric A. Bier, Maureen C. Stone, Ken Pier, William Buxton, and Tony DeRose. Toolglass and Magic Lenses: The see-through interface. In James T. Kajiya, ed-

[1] One interesting but unasked question is "Is it possible to provide tactile or similar feedback to a user whose hand is in the proximity of the surface, but not directly touching the surface?".

itor, *Computer Graphics (SIGGRAPH '93 Proceedings)*, volume 27, pages 73–80, August 1993.

2. George W. Fitzmaurice, Hiroshi Ishii, and William Buxton. Bricks: laying the foundations for graspable user interfaces. In *CHI'95 Conference*, pages 442–449, 1995.

3. Masaaki Fukumoto and Toshiaki Sugimura. ActiveClick: Tactile feedback for touch panels. In *CHI 2001 summary*, pages 121–122, 2001.

4. Y. Guiard. Asymmetric divisoin of labor in human skilled bimanual action: the kinematic chain as a model. *Journal of Motor Behavior*, pages 485–517, 1987.

5. Ken Hinckley, Randy Pausch, John C. Goble, and Neal F. Kassell. Passive real-world interface props for neurosurgical visualization. In *CHI'94 Proceedings*, pages 452–458, 1994.

6. Ken Hinckley and Mike Sinclair. Touch-sensing input devices. In *CHI'99 Proceedings*, pages 223–230, 1999.

7. IEEE. Proceedings of the fourth ieee international conference on automatic face and gesture recognition, 2000.

8. Hiroshi Ishii and Brygg Ullmer. Tangible Bits: Towards seamless interfaces between people, bits and atoms. In *CHI'97 Proceedings*, pages 234–241, 1997.

9. Hideki Koike, Yoichi Sato, Yoshinori Kobayashi, Hiroaki Tobita, and Motoki Kobayashi. Interactive textbook and interactive venn diagram: natural and intuitive interfaces on augmented desk system. In *CHI 2000 Proceedings*, pages 121–128, 2000.

10. S.K. Lee, William Buxton, and K. C. Smith. A multitouch three dimensional touch-sensitive tablet. In *CHI '85 Proceedings*, pages 21 – 25, 1985.

11. M. Mathews and W. Schloss. The radiodrum as a synthesis controller. In *Proceedings international computer music conference*, 1989.

12. Nobuyuki Matsushita, Yuji Ayatsuka, and Jun Rekimoto. Dual Touch: a two-handed interface for penbased PDAs. In *ACM UIST 2000 Proceedings*, pages 211–212, 2000.

13. Nobuyuki Matsushita and Jun Rekimoto. HoloWall: Designing a Finger, Hand, Body, and Object Sensitive Wall. In *Proceedings of UIST'97*, October 1997.

14. Shunichi Numazaki, Akira Morshita, Naoko Umeki, Minoru Ishikawa, and Miwako Doi. A kinetic and 3D image input device. In *Proceedings of the conference on CHI 98 summary*, pages 237–238, 1998.

15. Polhemus, Inc., Colchester, Vermont. *3SPACE ISOTRAK User's Manual*, 1987.

16. Jun Rekimoto and Masanori Saitoh. Augmented Surfaces: A spatially continuous workspace for hybrid computing environments. In *Proceedings of ACM CHI'99*, pages 378–385, May 1999.

17. Jun Rekimoto and Eduardo Sciammarella. ToolStone: Effective use of the physical manipulation vocabularies of input devices. In *Proc. of UIST 2000*, 2000.

18. Jun Rekimoto, Brygg Ullmer, and Haruo Oba. DataTiles: a modular platform for mixed physical and graphical interactions. In *CHI 2001 proceedings*, pages 269–276, 2001.

19. Meredith Ringel, Henry Berg, Yuhui Jin, and Terry Winograd. Barehands: implement-free interaction with a wall-mounted display. In *CHI 2001 summary*, pages 367–368, 2001.

20. Norbert A. Streitz, Jorg Geisler, Torsten Holmer, Shin'ichi Konomi, Christian Muller-Tomfelde and-Wolfgang Reischl, Petr Rexroth, Peter Seitz, and Ralf Steinmetz. i-LAND: An interactive landscape for creativity and innovation. In *CHI'99 Proceedings*, pages 120–127, 1999.

21. Brygg Ullmer and Hiroshi Ishii. The metaDESK: models and prototypes for tangible user interfaces. In *UIST'97 Proceedings*, pages 223–232, 1997.

22. John Underkoffler and Hiroshi Ishii. Illuminating Light: An optical design tool with a luminous-tangible interface. In *CHI'98 Proceedings*, pages 542–549, 1998.

23. Pierre Wellner. The DigitalDesk calculator: Tangible manipulation on a desk top display. In *ACM UIST'91 Proceedings*, pages 27–34, November 1991.

24. Pierre Wellner. Interacting with paper on the DigitalDesk. *Communication of the ACM*, 36(7):87–96, August 1993.

25. Thomas Zimmerman. Personal area networks: Nearfield intrabody communication. *IBM Systems Journal*, 35(3-4):609–617, 1996.

26. Thomas G. Zimmerman, Joshua R. Smith, Joseph A. Paradiso, David Allport, and Neil Gershenfeld. Applying electric field sensing to human-computer interfaces. In *CHI'85 Proceedings*, pages 280–287, 1995.