

Chapter 7

Augmenting Interactive Tabletops with Translucent Tangible Controls

Malte Weiss, James D. Hollan, and Jan Borchers

Abstract Multi-touch surfaces enable multi-hand and multi-person direct manipulation interfaces. However, they lack haptic feedback. Tangible interfaces (TUIs) are a promising approach to this issue but most are special-purpose with fixed physical and visual appearance. This chapter provides an overview of recent work to add haptic feedback to interactive surfaces, including haptic and tactile displays, tangibles on tabletops, general-purpose controls, and typing on multi-touch tables. The focus of the chapter is Silicone Illuminated Active Peripherals (SLAP). SLAP Widgets are physical translucent general-purpose controls, such as buttons, knob, sliders, and keyboards, that can be used to manipulate virtual objects on interactive tabletop surfaces. They combine benefits of physical and virtual controls, providing the strong affordances and haptic feedback of physical controls and enabling the dynamically changeable appearance possibilities of virtual controls. SLAP Widgets are particularly promising for tasks in which eyes-free manipulation is advantageous and their plasticity encourages development of context sensitive controls and exploration of alternative interface forms.

Introduction

Interactive multi-touch surfaces have recently emerged as an interesting extension to the established direct manipulation graphical desktop metaphor. Direct manipulation [1, 2] provides a natural way of interacting with graphical user interfaces (GUIs). On interactive multi-touch surfaces, objects can be manipulated by directly touching and dragging them, allowing interaction without the indirection of keyboard or mouse. Furthermore, while traditional graphical interfaces enable an individual user to manipulate dynamic digital data, interactive tables facilitate collocated collaborative work, allowing multiple people to interact simultaneously

M. Weiss (✉)

Media Computing Group, RWTH Aachen University, 52074 Aachen, Germany
e-mail: weiss@cs.rwth-aachen.de

with the same computer. Tables are a common and familiar meeting space for all types of conversational exchanges among small groups of people. Interactive tabletop surfaces are especially well suited for presenting shared visual information without designating one person as the owner of the information. The horizontal surface of a table affords spreading, sharing, and manipulating a variety of materials. Tables provide a common working environment and direct manipulation ensures that users are aware of each other's operations. In terms of Ubiquitous Computing [3], multi-touch tabletops move users away from desktop computers to interactive systems that hide technology and accentuate direct natural interaction.

As interactive tabletops become more widely available and users have more experience with multi-touch applications on smart phones and other devices there is increasing motivation to design tabletop applications that can assist users with common everyday tasks. However, the direct transfer of conventional desktop applications to multi-touch tables is problematic. This is due to the fact that neither operating systems nor applications were designed with an expectation of multiple simultaneous inputs or gestural interaction. In addition, virtual controls, such as on-screen buttons, suffer from absence of haptic feedback, requiring visual attention during operation. Without visual monitoring input problems can result because of inability to feel a virtual control's boundary or current state. This is particularly troublesome for typing. Although most applications require typing, it is inconvenient and error-prone on virtual keyboards. In addition, the size of a finger in comparison to a mouse cursor can make precise interaction difficult and cause occlusion problems when operating small controls [4].

Since their introduction in 1997, Tangible User Interfaces [5], or in short *tangibles*, have proven to be useful interface components that provide natural haptic feedback during interaction. They allow users to manipulate data with real physical interface objects. However, most tangibles are either restricted to a specific purpose, e.g., the composition of a music piece, or have specific physical affordances associated with particular domain objects. Thus, bringing *conventional* general physical widgets, such as buttons, sliders, knobs, etc., to tabletops is a logical next step. They provide compelling physical affordances, guide users' actions, enable tactile feedback, and can be used in an eyes-free fashion. However, unlike easily changeable graphical widgets they have a fixed visual appearance. Moreover, they are usually expensive and tethered which restricts their use and is especially troublesome for tabletop interaction.

We propose a new class of interface objects that combine properties of both graphical and physical interfaces: translucent general-purpose tangibles. Silicone Illuminated Active Peripherals (SLAP) are a first instance of this new class. They consist of translucent general-purpose physical controls, such as buttons, knobs, sliders, and keyboards, that can be used to manipulate and display the state of virtual objects. Users can place SLAP Widgets on a multi-touch table and use them to interact with virtual objects, e.g., to change the brightness of a photograph or to navigate in an audio file. Like other tangibles they provide haptic feedback to aid interaction without requiring visual attention. An image for each widget, e.g., the label of a button or the state of slider, is back-projected onto the translucent widget and is thus

visible to users. The projected image can be dynamically changed to indicate state. For example, the layout of the SLAP Keyboard can be visually altered on the fly between language specific layouts, and its keycaps can be changed to aid entering mathematical or other special symbols. SLAP Widgets do not require any electronics or tethering and can be positioned wherever needed on the tabletop. When no longer required they can be placed aside and out of the way.

In this chapter, we (1) provide an overview of recent work to add haptic feedback to interactive surfaces, (2) address issues with haptic displays and tangibles on tables, (3) discuss benefits and tradeoffs involved in combining the strong affordances and haptic feedback of physical controls with the dynamic visual appearance changes of virtual objects, (4) present SLAP widgets as a first instance of this new class of translucent general-purpose tangibles, and (5) discuss future research directions for exploring tangible controls on interactive multi-touch tables.

Background

In this section, we present an overview of current research on adding haptic feedback to multi-touch tables. We cover haptic and tactile displays, tangibles on tabletops, general-purpose controls, transparent tangibles for dynamic relabeling, and typing on touch surfaces.

Haptic and Tactile Displays

Haptic and tactile displays, as defined by Poupyrev et al., are interactive devices that simulate the haptic sensation of physical objects and textures [6]. We refer to [6, 7] for a general overview.

Pin displays (e.g., [8, 9]) employ a small 2D array of pins that rise out of the surface when actuated to create a physical texture to, for example, simulate a button and its boundaries. Shape Memory Alloys (SMAs) that can assume specific shapes at different temperatures (e.g., [10]). Harrison and Hudson [11] use pneumatics to realize deformable areas on a multi-touch display. Other approaches add tactile feedback by using vibration [12] when virtual controls are triggered, e.g., by using linear vibrotactile actuators. The technologies employed to create haptic and tactile displays currently provide only limited physical affordances and feedback. Complex controls, such as knobs or sliders, cannot yet be realized. In addition, existing approaches are expensive and not applicable for use with large surfaces.

Tangibles on Tabletops

In their seminal *Bricks* paper [13], Fitzmaurice et al. highlighted advantages of the rich affordances of physical objects and introduced the concept of *Graspable User*

Interfaces, interfaces that allow interaction with physical handles as virtual controls. Users can place small physical blocks, called *bricks*, onto virtual objects on a surface and move, rotate, and scale the associated objects by manipulating the bricks. Ishii and Ullmer extended the bricks concept in their pioneering work on Tangible Bits. Inspired by earlier notions of Ubiquitous Computing [3] and the affordances of physical artifacts [13], they introduced *Tangible User Interfaces (TUIs)* [5, 14]. These interfaces give physical form to digital data [15], exploiting users' haptic abilities to assist manipulation of digital information.

Since tangibles expose strong physical affordances, they have been used in many tabletop applications to enhance interaction metaphors and improve haptic feedback. *Illuminating Light* [16] was an early system that explored use of tangibles on an interactive surface. In one sample tabletop application, users could place tangibles that represented specific optical artifacts, e.g., laser sources, mirrors, and beam splitters, onto a rear-projected tabletop display. Virtual beams, simulated and projected onto the tabletop, radiated from the laser sources and appeared to be reflected or refracted by the tangible elements placed in their paths on the surface. By adding tangibles representing mirrors, beam splitters, and lenses, users could simulate various optical phenomena (Fig. 7.1a). Due to the constant visual feedback, the system created the illusion that the input channel (placing and manipulating physical objects) and output channel (the display of the optical simulation) were the same, providing what has been termed *inter-referential input/output* [1, 2].

In a succeeding project called *Urp* [17], the same authors created a prototype to support architects in planning and designing urban areas. By placing models of buildings on a workbench, physical effects such as winds and shadows were simulated and could change depending on the time of day and placement of models (Fig. 7.1b). In addition to facilities similar to those in *Illuminating Light*, *Urp* provided tangibles to change the digital properties of other tangibles, e.g., a wand that switched the opaque facade of a building to glass that in turn resulted in changes to shadow simulation. Tangible tools were also introduced to, for example, measure distances between buildings.

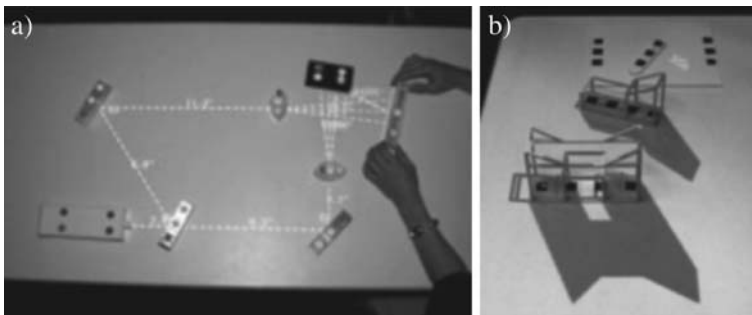


Fig. 7.1 Tangible interfaces on tabletops. (a) *Illuminating Light*. (b) *Urp*. Courtesy of Underkoffler and Ishii [16, 17]

Fig. 7.2 The *reacTable*.
Courtesy of Jordà et al. [18]



In the above projects, tangibles are geometric representations of their real-world counterparts. This limits the scalability and complexity of these systems. The *reacTable* by Jordà et al. [18] confronts these issues by using acrylic square plates for musical creation. Users are provided with six types of tangibles, including sound generators, filters, and audio mixers, that can be linked to each other (Fig. 7.2). Complex sounds are created by putting the tangibles close to each other. In allusion to real sound mixers, properties such as frequency and speed can be set by turning the specific tangibles. Volume is changed by finger dragging. The table's back projection gives visual feedback about the state and the connectivity of the tangibles.

The idea of synthesizing complex data with tangibles has more recently been explored as a form of tangible programming. In [19], Horn et al. present a tangible version of the *Scratch* programming language that aims to help children learn a programming language by putting building blocks together. They found that tangibles and virtual blocks were equally well understood but tangibles were more motivating and led to increased probability of group participation.

Actuated Tangibles

One inherent problem of all the aforementioned tangible systems is that data can only be manipulated by the user. The system itself cannot change a physical value, such as the position of a building in *Urp*. It can only reflect the consequences of users' changes and is therefore one-directional. In response to this, actuated tangibles have been developed. We refer to [20] for a detail overview of actuated tangibles.

The *Planar Manipulator Display* by Rosenfeld et al. [21] uses mobile wireless robots which can be freely positioned on a workbench by both the user and the software. The authors propose several applications, including an interior architecture planner: a user can move a piece of furniture to a desired position on the tabletop. Accordingly, all other pieces arrange themselves according to certain constraints, such as moving furniture away from windows to provide most light in the room.

In [22], Pangaro et al. presented the *Actuated Workbench*, which uses an array of electromagnets to freely position tangible "pucks" on an interactive surface. They

authors highlight several potential applications, such as undo of tangible operations, remote control of tangibles, and teaching. In a later paper [20], Patten and Ishii enrich the interaction by adding mechanical constraints, like rubber bands or collars. They present an application that automatically arranges telephone towers on a map, which are represented by tangible pucks. The algorithm ensures an efficient distribution of towers, however, by adding mechanical constraints to the tangibles the user can manually override decisions of the underlying algorithm to correct minor mistakes. For example, an oval-shaped ring around two pucks enforces a maximum distance between two telephone towers.

General-Purpose Controls

Most tangible interfaces are special-purpose, and their generality beyond a particular application domain is limited. Conventional controls, such as buttons, sliders, knobs, and keypad, are not only general but have strong physical affordances and well-known natural mappings. In this section, we review work on providing general-purpose controls to improve interfaces for interactive surfaces.

Block et al. developed VoodooSketch [23] (Fig. 7.3a), a system that allows users to design custom interactive palettes to complement multi-touch surfaces. Users can plug real physical controls (e.g., buttons, sliders, knobs, etc.) into the palette and edit parameters of objects on the surface, e.g., the thickness of a drawn line. Functions are mapped to controls by simply drawing labels next to them. For example, the word “opacity” written next to a slider enables users to set the opacity of an object on the surface by dragging the slider. Furthermore, a user can sketch controls using a special pen. For example, a drawn rectangle with the label “Save file” next to it acts as save button and is triggered when the user touches it with the pen. The interactive palette is based on VoodooIO, a flexible substrate material with embedded conductive layers [24] that identifies and monitors widgets when pushed into the surface. A paper with an imprinted dot pattern is bonded on top of this layer.

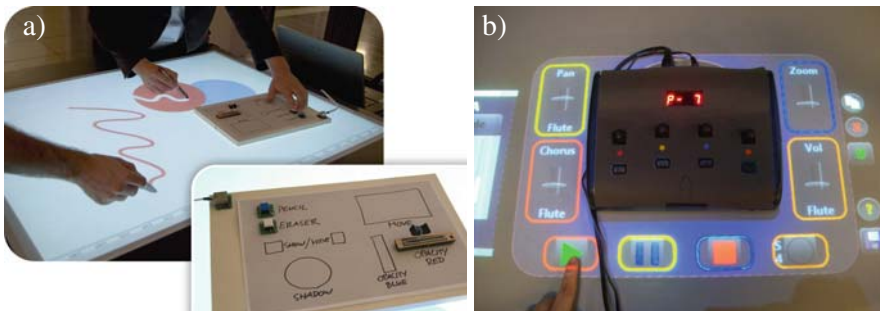


Fig. 7.3 General-purpose controls for tabletops. (a) VoodooSketch represents flexible interactive palettes. Courtesy of Block et al. [23]. (b) Portable device with physical controls provides haptic feedback when manipulating virtual objects. Courtesy of Fiebrink et al. [4]

A special digital Anoto¹ pen captures the drawing of controls and their labels by reading the dot-pattern printed on the paper. In a subsequent study [25], the authors demonstrated that handwritten labels are easy to learn, effective, and more efficient for assigning functions to widgets than conventional approaches, such as selecting a function from a pop-up menu. Physical components can just be removed when not needed anymore whereas drawn widgets remain on the paper. The latter does provide a natural way of saving a palette by just keeping the sheet of paper. A user can then place the interactive palettes anywhere on the surface. However, the palettes consume real estate on the interactive surface and they are tethered which limits mobility and interaction.

Fiebrink et al. point out that interaction with virtual controls lacks precision and propose the integration of physical control devices for tabletop environments [4]. They developed an audio mixer application that allows editing different tracks of a musical piece using a physical device containing four buttons and four knobs. As in *VoodooSketch*, users can dynamically map functions to the controls (Fig. 7.3b). Controls are surrounded by a visual interface, the “aura”, that exposes the function mappings of the controls and represents virtual counter-parts to the physical controls. Accordingly, values such as the volume or speed of a track can be set using both modalities, direct touch and physical controls, providing users with choices that can be matched with particular tasks. In order to map a function to a control, the user first touches a “Copy” icon in the aura. Changeable values in the mixer application are then highlighted and can be moved to the “clipboard” of the device by touching them. By touching the value in the clipboard and selecting a control in the aura the function is finally mapped to it. Due to this serialization, mappings cannot be performed in parallel. The authors comment that this might be desirable if group awareness and communication are critical. The devices allow saving and loading of mappings. In their studies, Fiebrink et al. found that users prefer physical devices when setting continuous values that require high precision while direct touch interaction is used for discrete values. The devices are tethered and opaque.

Similar to conventional tangibles, a major drawback of conventional electronic controls like sliders and knobs is that their physical state is fixed and decoupled from subsequent internal changes to the associated virtual object. For instance, a slider initially mapped to a virtual object does not change as the system changes the virtual object’s state. One solution is to employ motorized controls such as the Force Feedback Slider presented by Shahrokni et al. [26] and Gabriel et al. [27]. When first mapped to a virtual object, the physical position of the slider could be set to the current value of the object by using motors. Then a new value could be set by operating the physical control manually. However, motors require additional space on the device (and therewith on the table) and such controls are typically expensive and challenging to manufacture.

¹<http://www.anoto.com/>

Transparent Tangibles for Dynamic Relabeling

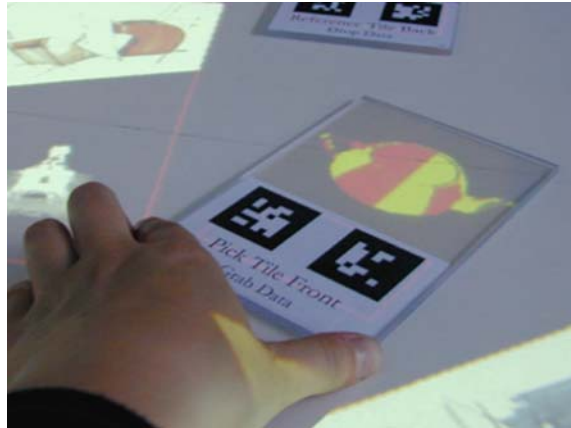
Although general-purpose tangibles provide haptic feedback for tabletop applications, they share the drawback that the visual appearances of the physical controls are fixed. They are opaque and additional graphics around the controls are required to denote their state. A top projection of current state onto the controls is one approach. However, this requires additional hardware, sensing of positions, and the projection will be on a user's hand when the tangible is manipulated. This can break the tight perceptual coupling of physical and virtual state [15]. In this section, we present projects that use transparent tangibles and rear-projection to provide dynamic relabeling while maintaining perceptual coupling.

Schmalstieg et al. enrich a virtual environment with transparent controls called *props* [28]. A table displays a virtual environment providing a stereoscopy view and user-centered projection. The user is equipped with two tracked tangibles, a transparent Plexiglas pad (about 20 × 25 cm) in the non-dominant hand and a pen in the form of a plastic tube in the dominant hand. The pad displays graphics depending on the task: it can represent a graphical tool palette, a see-through window to show different layers of a rendered landscape, or a volume tool to select objects in 3D space. Even though the pad is held above the table, the actual graphics are rendered on the tabletop using the table's projector, by tracking the user's head and the pad. This creates the illusion that the pad renders the graphics while keeping the tangible lightweight and low-cost.

DataTiles [29] combine the advantages of graphical and physical interfaces by providing transparent, acrylic tiles that can be placed on a *tray*, a flat-panel display enhanced with sensors. Tiles can, for example, represent applications (e.g., weather forecast), portals (e.g., to show webcam streams or printer status), and parameter controls (e.g., to navigate through the video on another tile). Tiles are automatically activated when placed on a grid on the panel and can be composed by putting them next to each other. Similar to props [28], each tile relies on back-projection when placed on the table. In addition, tiles may also expose printed high-resolution content which is then combined with the projected graphics. Users can manipulate a tile's content by using a pen and some tiles contain grooves to guide the user's motion. For example, a parameter tile with a circular groove can be used to navigate through the video on a tile next to it. The tangibles are detected using RFID technology and pen position is sensed by a pen table behind the display.

Tangible Tiles by Waldner et al. [30] extends these interaction ideas. In contrast to *DataTiles*, transparent tiles are visually tracked using tags that allow them to be freely positioned and orientated on the interactive surface (Fig. 7.4). Virtual objects, such as images, are shown on the table and can be manipulated by placing and moving tiles on the table. The authors provide two kinds of tiles: *container tiles* are used to move, copy and reference objects on the table and *function tiles* allow manipulating objects, e.g., to delete or magnify them. Each tile is labeled with its specific function. Although both *DataTiles* and *Tangible Tiles* represent general-purpose tangibles that can be relabeled dynamically, the tile concept, as Waldner et al. point out, provides only limited physical affordances.

Fig. 7.4 Tangible tiles.
 Courtesy of Waldner et al.
 [30]



Typing on Touch Surfaces

Typing is one of the most frequent input methods for desktop computers. As multi-touch surfaces find their way into everyday applications, the issue of typing text becomes increasingly crucial. Many researchers have explored typing on small touch displays but only recently have studies started to examine typing on large interactive surfaces. Hinrichs et al. [31] were one of the first to examine text-entry for tabletop displays. They evaluated devices according to visual appearance and performance factors such as their space requirements, rotatability, interference with direct-touch interaction and mobility, and ability to support multi-user interaction. They compared external text-entry methods like physical keyboards and speech recognition with on-screen methods such as handwriting.

While physical keyboards are a highly efficient and optimized text entry method for desktop applications, they are less appropriate for large interactive surface since they require users to switch between direct-touch and typing on a separate external keyboard. Furthermore, they require considerable space, cannot be moved and stored easily, typically are tethered, and, with few exceptions like the Optimus Maximus keyboard,² have a fixed keyboard layout. Mobile physical keyboards can be used for text-entry on tabletops but they have many of the same difficulties of other physical keyboards and still require switching between interacting with the surface and the keyboard. Speech recognition allows hands-free input but is error prone and for most users slower than typing. Furthermore, speech input can be disruptive when multiple users are present.

On-screen keyboards on mobile devices have been extensively explored and optimized. They can be dynamically relabeled and displayed where needed. However, lack of haptic feedback results in typing errors, “a general sense of uncertainty when

²<http://www.artlebedev.com/everything/optimus/>

typing” [32], and can be a problem for touch typists who rely on the sense of touch to guide text input. In addition, they require visual attention. Handwriting on tabletops is a potential alternative for text input. It is a mobile approach, only requiring a pen or stylus, and supports multi-user interaction. However, it is considered a slow input technique, and accurate handwriting recognition remains a challenging research problem. Gestural alphabets increase speed and accuracy of handwriting on touch-sensitive surfaces. They have similar advantages to handwriting but involve the cost of increased time and effort to learn as well as recognition challenges.

Hinrichs et al. conclude that there is no “perfect” text input method for interactive tabletops and the selection of an appropriate method depends on the specific task. Further empirical studies, especially on tabletops, need to be conducted. We expect text input to remain an important area of research.

SLAP Widgets

In this section, we introduce SLAP Widgets, transparent general-purpose widgets that can be used to manipulate virtual objects on interactive tabletops. Our current widget set contains keyboards, keypads, knobs, and sliders. They not only provide tactile feedback but their visual appearance can be dynamically altered. At first, we introduce the multi-touch infrastructure and the SLAP widget set. We then describe the gesture-based pairing mechanism to link SLAP widgets to virtual objects. We conclude with usage scenarios and user studies that compare performance of SLAP widgets with on-screen virtual controls. For more details about the design of SLAP Widgets also see our paper [33].

Multi-touch Table

A multi-touch table as shown in Fig. 7.5 provides the basic infrastructure for sensing physical SLAP Widgets as well as for displaying application graphics such as virtual objects (e.g., photographs, movies, text documents) that users can interact with and modify. The tabletop consists of three layers: an acrylic panel, a layer of foamed silicone film, and a diffuse matte. Our system uses a combination of two infrared-based sensing technologies. IR light is projected into the edges of the acrylic panel and changes in surface pressure are detected by an IR camera positioned below the table surface as users touch the table at various points. This sensing technology is known as FTIR (Frustrated Total Internal Reflection) [34]. Additional infrared LEDs under the table provide Diffuse Illumination (DI) as explained in [35]. The combination of FTIR and DI sensing technologies leads to robust detection of contact pressures from fingertips as well as placement of physical objects on the tabletop. For the later, we employ DI to sense markers on objects placed on the tabletop. FTIR is used for the detection of regular touches, keystrokes on the keyboard, and interactions with other widgets. A short-throw projector beneath the table renders the graphics onto the diffusor.

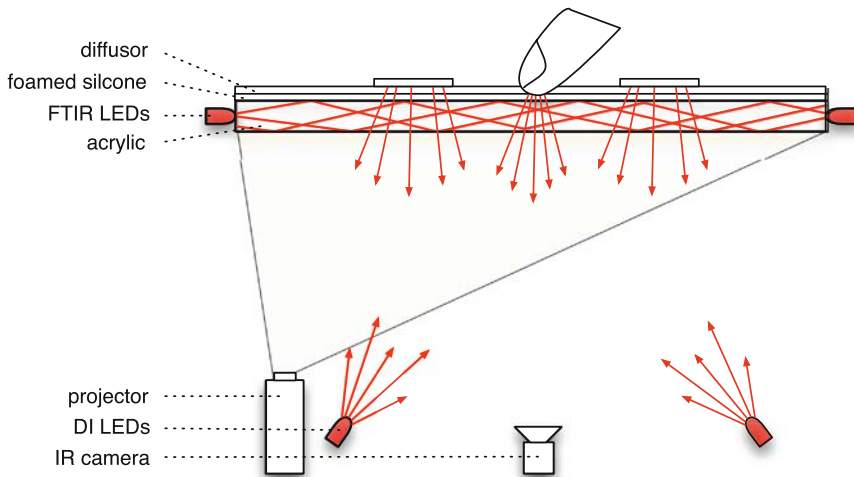


Fig. 7.5 Our multi-touch table combines FTIR and DI to detect both, finger touches and lightweight objects

Widget Set

As shown in Fig. 7.6, all widgets are constructed from transparent acrylic and silicone. This enables the underlying graphics to shine through (Figs. 7.8 and 7.9). Reflective markers of foam and paper mounted beneath each widget create identifying “footprints”, which are placed to minimize occlusion of the graphics. As illustrated in Fig. 7.7a, the arrangement of the markers in each footprint determines the widget’s type, provides a unique id, and indicates its current state (e.g., the rotation angle of the knob). Figure 7.7b shows example widget footprints as seen by the table’s camera. SLAP Widgets are registered by the distinctive arrangement of reflectors and the projected visual representations of widgets are aligned with these reflectors. Touches and moving parts such as the slider’s handle (I) and the knob’s arm (II) are tracked to update the widget state.

Keyboard – A keyboard is arguably the most necessary computer input device. The SLAP Keyboard adopts the dynamic relabeling advantage of virtual keyboards but unlike virtual keyboards its tangible surface and keys provide haptic feedback. It can be positioned anywhere on the surface and after pairing with an application can be used to enter text as if using a traditional keyboard. In addition, the keyboard layout can be modified on the fly, e.g., in order to show shortcuts (Fig. 7.8) or language-specific layouts.

The SLAP Keyboard is based on a flexible iSkin³ silicone keyboard cover (Fig. 7.6a). It is mobile and easy to collapse (see requirements in [31]). PVC caps glued onto each key and two rigid strips cemented on the edges of the keyboard

³<http://www.iskin.com>

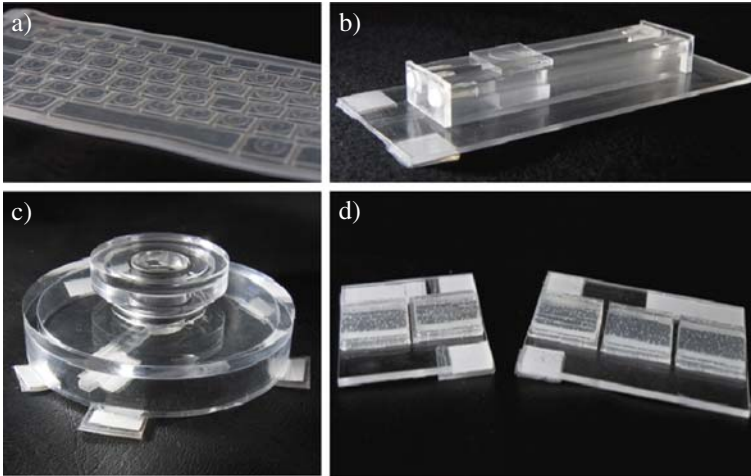


Fig. 7.6 The SLAP widget set. (a) Keyboard. (b) Slider. (c) Knob. (d) Keypads

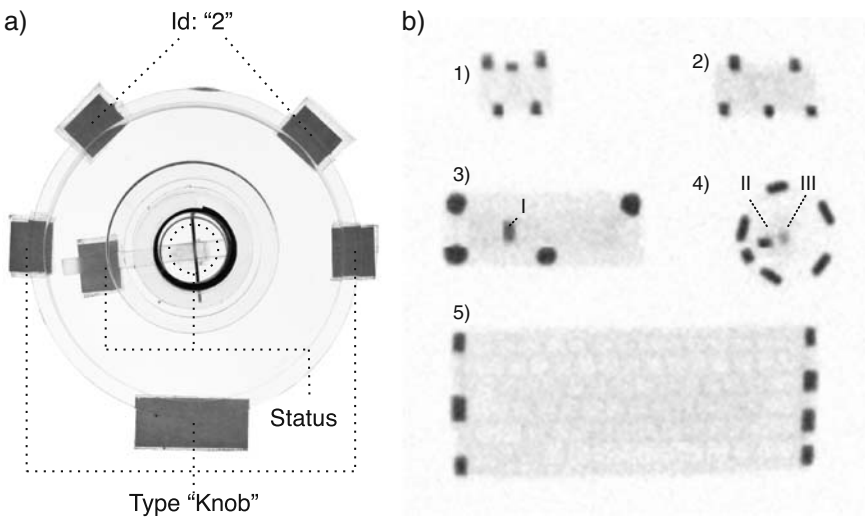


Fig. 7.7 Footprints of SLAP widgets. (a) Knob footprint. The arrangement of markers encodes type, id, and status (rotation angle, press/release state). (b) Footprints as recorded by camera (inverted for better perception). 1–2) Keypad with two and three buttons. 3) Slider with sliding knob (I). 4) Knob with angle indicator (II) and push indicator underneath the rotation axis (III). 5) Keyboard

increase tactile feedback and structural stability. Keycap labels and graphics are dynamically registered as the location of the SLAP Keyboard is tracked. Fingertip forces are conveyed directly through the keys onto the multi-touch surface making use of the FTIR effect, detected as blobs in particular key regions, and interpreted

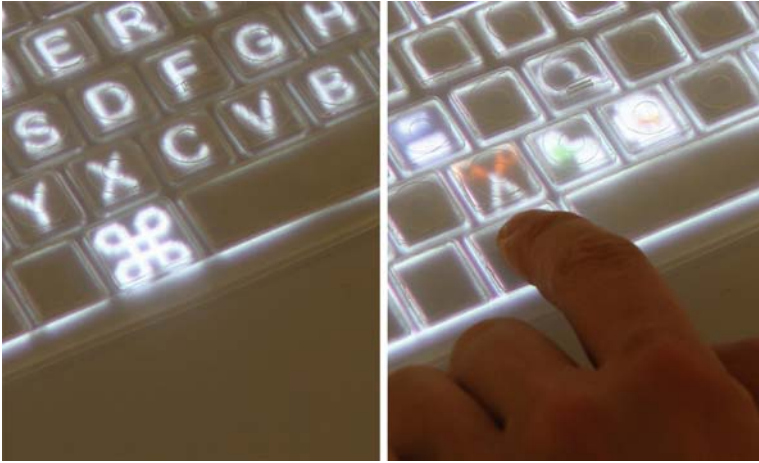


Fig. 7.8 Dynamic relabeling of SLAP keyboard



Fig. 7.9 SLAP Knob user interface. a) Selecting image property from menu. b) Setting continuous value. c) Relative navigation for frame stepping in videos

as keystrokes. The IR camera in the table provides a high frame-rate for reliable sensing of key presses.

Keypads – Some applications frequently do not require a full keyboard. For example, a video player may need only a few buttons for playing/pausing, rewinding, and fast-forwarding. Fewer buttons are easier to locate than arbitrarily assigned keys on a full keyboard. For these situations, we designed the SLAP Keypad. We have built keypads with two and three keys. A typical application for a three-keypad would be the video navigation we just mentioned.

A keypad’s base is rigid, and only the actual buttons are made of silicone (Fig. 7.6d). At 20×15 mm, its keys are also much larger than those of the SLAP keyboard. Otherwise it is similar; fingertip force is conveyed directly and labels/graphics are displayed dynamically. Multiple two- and three-button widgets can be aggregated to create larger tool palettes.

Knob – The SLAP Knob physically enables turning and pushing. These two simple functions are mapped onto different virtual representations depending on the virtual object to which it is linked. The acrylic knob rotates on a clear acrylic base

(Fig. 7.6c). It is vertically spring loaded and can be pressed as a button. An internal reflector arm orbits the axis and indicates an angular position to the camera. A transparent silicone tip in the center of the widget exposes the push state of the knob to the camera: When released, the center is invisible in the camera image. When pushed down, the tip touches the tabletop and causes a FTIR spot in the center, which is detected by the camera (Fig. 7.7b-4). When paired to time-based media, e.g., a video or audio object, the knob can be used to intuitively navigate through the video or adjust volume (Fig. 7.9b, c). However, by using the push mechanism, more complex interactions are possible. When the knob is linked to an image object, pushing it displays a properties menu. By rotating the knob, the user shuffles through a circular menu of properties (Fig. 7.9a). To select a property, such as image brightness or saturation, the user pushes the knob once again. The current value is then displayed underneath it and can be changed with a high degree of precision by turning the knob (Fig. 7.9b). A final push confirms the new value and lets the user choose another property.

Slider – Slider bars are quite common in graphical user interfaces (e.g., scroll-bars, parameter adjustment bars). A slider can be used for any interaction in which a continuous value needs to be set. For example, it could be used as a physical timeline for fast navigation in a video, or as an analog slider for setting the size of text characters. As with all SLAP widgets, the possibilities are numerous and depend solely on the virtual object. Just as the knob, the slider is made entirely of acrylic (Fig. 7.6b). Two engraved sides act as rails guiding the linear motion of the sliding knob. For stabilization the slider is mounted onto an acrylic sheet.

Pairing

Widgets must be explicitly linked to virtual objects before they can manipulate them. We refer to this as *pairing*. Inspired by Mayrhofer and Gellersen [36], we implemented a synchronous double tapping gesture: a user simultaneously taps twice next to the widget and onto the virtual object. We used this gesture to avoid recognition problems when multiple users might touch the surface at the same moment.

When first placed on a surface, a widget displays a pulsing blue halo around itself to provide feedback that it has been detected successfully. In this state the widget is not associated with any object. By performing the pairing gesture with a virtual object, an association is attempted. A green flashing halo around both objects and a connecting line between them indicate a successfully established pairing. If a virtual object refuses the association, i.e., if the widget cannot manipulate the particular object, a red halo indicates this problem.

If a previously associated widget is removed and returned to a surface, it will automatically restore its previous association. This permits collaborators to toss controls back and forth without loss of configuration. Pairings are released by repeating the synchronous double tapping gesture. Multiple widgets may be associated with a single virtual object and vice versa.

Usage Scenarios

SLAP Widgets offer versatility and ease-of-use. Having no electronic parts, they are simple, affordable, flexible, and robust. Users can literally slap a widget onto the multi-touch surface and it is immediately ready for use. Versatility is provided by pairing and dynamic relabeling. Although each widget has a specific physical structure, cast from silicone or built from acrylic, its functionality can vary significantly based upon the application used with it and the resulting dynamically changeable visual display.

SLAP Widgets can be used in any application that requires parameter changing functionality, expert shortcuts, or text entry. Since it is often desirable to have a large number of virtual objects on the touch surface but not to have a multitude of physical controls cluttering the surface, SLAP fades controls into view when they are required on a virtual surface, and lets them disappear when they are physically removed from the table. This simplifies interaction, maximizes use of display space, and decreases cognitive load. SLAP supports flexible interaction by providing a small number of controls to interact with an arbitrary number of virtual objects. The following usage scenarios are designed to communicate and emphasize the flexibility of SLAP Widgets.

Collaborative Usage Scenario – One primary advantage of multi-touch tables is to support collaborative group work. Situated around a multi-touch table, several collaborators can work together. One individual can be typing annotations with a SLAP keyboard while a second person is simultaneously interacting with other components of the application or even using another SLAP keyboard to also be entering text. This is very different from the normal situation in which one keyboard must be shared and the cable can restrict easy access. Even with only one SLAP keyboard sharing becomes a much more trivial matter. The flexible silicone keyboard can be tossed between users with no fear of damage and no cable restrictions.

Video Ethnography Scenario – Video ethnographers often need to analyze immense amounts of video data. Typically they work on desktop workstations using existing tools, such as video players and spreadsheets, to do their analysis. Multi-touch tables pose an alternative to the current ethnographic working environment, presenting users with much larger screen space, providing a collaborative space, and enabling new methods for interacting with the data.

We are developing an application for video ethnography. A major task that all ethnographers require is fine-scale video navigation. To assist navigation, we are implementing frame-by-frame navigation using the SLAP Knob. Alternatively, we anticipate using a SLAP Slider for rough navigation. For annotations related to video clips or images, the SLAP Keyboard will be used. Linked with the object, the table projects the keyboard layout, and then the user can quickly enter relevant notes or rapidly switch to another layout for easily coding specific attributes or bookmarking frames of interest. The keypad buttons can change to small thumbnails of the bookmarked frames to assist navigation and a SLAP slider can be used to browse through the bookmarked scenes.

Image Editing Scenario – Editing images represents another interesting scenario for using SLAP Widgets. The SLAP Knob provides an intuitive facility for browsing and modifying image properties. We implemented a menu to cycle through parameters like brightness, contrast, saturation, etc. (Fig. 7.9a). This eliminates the need for complicated menus and submenus that often mask useful features from the novice user. When pushing down the knob, the user can change the specific parameter (Fig. 7.9b). Pushing again returns to the menu selection. A crucial benefit of SLAP Widgets for image editing is that the user can focus visually on the image as a property is adjusted since tactile feedback removes the need to visually attend to the control.

Interface Designer Usage Scenario – Our widget framework can also serve as a toolkit for interface designers working on tabletop applications. They can take advantage of the available widgets and develop a SLAP-based facility for their work. For instance, a designer fashioning an audio mixing application may want to place sliders to represent volume and equalizer levels, knobs to represent gain and fader settings, and keypads for playback controls. In fact, designers may even choose to use SLAP Widgets on a table to cooperatively prototype a traditional application for the desktop.

User Studies

In this section, we present user studies that evaluate the SLAP Widgets. We first describe a quantitative study that compares specific SLAP Widgets with their virtual counterparts and then provide results from a qualitative study.

Knob Performance Task

In our first experiment, a video navigation and annotation task, users were asked to navigate to specific video frames and tag them. This task required users to manipulate controls while visually attending to the video. Since purely virtual controls typically require visual attention, we anticipated that the SLAP Knob would result in faster and more accurate performance because it can be manipulated in an eyes-free fashion. Specifically we hypothesized the following:

- *Hypothesis 1:* Navigation times with SLAP Widgets will be less than with virtual controls.
- *Hypothesis 2:* Navigational overshoots with SLAP Widgets will be less frequent than with virtual controls.
- *Hypothesis 3:* Task completion times with SLAP Widgets will be less than with virtual controls.

The experiment consisted of two conditions that differed only in the use of SLAP Widgets or virtual controls. All controls were placed at the same positions and orientations.

- “*SLAP*” *Condition*: All controls, two keypads and a knob, were *SLAP* Widgets with their respective rear-projections.
- “*Virtual*” *Condition*: All controls were virtual, that is, no widgets were placed on the table, but the graphics were the same as in the *SLAP* condition.

The keypad buttons were triggered by regular touches. The virtual knob used the standard method of tracking as commonly used by today’s desktop applications: When the user holds down her (index) finger in the knob area, the knob rotation follows the finger until it is released, even if dragged outside the area. Each condition involved four trials, and each trial consisted of three instances of navigating to a frame in a video and marking it. Eight video clips were randomly sequenced for each participant; four for the first condition and four for the second. Each participant was randomly assigned to a condition.

Volunteer participants were recruited from a university campus using a general posting in a cafeteria and at a presentation on multi-touch technology. A total of 21 volunteers participated, 19 male and 2 female, between the ages of 22 and 36 with an average age of 25.7. Three were left-handed, 18 right-handed, and none reported any color vision deficiency.

Participants were presented with a multi-touch table with a video window, a bookmark pad, a control pad, and a navigation knob (Fig. 7.10). Depending on the condition, *SLAP* widgets were or were not in place. The goal of finding and tagging three frames in a video clip was explained. The task was to navigate using a knob and keypad, locate tinted frames, and tag them using a bookmark keypad. Frames tinted in red were to be tagged with a red bookmark, similarly for green and blue. A host computer recorded all actions in a time-coded log file for later analysis.

Typically, a participant would press the *Play* button to start the video, press the *Stop* button when a tinted frame was noticed, navigate frame by frame using the navigation knob until the exact tinted frame was displayed, press a bookmark button to tag it, and press *Play* to continue searching for any remaining tinted frames.

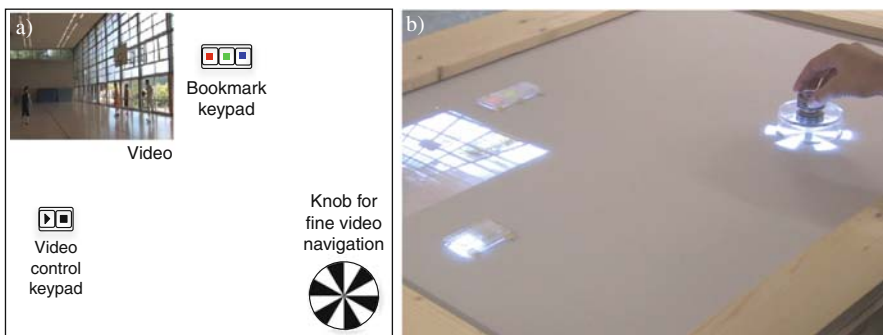


Fig. 7.10 Quantitative test setup. (a) Tabletop layout. (b) Fine-navigation using the *SLAP* Knob

Video navigation to specific target frames was significantly faster using the SLAP Knob compared to virtual graphics, and it also resulted in fewer overshoots. Moreover, it took participants less time to complete a task using SLAP Widgets than with their virtual counterparts. The study reveals that navigation using virtual knobs required more time and produced more overshoots of the target keyframe compared to the SLAP Knob. We believe the reason for this difference to be that the virtual knob lacks tactile feedback and thus requires visual attention. Participants needed to look to position their fingers at the virtual knob and when their finger drifted away from the central point, the irregular scrolling speed of the video that resulted forced participants to correct their finger position. The SLAP Knob instead was grabbed and turned mostly without visual attention, leading to fewer overshoots and shorter interaction times.

Qualitative Evaluation

Are SLAP Widgets easy to associate and manipulate? What do people like, dislike, or want to change about them? These are questions we addressed in a set of tasks designed to familiarize participants with SLAP Widgets.

Participants were presented with a multi-touch table displaying a video window, an image window, and a text field. The SLAP Widgets were described in a 5-min demonstration of their use including synchronous pairing gestures. Participants were requested to perform the following series of control, navigation, and editing tasks followed by an interview to provide feedback. The tasks and interview were recorded and reviewed.

- *Video Control:* Place a keypad widget on the table, associate it with the video window, and control the video using Play and Pause buttons of the keypad widget.
- *Video Navigation:* Place a SLAP Slider and SLAP Knob on the table, associate them with the video window, scroll through the video using the slider for gross navigation and the knob for fine steps between frames.
- *Image Editing:* Re-associate the SLAP Slider and SLAP Knob to the image window, adjust brightness with the slider and saturation with the knob.
- *Text Editing:* Place a SLAP Keyboard on the table, associate it with the text field, type your name, re-associate the knob to the text field, and modify text color with the knob.

All participants were expert computer users experienced with graphical user interfaces and recruited from a university campus. Seven male and three female participants, between ages of 21 and 28, volunteered to participate and consented to video recording.

Most (9/10) participants declared manipulating the SLAP Widgets was intuitive and self-evident. One participant emphasized that widgets map well-known physical control elements to their virtual equivalents and may be particularly well adapted for people not familiar with virtual controls. Another participant commented on how the

widgets permitted resting her hands on them while not using them (something not possible with virtual keyboards and controls).

The pairing gesture was immediately understood by all participants and used readily. Comments indicated that it felt similar to setting a foreground GUI window. Some (4/10) participants suggested alternative pairing gestures such as placing a widget on a virtual object and sliding it to a comfortable position not occluding any virtual objects (“grasp and drag” of control properties), but also felt that synchronous double-tapping was particularly appropriate for the keyboard. Some (4/10) participants felt the SLAP Widgets were too quiet and could benefit from auditory feedback, particularly the keyboard.

Feedback on the keyboard was mixed, some participants suggested improvements. It was felt that the making it easier to feel the edges and keycap contours as well as providing a more traditional tactile response would improve the keyboard. Although participants appreciated the concept of the haptic SLAP Keyboard, most still felt more comfortable typing on the virtual keyboard. This may have resulted from the fact that the DI interface at times created false positives due to hover effects and it appeared difficult for participants to know how hard they had to press the silicone keys. We plan to address both issues in future iterations of SLAP keyboard prototypes.

Future Trends

Interactive surfaces become increasingly common and available to be used in everyday applications. Future operating systems will support multi-touch interaction by default and enable more interface designers to think beyond the current concept of single cursor interaction. Tabletop applications will rapidly move from the simple proof-of-concept prototypes (e.g., photo sorting) that we see currently to practical applications. Interactive multi-touch tabletops will play an crucial role in computer supported collaborative work applications. General-purpose tangibles are particularly valuable in such applications since they not only provide haptic feedback but allow users to be aware of all actions on the table in a variety of domains.

We expect that the emerging trend of organic interfaces [37, 38] will evoke actuated deformable tabletop controls that actively reflect the current system state. Furthermore, we assume that future tangibles will not only be limited to the tabletop, but also the space above and around the table will be incorporated into the applications by making use of gestures and dynamic tangibles whose representations reach beyond the tabletop projection. Multi-touch systems using switchable diffusors, as in SecondLight [39], that allow projection onto the surface as well as on tangibles above represent a particularly promising research direction.

These new interaction modalities require enhanced technologies. Currently, vision is the only way for reliable detection of objects, patterns, and markers on tabletops but we assume that new ways for object detection, maybe similar to RFID chips as in DataTiles [29], will continue to be explored. In terms of transparent tangibles, it is likely that visual markers for position and orientation detection will be

completely hidden in the future, as is already being investigated in recent research [40, 41]. However, the development of tabletop applications is still constrained due to technical limitations. Text input using handwriting requires a high camera resolution, fast typing with tangible keyboards, such as the SLAP keyboard, demands a high camera frame rate. Collaborative tabletop interaction naturally takes place on large surfaces, but this requires a high display resolution or multiple synchronized displays. An interdisciplinary development, including Human-Computer Interaction, Computer Science, and Electrical Engineering, will be essential to face these challenges.

Conclusion

Interactive multi-touch horizontal surfaces have considerable potential to become a common part of everyday computing applications. Tabletops provide a natural environment for collaboration and multi-touch tables allow direct manipulation of digital data while supporting the awareness of other users at the table. They are likely to become a crucial part of the overall interactive computing ecology and as with other technologies will need to be carefully integrated into that increasing complex of mobile and distributed ecology.

References

1. Hutchins EL, Hollan JD, Norman DA (1985) Direct manipulation interfaces. *Human-Computer Interaction* 1(4):311–338
2. Hutchins EL, Hollan JD, Norman DA (1986) Direct manipulation interfaces. In: *User centered system design: New perspectives on human-computer interaction*, Lawrence Erlbaum Associates, Hillsdale, NJ and London, pp 87–124
3. Weiser M (1991) The computer for the 21st century. *Scientific American* 265(3):94–104
4. Fiebrink R, Morris D, Morris MR (2009) Dynamic mapping of physical controls for tabletop groupware. In: *CHI '09: Proceedings of the 27th international conference on human factors in computing systems*, ACM Press, New York, pp 471–480
5. Ishii H, Ullmer B (1997) Tangible bits: Towards seamless interfaces between people, bits and atoms. In: *CHI '97: Proceedings of the SIGCHI conference on human factors in computing systems*, ACM Press, New York. pp 234–241
6. Poupyrev I, Nashida T, Okabe M (2007) Actuation and tangible user interfaces: The Vaucanson Duck, Robots, and shape displays. In: *TEI '07: Proceedings of the 1st international conference on tangible and embedded interaction*, ACM Press, New York, pp 205–212
7. Benali-Khoudja M, Hafez M, Alex J, Kheddar A (2004) Tactile interfaces: A state-of-the-art survey. In: *Proceedings of the international symposium on robotics*, Paris, France, pp 721–726
8. Craig IS, Chanter CM, Southall AL, Brady AC (2001) Results from a tactile array on the fingertip. In: *Eurohaptics*, Birmingham, UK, pp 26–28
9. Yang GH, Kyung KU, Srinivasan MA, Kwon DS (2006) Quantitative tactile display device with pin-array type tactile feedback and thermal feedback. In: *Proceedings of the IEEE international conference on robotics and automation*, Orlando, Florida, USA, pp 3917–3922
10. Oelho M, Maes P (2009) Shutters: A permeable surface for environmental control and communication. In: *TEI '09: Proceedings of the 3rd international conference on tangible and embedded interaction*, ACM Press, New York, pp 13–18

11. Harrison C, Hudson SE (2009) Providing dynamically changeable physical buttons on a visual display. In: CHI '09: Proceedings of the 27th international conference on human factors in computing systems, ACM Press, New York, pp 299–308
12. Hoggan E, Brewster SA, Johnston J (2008) Investigating the effectiveness of tactile feedback for mobile touchscreens. In: CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on human factors in computing systems, ACM Press, New York, pp 1573–1582
13. Fitzmaurice GW, Ishii H, Buxton WAS (1995) Bricks: Laying the foundations for graspable user interfaces. In CHI '95: Proceedings of the SIGCHI conference on human factors in computing systems, ACM Press/Addison-Wesley Publishing Co, New York, pp 442–449
14. Ishii H (2008) The tangible user interface and its evolution. *Communications of the ACM* 51(6):32–36
15. Ishii H (2008) Tangible bits: Beyond pixels. In: TEI '08: Proceedings of the 2nd international conference on tangible and embedded interaction, ACM Press, New York. pp xv–xxv
16. Underkoffler J, Ishii H (1998) Illuminating light: An optical design tool with a luminous-tangible interface. In: CHI '98: Proceedings of the SIGCHI conference on human factors in computing systems, ACM/Addison-Wesley Publishing Co., New York, pp 542–549
17. Underkoffler J, Ishii H (1999) Urp: A luminous-tangible workbench for urban planning and design. In: CHI '99: Proceedings of the SIGCHI conference on human factors in computing systems, ACM Press, New York, pp 386–393
18. Jordà S, Geiger G, Alonso M, Kaltenbrunner M (2007) The reacTable: Exploring the synergy between live music performance and tabletop tangible interfaces. In: TEI '07: Proceedings of the 1st international conference on tangible and embedded interaction, ACM Press, New York, pp 139–146
19. Horn MS, Solovey ET, Crouser RJ, Jacob RJK (2009) Comparing the use of tangible and graphical programming languages for informal science education. In: CHI '09: Proceedings of the 27th international conference on human factors in computing systems, ACM Press, New York, pp 975–984
20. Patten J, Ishii H (2007) Mechanical constraints as computational constraints in tabletop tangible interfaces. In: CHI '07: Proceedings of the SIGCHI conference on human factors in computing systems, ACM Press, New York, pp 809–818
21. Rosenfeld D, Zawadzki M, Sudol J, Perlin K (2004) Physical objects as bidirectional user interface elements. *Computer Graphics and Applications*. IEEE 24(1):pp 44–49
22. Pangaro G, Maynes-Aminzade D, Ishii H (2002) The actuated workbench: Computer-controlled actuation in tabletop tangible interfaces. In: UIST '02: Proceedings of the 15th annual ACM symposium on user interface software and technology, ACM Press, New York, pp 181–190
23. Block F, Haller M, Gellersen H, Gutwin C, Billinghurst M (2008) VoodooSketch: Extending interactive surfaces with adaptable interface palettes. In: TEI '08: Proceedings of the 2nd international conference on tangible and embedded interaction, ACM Press, New York, pp 55–58
24. Villar N, Gellersen H (2007) A malleable control structure for softwired user interfaces. In: TEI '07: Proceedings of the 1st international conference on tangible and embedded interaction, ACM Press, New York, pp 49–56
25. Block F, Gutwin C, Haller M, Gellersen H, Billinghurst M (2008) Pen and paper techniques for physical customisation of tabletop interfaces. In: Horizontal interactive human computer systems, 2008. TABLETOP 2008: Proceedings of the 3rd IEEE International Workshop on 2008, Amsterdam, The Netherlands, pp 17–24
26. Shahrokni A, Jenaro J, Gustafsson T, Vinnberg A, Sandsjö J, Fjeld M (2006) One-dimensional force feedback slider: Going from an analogue to a digital platform. In: NordiCHI '06: Proceedings of the 4th Nordic conference on human-computer interaction, ACM Press, New York, pp 453–456

27. Gabriel R, Sandsjö J, Shahrokni A, Fjeld M (2008) BounceSlider: Actuated sliders for music performance and composition. In: TEI '08: Proceedings of the 2nd international conference on tangible and embedded interaction, ACM Press, New York, pp 127–130
28. Schmalstieg D, Encarnação LM, Szalavári Z (1999) Using transparent props for interaction with the virtual table. In: I3D '99: Proceedings of the 1999 symposium on interactive 3d graphics, ACM Press, New York, pp 147–153
29. Rekimoto J, Ullmer B, Oba H (2001) DataTiles: A modular platform for mixed physical and graphical interactions. In: CHI '01: Proceedings of the SIGCHI conference on human factors in computing systems, ACM, New York. pp 269–276
30. Waldner M, Hauber J, Zauner J, Haller M, Billinghamurst M (2006) Tangible tiles: Design and evaluation of a tangible user interface in a collaborative tabletop setup. In: OZCHI '06: Proceedings of the 18th Australia conference on computer-human interaction, ACM Press, New York, pp 151–158
31. Hinrichs U, Hancock M, Collins C, Carpendale S (2007) Examination of text-entry methods for tabletop displays. In: Horizontal interactive human-computer systems, 2007. TABLETOP '07: Proceedings of the 2nd annual IEEE international workshop on 2007, Newport, Rhode Island, pp 105–112
32. Hirche J, Bomark P, Bauer M, Solyga P (2008) Adaptive interface for text input on large-scale interactive surfaces. In: Horizontal interactive human-computer systems, 2008. TABLETOP '08: Proceedings of the 3rd annual IEEE international workshop on 2008, Amsterdam, Netherlands, pp 153–156
33. Weiss M, Wagner J, Jansen Y, Jennings R, Khoshabeh R, Hollan JD, Borchers J (2009) SLAP Widgets: Bridging the gap between virtual and physical controls on tabletops. In: CHI '09: Proceedings of the 27th international conference on human factors in computing systems, ACM Press, New York, pp 481–490
34. Han JY (2005) Low-cost multi-touch sensing through frustrated total internal reflection. In: UIST '05: Proceedings of the 18th annual ACM symposium on user interface software and technology, ACM Press, New York, pp 115–118
35. Matsushita N, Rekimoto J (1997) HoloWall: Designing a finger, hand, body, and object sensitive wall. In: UIST '97: Proceedings of the 10th annual ACM symposium on user interface software and technology, ACM Press, New York, pp 209–210
36. Mayrhofer R, Gellersen H (2007) Shake well before use: Authentication based on accelerometer data. In: Pervasive Computing, Springer, Heidelberg, pp 144–161
37. Co E, Pashenkov N (2008) Emerging display technologies for organic user interfaces. *Communication of the ACM* 51(6):45–47
38. Holman D, Vertegaal R (2008) Organic user interfaces: Designing computers in any way, shape, or form. In *Communication of the ACM* 51(6):48–55
39. Izadi S, Hodges S, Taylor S, Rosenfeld D, Villar N, Butler A, et al. (2008) Going beyond the display: A surface technology with an electronically switchable diffuser. In: UIST '08: Proceedings of the 21st annual ACM symposium on user interface software and technology, ACM Press, New York, pp 269–278
40. Koike H, Nishikawa W, Fukuchi K (2009) Transparent 2-D markers on an LCD tabletop system. In: CHI '09: Proceedings of the 27th international conference on human factors in computing systems, ACM Press, New York, pp 163–172
41. Kakehi Y, Hosomi T, Iida M, Naemura T, Matsushita M (2006) Transparent tabletop interface for multiple users on lumisight table. In: TABLETOP '06: Proceedings of the 1st IEEE international workshop on horizontal interactive human-computer systems, IEEE Computer Society, Washington, DC, pp 143–150