

Madgets: Actuating Widgets on Interactive Tabletops

Malte Weiss Florian Schwarz Simon Jakubowski Jan Borchers

RWTH Aachen University
52056 Aachen, Germany
{weiss, schwarz, jakubowski, borchers}@cs.rwth-aachen.de

ABSTRACT

We present a system for the actuation of tangible magnetic widgets (Madgets) on interactive tabletops. Our system combines electromagnetic actuation with fiber optic tracking to move and operate physical controls. The presented mechanism supports actuating complex tangibles that consist of multiple parts. A grid of optical fibers transmits marker positions past our actuation hardware to cameras below the table. We introduce a visual tracking algorithm that is able to detect objects and touches from the strongly sub-sampled video input of that grid. Six sample Madgets illustrate the capabilities of our approach, ranging from tangential movement and height actuation to inductive power transfer. Madgets combine the benefits of passive, untethered, and translucent tangibles with the ability to actuate them with multiple degrees of freedom.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces. - Input devices and strategies.

General terms: Algorithms, Design, Human Factors

Keywords: Actuation, tangible user interfaces, widgets, tabletop interaction, multi-touch

INTRODUCTION

Actuating tangible user interfaces allows systems to maintain consistency between physical objects and their virtual representations. This ensures a bilateral communication between the user and the system. While the user can manipulate tangibles taking advantage of their strong affordances and haptic feedback, the system can actuate these objects to react to internal or external events, and to bring them in sync with the internal application state.

In 2002, Pangaro et al. brought this concept to tabletops. Their Actuated Workbench [17] allows users to place pucks on a tabletop device; the system can move them using an array of electromagnets. However, the system is limited to translating rigid tangibles. More recent tabletop systems introduce complex tangibles, and there is an increasing de-

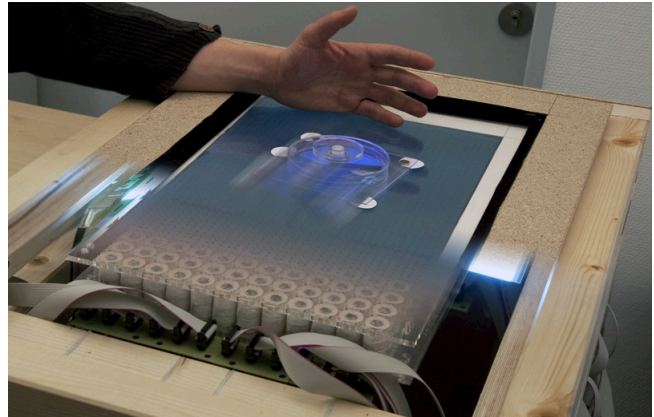


Figure 1: Madgets are composite tangible widgets that can be actuated on interactive tabletops.

mand for malleable controls [21]. SLAP Widgets [28], e.g., provide general-purpose controls for tabletops, such as keyboards, knobs, and sliders, but these cannot be actuated with existing approaches.

In this paper, we present an interactive tabletop that enables the interaction with and the actuation of complex physical controls (Figure 1). Our controls are translucent tangible general-purpose widgets, such as sliders, knobs, and buttons, with attached magnets that can be actuated using an array of electromagnets. Unlike previous electromagnetic actuation techniques beneath the tabletop [17, 18], our tangibles are not regarded as atomic units. Instead, our algorithm decomposes the widgets into rigid bodies that can be actuated independently. We explicitly model the physical properties of each control and its dynamic weight distribution subject to its current state. Thus, we can, e.g., freely move a slider widget across the table while moving its sliding knob independently. We will show that this actuation concept prepares the ground for a new class of actuated tabletop tangibles: magnetic widgets, or *Madgets*. Beside moving Madgets across the table, we make use of new actuation dimensions, such as height, force feedback, and power transfer. Furthermore, Madgets are low-cost, easy to prototype, and do not require any built-in electronics or power source for actuation or tracking.

After an overview of existing actuation techniques for interactive tabletops, we introduce the hardware setup of our table and illustrate the design of a Madget. Following that, we explain our actuation algorithm that ensures consistency be-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

UIST'10, October 3–6, 2010, New York, New York, USA.
Copyright 2010 ACM 978-1-4503-0271-5/10/10...\$10.00.

tween physical controls and their virtual counterparts. We also introduce a tracking algorithm that detects widgets even from highly subsampled input video. Finally, we explore the design space that our actuation algorithm enables and discuss future work.

RELATED WORK

Enriching tabletop displays with actuated haptics has gained much interest over the last decade. Shape displays add height as another component to the RGB output of displays [21]. Some approaches employ an array of motorized sticks that can be raised, e.g., Lumen [20], or form a deformable surface to project on from above, e.g., FEELEX [9] or Relief [15]. Harrison et al. [5] use pneumatics on inflatable areas of a multi-touch screen to create dynamic buttons. Ferrofluid liquids are another promising direction to create shape displays [7]. One drawback common to all these approaches is that they produce height maps that limit the design space to convex objects, such as buttons. Poupyrev et al. [21] propose to increase realism by adding texture information. However, height maps cannot reproduce complex controls, such as a knob or a slider. Projects that intend to create arbitrary 3D shapes, such as Claytronics [4], are still in their simulation stage and currently not feasible.

Tangible user interfaces provide rich haptic feedback and strong affordances to tabletop applications (see [8] for an overview) and, as shape displays, they can combine the benefits of physical and virtual representations. Many systems use the table's projection to enrich physical tabletop controls with dynamic visuals, e.g., DataTiles [22] or AudioPad [19]. More recently, SLAP Widgets [28] bring general-purpose controls, such as sliders, knobs, buttons, and keyboards, to tabletops, and Lumino [1] introduces stackable tangible blocks that communicate their state to the table using fiber optics. As long as tracking is ensured, there are no limitations regarding the complexity of tangibles. However, as tangibles are decoupled from the table hardware, their actuation is difficult.

The Universal Planar Manipulator (UPM) applies vibration patterns to a single horizontal plate to move physical objects [23]. The presented actuation algorithm supports arranging multiple objects at the same time. However, the movement and configuration of more complex physical controls is difficult with this approach. Furthermore, the motion induced by vibration patterns is rather slow and noisy.

The Actuated Workbench [17] uses an array of electromagnets to actuate tangible pucks. Both user and system can freely change the position of pucks. In a later project [18], Patten et al. added so-called "mechanical constraints". While the table iteratively solves a spatial optimization problem, such as the optimal distribution of telephone towers on a map, it translates the pucks that represent the intermediate results. Users can add tangible constraints, such as a rubber band around two pucks that ensures a maximum distance. The employed actuation algorithm maintains physical-virtual consistency for the position of rigid and circular tangibles. Composite tabletop tangibles that involve dimensions such as rotation, height (e.g., push buttons), or physical state (e.g., sliders and knobs), cannot be actuated with this algorithm.

Industry has successfully built motors into widgets such as sliders and knobs to actuate sound mixers and hi-fi systems. In research, this technique is used to add force feedback to tangible interaction [3, 26]. Other systems utilize robots on tabletops to ensure a bidirectional communication between user and system [14, 25]. However, motors in tangibles require batteries or tethering them, thus limiting their applicability on tabletops. They lead to a larger form factor and higher weight, technical complexity, and price, limiting the degrees of freedom when designing widgets.

DESIGN CONSIDERATIONS

Our main goal when designing Madgets was to create devices that were flexible, lightweight, and easy to build and prototype. Batteries or motors inherently introduce constraints to the widget design. Cables consume real estate on the table and limit flexibility when using the widgets, e.g., when handing over controls. Therefore, we decided to create passive devices that do not contain any electronics. This also allowed us to hide the underlying technology, following the idea of Ubiquitous Computing [27]. Additionally, we wanted our system to be able to freely position complex controls on the tabletop, as proposed by earlier systems [17, 18], but also to configure them and to give force feedback to the user.

Following the spirit of SLAP widgets [28], we wanted controls that could be relabeled on the fly. In order to maintain the illusion that the physical controls and their digital projections merge into one tangible user interface, we ruled out top projection as it causes occlusion issues. As tangible interaction and direct manipulation of virtual contents can both be used for a rich and natural interaction with tabletops, we planned to support multi-touch input as well.

HARDWARE SETUP

In the following, we describe the hardware setup of our actuated tabletop. Figure 3 illustrates all components.

Display

A 24" (52.2 cm × 29.3 cm) TFT panel (Figure 3b), detached from a Samsung SyncMaster 2494LW monitor, displays the user interface at a resolution of 1920 × 1080 pixels (93 dpi). An electroluminescent (EL) foil (Figure 3c) beneath the panel provides a uniform backlighting. We chose EL foil because it can be cut and punched, which is required for our sensing technique.

Actuation

Inspired by the Actuated Workbench, we use an array of electromagnets (Figure 3f) to actuate objects on the tabletop (Figure 2). Our matrix consists of 19 × 12 electromagnets with a diameter of 19.5 mm and a length of 34.5 mm. Each magnet is wound with 3500 turns of enameled copper wire with a diameter of 0.16 mm. The magnets are driven at 46 V DC. This induces a current of 0.32 A for a fully engaged electromagnet. Each magnet's core is an iron rod (Figure 3e) with a diameter of 8 mm.

An Arduino microcontroller board¹ triggers the magnets via 10 custom-made shields. Each shield provides up to 32 independent output channels that create pulse width modulation

¹<http://www.arduino.cc/>



Figure 2: Hardware setup. Left: Array of 228 electromagnets assembled from three modules. Middle: Cameras beneath the table capture the grid of fiber optic cables. Right: Alignment of fiber optic cables. a) Coil bobbin. b) Magnetic core. c) Fiber optic cable.

(PWM) signals. The Arduino controller also generates serial output to set the binary polarization for each magnet. A separate board provides the power and polarization flipping for each magnet.

Sensing

Since we have to detect the precise location of objects while allowing multi-touch interaction, we employ a visual tracking approach for sensing. A visual technique also does not interfere with the magnetic fields. As the electromagnets occlude most of the incoming light from above, we apply an approach similar to FiberBoard [11]. A matrix of polymer fiber optic cables (Figure 3g) represents a down-sampled multi-touch sensor. Four cables are placed between each magnetic core and its coils, and 12 around each magnet (Figure 2), yielding a total resolution of 58×37 dots (3.6 dpi). Each cable has a diameter of 0.5 mm and a length of about 45 mm. Similar to FlyEye [29], we melted the ends of each fiber optic cable using a laser cutter. This creates small lens-like pinheads that increase the numerical aperture of the cable. At the top, the cables pierce the EL foil and touch the LCD panel. Below the magnets, the grid of cables is glued to the base of the magnets.

Unlike FiberBoard, we employ Diffused Surface Illumination (DSI) instead of FTIR by mounting a 6 mm Endlighten acrylic layer (Figure 3a) on top of the LCD. A ribbon of 850 nm LEDs (Figure 3d) around the acrylic feeds infrared (IR) light into the surface. The Endlighten layer contains micro-particles that diffusively reflect the incoming IR light [6]. A finger or an object placed on the surface reflects this IR light through the LCD panel to the fiber optic cables. Although the panel attenuates the IR light, we can still extract touch events from the signal [10]. The fiber optic cables transmit the light to three Point Grey Firefly MV cameras with an attached IR filter beneath the table (Figure 2 and 3h). The cameras capture the grid of fiber optic cables at 30 fps in 640×480 and send the signal to our tracking software. The use of DSI allows us to detect objects on the surface and slightly above it. Furthermore, we do not need a compliant layer, such as silicone or latex, that could blur the visual interface.

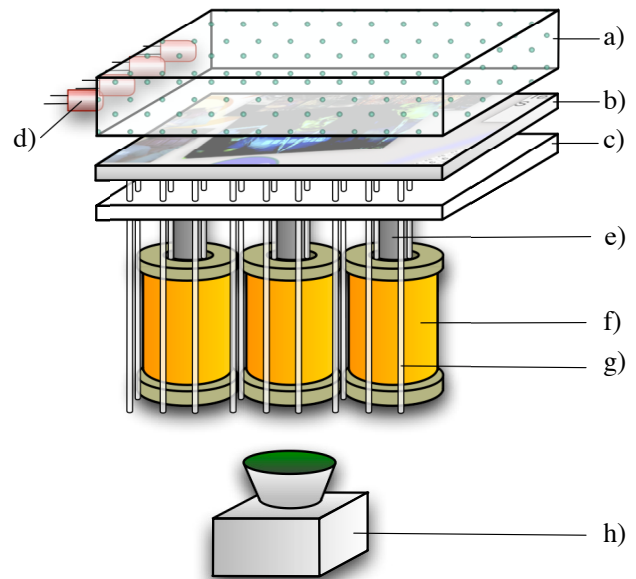


Figure 3: Hardware setup. a) Endlighten acrylic. b) TFT panel. c) EL foil. d) IR LED. e) Iron rod core. f) Electromagnet. g) Fiber optic cable. h) IR camera.

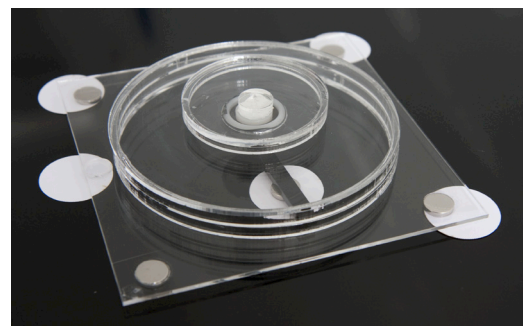


Figure 4: Knob Madget. Five paper markers communicate type, position, and state of the control. Five small permanent magnets enable actuation.

Widget design

Like SLAP Widgets [28], our controls are made of transparent acrylic. This allows us to change their appearance dynamically using the LCD screen. Each widget is mounted on a set of cylindric markers that provide reference points for tracking. The arrangement of markers yields a footprint encoding the type and unique ID of each widget. Moving parts, such as a knob's orbiting arm, are also linked to a marker that represents the control's state (Figure 4). Furthermore, a set of permanent magnets is attached to the widget for actuation, i.e., to align it on the table, to hold it in place, or to move subparts of it. When designing the widgets, we made them as lightweight as possible to reduce friction.

ACTUATION

Our actuation algorithm triggers distinct electromagnets in the array to attract or repel the permanent magnets attached to the widgets. In contrast to previous approaches, we consider the relationships between the widget's magnets to enable more degrees of freedom for actuation. Each control is decomposed into a set of rigid bodies that are linked with joints. The knob in Figure 4, e.g., consists of a base rigid body to position the control on the tabletop and a moveable inner body to set a continuous value. The two rigid bodies are connected by a pivot joint. For each rigid body our algorithm can apply different actuation forces to its permanent magnets:

- *Tangential actuation* moves magnets in the plane of the tabletop. This is used to translate and orient widgets on the tabletop but also to set physical values in x/y direction, e.g., the position of a slider.
- *Normal actuation* applies force in the normal direction of the tabletop, i.e., away or towards the surface in z direction. This is, e.g., applied to pushbuttons to move the button plate up and down.

In addition to these forces each electromagnet can be triggered to create an arbitrary PWM signal. This can be used, e.g., to transfer electrical power to a widget as described later.

Model

Our algorithm applies a target force $F(P_i) \in \mathbb{R}^3$ to each permanent magnet P_i centered at position $pos(P_i) = (x_i, y_i, z_i)^T$ to move it to a new position. To achieve that, it computes the required pulling or repelling force for each electromagnet E_j centered at position $pos(E_j) = (x'_j, y'_j, z'_j)^T$ below the surface. The vertical spacing between permanent magnets and electromagnets $d_z = z_i - z'_j$ is approximately 1 cm and constant for all magnets. Since a single electromagnet can affect multiple permanent magnets, we add the forces of multiple electromagnets for a single permanent magnet. This approximation has been successfully applied by Pangaro et al. [17]. Thus, the target force $F(P_i)$ for the permanent magnet P_i reads

$$F(P_i) = \sigma(P_i) \cdot \sum_j F(E_j) \cdot \begin{pmatrix} f_x(E_j, P_i) \\ f_y(E_j, P_i) \\ f_z(E_j, P_i) \end{pmatrix} \quad (1)$$

where $\sigma(P_i)$ is a constant scalar factor of the permanent magnet derived from size, material, and magnetic properties. $f_{x/y/z}$ are distance dependent factors for each spatial

component. $F(E_j)$ is the unknown force that determines the power to drive the electromagnet E_j . Note that the factors $\sigma(P_i)$ and $F(E_j)$ for magnetic forces can be positive or negative depending on the polarization. Regarding all permanent magnets equation (1) yields a system of equations that we have to solve to achieve the given target forces.

The force of electromagnet E_j exerted on a permanent magnet P_i decreases quadratically with their distance $d = \|pos(E_j) - pos(P_i)\|$ and is nearly proportional to the factors $F(E_j)$ and $\sigma(P_i)$ of both magnets. To reduce the complexity of the system of equations, we introduce a threshold d_{\max} denoting the maximum distance at which a permanent magnet is noticeable affected by an electromagnet. The resulting force factor $f_x(E_j, P_i)$ for electromagnet E_j pulling or pushing a permanent magnet P_i in x direction is calculated by

$$f_x(E_j, P_i) = \begin{cases} \frac{1}{d^2} \cdot \left(\frac{x'_j - x_i}{\sqrt{(x'_j - x_i)^2 + (y'_j - y_i)^2}} \right), & d \leq d_{\max} \\ 0, & \text{otherwise.} \end{cases}$$

The factor for the y direction is calculated accordingly. As we can only track the z position of an object with limited accuracy, we neglect this position in the following force factor term. The proportion of E_j pulling or pushing a permanent magnet P_i into z direction is calculated by

$$f_z(E_j, P_i) = \begin{cases} \frac{1}{d^2} \cdot \lambda, & d \leq d_{\max} \\ 0, & \text{otherwise} \end{cases}$$

where λ is the ratio between the maximum x/y force and the maximum z force depending on the electromagnetic field.

We have to find the $F(E_j)$ for equation (1) whereas the factor for each electromagnet is bounded by its maximum power F_{\max} :

$$-F_{\max} \leq F(E_j) \leq F_{\max}$$

If a solution exists, this system is under-determined at most because forces generated by distinct electromagnets can be substituted by other ones. This allows us to optimize the solution by taking additional constraints into account. Considering these characteristics, we applied linear optimization to solve the system of equations. Our objective function that we want to minimize reads

$$\sum_j \lambda_j \cdot |F(E_j)| \quad (2)$$

where λ_j are the weights for each electromagnet. We utilize the *Coin-or linear programming* library² to minimize the objective function. The easiest way to choose the weights λ_j is to set all to 1. This minimizes the total force and, hence, minimizes power consumption and heating. However, as explained later, we dynamically adapt these weights to optimize the performance of the electromagnets and to protect them from damage.

²<https://projects.coin-or.org/Clp>

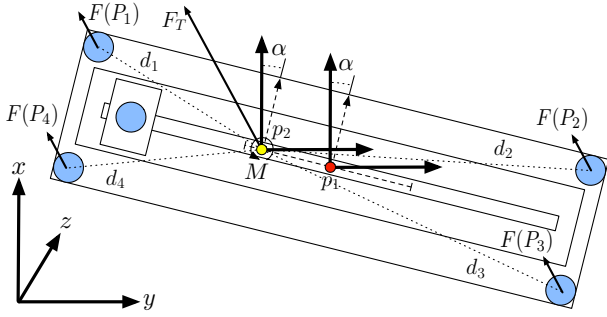


Figure 5: Slider widget base. Actuation is realized by applying forces to the permanent magnets $P_{1...4}$ (blue). If the sliding knob is moved, the widget's center of gravity p_2 (yellow) moves along the dotted line. The widget is affected by the force F_T and the torque M .

Widget actuation

To provide the movement, orientation, and configuration of complex controls, the algorithm must be aware of their physical properties. We illustrate this actuation step based on a slider shown in Figure 5. To start the actuation of a control, the application designer defines a target position p_T and angle α_T referred to the geometrical center p_1 (red) of the widget. The algorithm computes the path leading from the current to the target position. Our current implementation uses a simple straight line, but this could easily be extended to spline paths, e.g., in order to avoid collision of controls.

An actuation frame is the time period a distinct magnet configuration endures. Since we constantly update all magnets with 30 fps, every actuation frame takes $t_{af} \cong 33$ ms. In each actuation frame, we use a rigid body physics model for actuation, and shift all forces that influence a widget into the center of gravity p_2 . For a widget that contains adjustable parts, such as the slider, the center of gravity moves and the mass distributions varies when the widget's physical state is changed. For widget actuation, target velocity v_t and angular velocity ω_t are predefined. Dependent on these velocities and t_{af} , a translation by Δp_1 and a rotation by $\Delta \alpha$ is achieved after the next frame. This displacement is converted to Δp_2 related to the center of gravity ($\Delta \alpha$ stays unchanged). Taking the current two-dimensional velocity v_c in x and y direction and the current angular velocity ω_c derived from sensing into account, the force F_T that has to be applied for translation is

$$F_T = m \cdot \left(\frac{\Delta p_2}{t_{af}^2} - \frac{v_c}{t_{af}} \right)$$

and the torque M around the z -axis for rotation is

$$M = J \cdot \left(\frac{\Delta \alpha}{t_{af}^2} - \frac{\omega_c}{t_{af}} \right)$$

with given mass m and moment of inertia J . Force and torque are distributed to the permanent magnets where

$$M = - \sum_i \left((d_{ix} \cdot F(P_i)_y) - (d_{iy} \cdot F(P_i)_x) \right) \quad (3)$$

with levers d_i (Figure 5) and

$$F_T = \sum_i \begin{pmatrix} F(P_i)_x \\ F(P_i)_y \end{pmatrix} \quad (4)$$

must be fulfilled at once. To derive $F(P_i)_x$ and $F(P_i)_y$ satisfying this system of equations we offer three approaches.

Preferring translation, the force applied to each permanent magnet is set to $\frac{F_T}{\|F_T\|} \cdot \theta_i$, where θ_i is variable. Thus, we derive a further under-determined system of equations that is solved using linear optimization. By solving $\sum_i |\theta_i| \rightarrow \min$ we minimize the absolute force. Accordingly, actuation is optimized for rotation by fixing each force $F(P_i)$ perpendicular to the corresponding lever d_i . The third approach solves

$$\sum_i F(P_i)_x^2 + F(P_i)_y^2 \rightarrow \min$$

by utilizing Lagrange multipliers subject to equation (3) and (4). Combining rotation and translation this method avoids peak forces applied to single permanent magnets. Consequently, all practices yield valid solutions by using different configurations of the magnets. In fact, the main difference is the achievable maximum velocity that is around 5–8 cm per second and 10–15 degree per second.

To overcome friction, the mass m_i resting on every permanent magnet and varying interrelated with the center of gravity and the friction coefficient μ_F must be known. The target forces $F(P_i)$ for each permanent magnet are scaled up to compensate for friction. The target z force $F(P_i)_z$ for each magnet must be set to 0 to keep the friction as constant as possible. If distinct parts of a widget have to be moved, a z force is applied to the fixed permanent magnets keeping the widget in position.

Note that all these forces are calculated for a discrete actuation frame. This approximation only works if the frame rate in the control loop consisting of actuation and sensing is high enough.

Heat prevention

Since we noticed that the electromagnets become quite hot after 40–60 seconds, we dynamically adapt the weights λ_j in equation (2). If the estimated temperature of a magnet E_j exceeds a distinct threshold (e.g., 60° C), λ_j is doubled for a specific time period. Consequently, this magnet will be substituted by other ones, while the solution space is not affected by this method.

TRACKING

Widget actuation requires us to precisely determine widget position and orientation on the table. However, since we arranged a fiber optic matrix between the opaque magnets, the incoming camera image exposes a very coarse dot matrix of the touch surface. As mentioned before, each widget contains a set of markers that make up a unique footprint. A unicolor marker would lead to an ambiguous touch detection (Figure 6a). Since we cannot increase the spatial resolution of the incoming dot image, we increase the resolution of each dot by introducing *gradient fiducials* to the widgets.

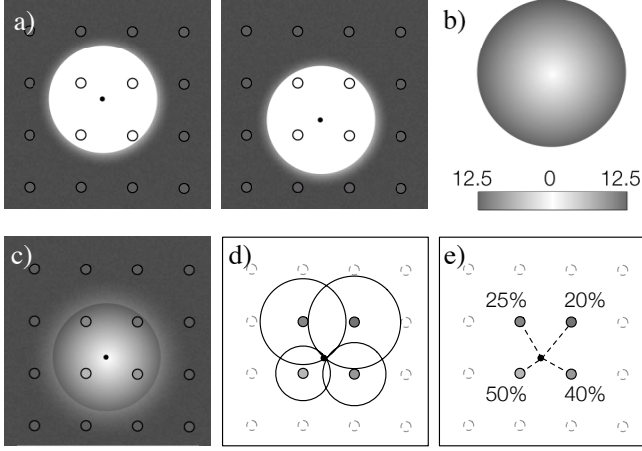


Figure 6: Gradient marker approach. a) Unicolored markers are ambiguous. b) Gradient markers encode radius as brightness. c) Marker communicates subsampled matrix of radii. d) Exact solution. e) Approximate solution by bilinear weighting.

Gradient markers

We apply the idea of Kojima et al. who project gradient patterns onto robots on a tabletop to determine their position [14]. We use circular markers with a diameter of 25 mm for each widget and print a circular gradient pattern onto them (Figure 6b). The pattern linearly fades from white in the center to dark gray at the border. The brightness value of a marker’s point maps to a specific radius. If a marker is placed on the tabletop, it reflects the incoming IR light according to the gradient and, therefore, provides a subsampled matrix of radii (Figure 6c). For the position p_i of each dot D_i , the center c of the marker lies on a circle around this dot with radius r_i . By intersecting all circles, the exact center can be computed (Figure 6d). Due to camera noise and minor variations in the fiber optic cables, the best matching center point can be derived by minimizing the squared distance error:

$$\sum_i (\|c - p_i\| - r_i)^2 \rightarrow \min$$

We approximate this non-linear problem by computing c as an affine combination of all points p_i (Figure 6e) weighted according to their proximity to the center of the marker:

$$c = \frac{1}{\sum_i \omega_i} \cdot \sum_i \omega_i \cdot p_i$$

with weights

$$\omega_i = r_{\text{marker}} - r_i \quad (5)$$

where r_{marker} denotes the radius of the marker.

Calibration

To compensate for variations in the IR light distribution in the Endlighten layer and minor differences in the light conduction of the fiber optics, we calibrate the table in advance.

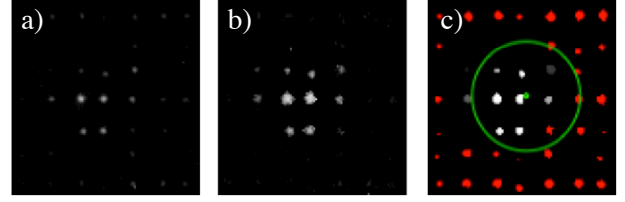


Figure 7: Extract from image processing pipeline. a) Raw input image of fiber optics beneath marker. b) Normalized image after mapping to range between background and foreground brightness. c) Cell representatives and detected marker. Red dots are grid cells below the threshold.

We capture a frame with no objects on the table (background frame), a white sheet placed on the tabletop (foreground frame with maximum brightness), and a dark gray sheet (minimum brightness reflected by markers).

Accordingly, we obtain a low threshold l_i (dark gray layer) and a high threshold h_i (white sheet) for each dot D_i in the matrix. We replace the weights in equation (5) by

$$\omega_i = \frac{b_i - l_i}{h_i - l_i} \quad (6)$$

where b_i is the current brightness value of the dot.

In order to exclude all pixels in the camera image that are not covered by fiber optics, we precompute a binary map of *active pixels*. A pixel is considered as active if the difference between its corresponding foreground and background pixel exceeds a certain threshold.

Image processing pipeline

The spot detection extracts a subsampled grid from each camera image, where each cell corresponds to an area containing exactly one fiber optic cable. Subsequently, it computes a representative for each cell by averaging the brightness values of its active pixels. The spot detection then computes the connected components of all dots D_i in the grid that exceed the lower threshold ($b_i > l_i$). Then, for each component it determines the weights in equation (6) and the approximated center point of the marker as described above. The most important steps of the image processing pipeline are illustrated in Figure 7.

In addition, we detect finger touches by searching for local maxima above a certain threshold. Despite the low input resolution, the algorithm can identify multiple touches and dragging gestures for basic tasks, such as moving and transforming images.

Widget detection

We use a simple pattern matching approach to detect Madgets on the table. Our algorithm first filters out all spots with inappropriate shape and size, before iterating over all spots to identify widget footprints. When it detects a footprint, it computes the mapping from footprint to tabletop coordinate system by minimizing the sum of squared distances between the footprint items and the corresponding spots.

EXPLORING THE DESIGN SPACE

The ability to hold a Madget in place while actuating parts of it enables a variety of new applications for tangibles on tabletops. These are described below.

General-Purpose Widgets

Our actuated table supports moving and configuring general-purpose Madgets, such as buttons, sliders, knobs, etc. A user can, e.g., place a slider on the table to navigate through a video. After starting the video, the slider's knob follows the relative time position in the video, providing haptic feedback of its progress. The LCD panel displays the corresponding timeline slider visuals beneath the physical slider. If desired, the user can take action and move the knob to a different position in the video.

Persistence. Madgets let us transfer technologies that are established in GUI desktop applications to tangible controls. E.g., the physical configuration of a Madget can be saved and restored later. This is especially useful when several people use a table in sequence for individual tasks. As a direct corollary, we can provide functions such as physical undo and redo for widgets, as suggested by Pangaro et al. [17].

Remote Collaboration. We can also expand the idea of remote collaboration [17] to tabletop widgets. Multiple users can collaboratively work on the same task using private and shared physical controls. If a crucial value is manipulated and relevant to all remote users, a shared control is helpful. If the cost of a product, e.g., should be estimated, each user could own a knob that is linked to the same value. Now one user can feel when the other sets a different value, and physically intervene if necessary. Therefore, a shared Madget enables *tangible presence* [2] of another user, functioning as a physical input and output device and increasing the awareness of a remote user's actions by providing haptic feedback.

Ad-hoc use. Madgets support using them in a more dynamic, ad-hoc manner. If a Madget is out of reach, the user can draw a gesture (e.g., a circle for a knob) to pull it to a desired position, or operate it indirectly using one of the various techniques to reach distant objects [16]. Likewise, the Madget can be moved outside the work space using simple gestures. This narrows the gap between virtual and physical widgets in terms of flexibility.

Going 3D: Height

As mentioned earlier, height has become increasingly important for tangibles on tabletops. However, actuated tangibles have been mostly limited to two-dimensional movement. Using our system, we can keep a Madget in place and lift parts of it up from the table. Physical constraints are in place to prevent tangential movement.

Buttons. As demonstrated in Figure 8, we applied this concept to a radio button Madget with three buttons. The Madget is mounted on an acrylic base with four permanent magnets, one at each corner. A button consists of an acrylic box that is open at the top and contains a thin, lightweight acrylic plate. A magnet in the center of each plate enables its actuation. Our actuation algorithm holds the Madget in place by attracting the base magnets using inverse polarization. Fur-

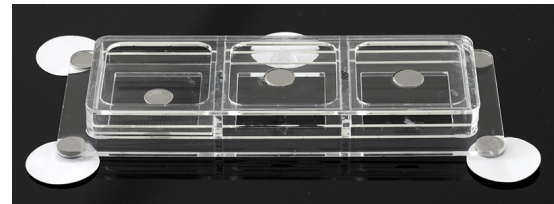
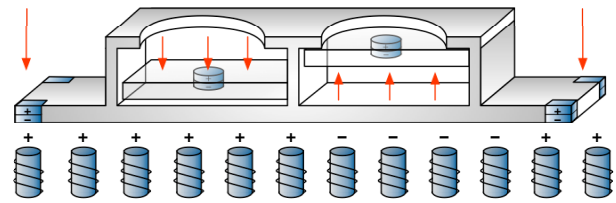


Figure 8: Actuated radio buttons. Top: actuation concept. Bottom: prototype with three buttons.

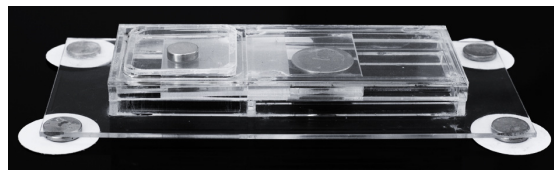
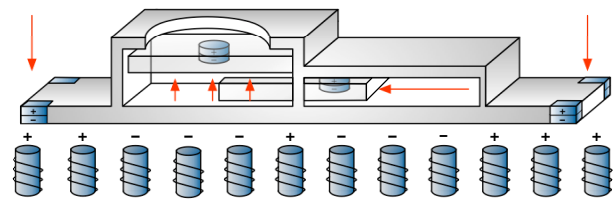


Figure 9: Check box with locking mechanism. The clutch slides sideways to lock the button pad in place.

thermore, it pushes the plate of each unchecked button up by applying a repellent magnetic field. Horizontal motion of each plate is inhibited by vertical acrylic rails. Thus, the user cannot only feel the button shapes (as in [28]) but also their current state. The tabletop LCD panel adds visual feedback from below. If the user wants to select a different button, she pushes it down against the magnetic force, and a small marker beneath the plate magnet produces a spot in the camera. The button state is changed accordingly, the graphics are updated, and the other, now unchecked, buttons are raised.

Clutch. The concept of height enables subsequent movements of controls that involve both dimensions, tangential and normal motion. As an example, we implemented a clutch control that moves a piece of acrylic to lock or unlock a moveable part (Figure 9). Adapting the technique described in the previous section, we created a pushbutton widget that raises an acrylic plate to provide haptic feedback when pressing the button. However, we added the ability to lock the button, or, in the terminology of GUI applications, to “gray it out”. We added a bar pad with a single magnet that can be moved in tangential direction only. To disable the button, the push plate is raised (normal direction), and, thereafter, the bar is shifted beneath the plate. Now the button cannot be pushed anymore. The inverse order of actuation enables the

control again. The sequence of actuations can be extended arbitrarily to create more complex constraints, e.g., dynamic range limits for sliders.

Force feedback

Beside the inherent haptic feedback of tangible controls, actuation can provide active force feedback as an additional output channel.

Resistance. By default, moveable parts such as the turning arm of a knob can be rotated freely. Actuation, however, allows us to change the resistance or the perceived friction of an object by adapting the PWM signal. By attracting a slider knob we can make it harder to move. This can be helpful if an important value is changed or to signal the user that she is exceeding a default range. If a knob, e.g., controls the volume of music in a public place, the knob could be harder to rotate beyond a specific threshold. In the same fashion, the resistance of the aforementioned buttons could be modified by pushing the plate up with different power levels. Applying a maximum attracting force to parts of Madgets could be used to lock them, e.g., while they are to be operated exclusively through program control, or by a remote user.

Vibration feedback. The tangential actuation in our algorithm can be used to let Madgets vibrate. This can act as an audio-haptic signal that a Madget needs attention, e.g., when a critical value is reached or when a remote user has changed a value. Vibration patterns could also be used to create more complex feedback.

Dynamic notches. Notches let users detect steps in the scale of a continuous control without looking. We simulate this by adding a simple beat feedback whenever a certain step is exceeded. Any continuous Madget can be equipped with a cylindrical magnet in a closed acrylic pipe. By creating a repelling magnetic field beneath the magnet, it is raised quickly and hits the top of the pipe, making the Madget vibrate noticeably. The step size of the notches can be set dynamically.

Water wheel Madgets

The benefits of passive controls often come at the price of a restricted design space when creating them. In this section, we present two concepts to transfer energy from the actuation table to a Madget to enrich its functionality. In allusion to the drives for ancient mills, we call these Madgets water wheels.

Inductive energy transfer. Induction is a well understood charging technique for devices ranging from electric tooth brushes to mobile phones. We can apply this basic principle to transfer power from the table to Madgets in order to support electronics without the need for batteries or cables. Figure 10 shows the induction Madget, our first prototype. It consists of a square acrylic plate and, like all other Madgets, of magnets at its corners to hold it in place. An inductor and a red LED are placed in the middle of the plate. Holding the Madget in place and creating a magnetic field with a sinusoid PWM signal beneath it transfers power to the inductor, causing the LED to blink with a specific frequency. In further iterations, the LED could be replaced with other electronics, such as memory chips to store data, or a small radio circuit

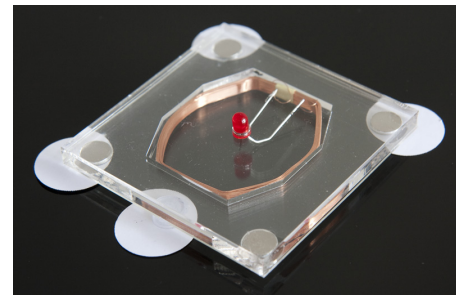
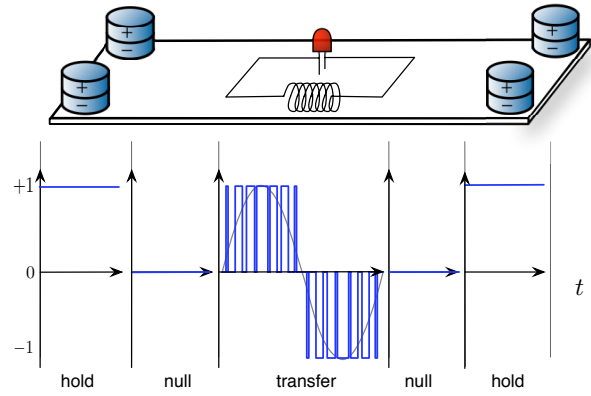


Figure 10: Induction Madget demonstrating wireless power transfer. PWM signals are area-specific, and premultiplied with a polarization factor.

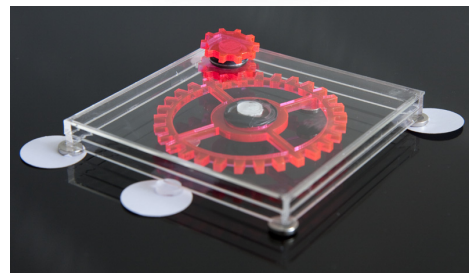
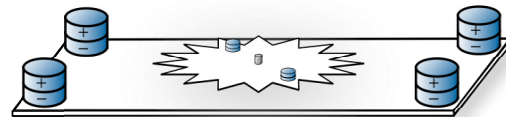


Figure 11: The motor Madget, a building block for complex actuations, uses an actuated gear wheel.

to wirelessly transmit information from additional sensors on the Madget to the environment.

Motors. The motor Madget consists of a tangentially pivoted acrylic gear wheel that contains two magnets on opposite sides. As in a motor, we apply a dynamic magnetic field to rotate the gear wheel at a constant velocity. The wheel can act as a basic building block to create controls with more complex mechanics, such as small robots (Figure 11).

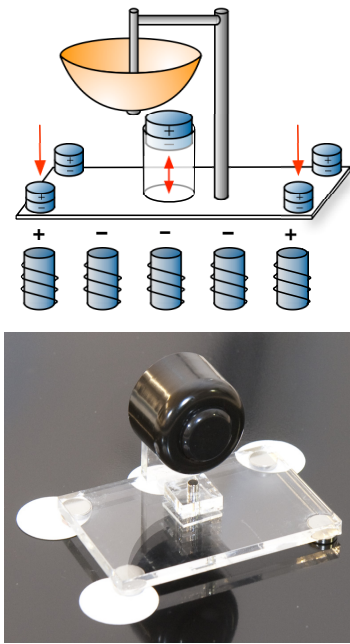


Figure 12: The bell Madget enables mechanical alerts, using a bike bell and magnetically repelled beater.

Mechanical Audio Feedback

Height and force feedback also enable mechanical sound output. Our first prototype, a bell Madget (Figure 12) consists of an off-the-shelf bicycle bell mounted on an acrylic plate. At the bottom of the widget, a cylindrical permanent magnet, the beater, is placed within an acrylic pipe that constrains its motion to up and down. To create a ping, the widget is held in place, and the system creates a strong repelling magnetic field impulse under the beater. The beater skyrockets, hits the bell, and creates the ping sound. Varying the actuation could create different sounds, and many other mechanical acoustic Madgets are imaginable.

PROTOTYPING

Prototypes are crucial in the development process of new devices. However, the number of conducted iterations are often not just quality-driven but also a matter of time and cost constraints. Our system provides a platform for rapid prototyping of medium fidelity prototypes. Using a laser cutter, crafting a new widget is a matter of an hour, as a vast palette of low-cost materials is available and no electronics are involved. Tracking and actuation are maintained by glueing respective markers to the control. Registering new controls in software currently takes about two hours; we are developing an application to simplify this process. Once integrated, the designer can freely modify the visual and haptic behavior of her controls. She can create mappings between continuous values and actuation, e.g., she can link a value to a magnetic field to push up a plate of a button. If a user test on the device yields that the button's resistance is inappropriate, the designer just changes the value rather than building a new prototype with different mechanics. Thus, our system suggests performing more iterations on medium fidelity prototypes, which lowers cost and effort in hardware design.

CONCLUSION AND FUTURE WORK

We presented a novel actuation mechanism based on magnetic actuation that can move and configure tangible widgets on an interactive tabletop. Our algorithm decomposes complex controls into rigid bodies and regards their dynamic weight distribution during actuation. We track objects on the tabletop by employing a grid of fiber optics that let tracking cameras see past the actuation hardware. Our system provides low-cost physical widgets and new actuation dimensions, such as height, force feedback, and power transfer.

Although our markers leave sufficient space for the visual output to give the controls a dynamic appearance, they still have a diameter of 25 mm each. We intend to replace them with transparent markers, e.g., by using technologies such as polarization [13] or infrared blocking films [12]. Furthermore, we want to optimize the background lighting of the LCD panel. As the fiber optics cause small black dots in the graphical user interface, we aim to compensate for this by feeding additional background lighting through the fiber optics from inside the table. As an alternative, we consider to incorporate one of the upcoming thin multi-touch technologies, such as Interpolating Force Sensitive Resistance [24]. This would make our system more compact and less sensitive to ambient IR light.

We are currently working to identify appropriate feedback techniques when interacting with Madgets. For example, how should we indicate the anticipated trajectory of an actuated control, or deal with physical operations triggered by a remote user? On a higher level, how do users respond to multitouch systems that exhibit actuation? Since actuation reduces the user's control for the sake of physical-digital consistency, we need to find a way to strike a balance between these conflicting goals.

Finally, we are developing a rapid prototyping toolkit using Madgets that offers easy registration and integration of physical prototypes into actuated tabletops. This will allow designers to easily add actuation mechanisms to their controls using rotating parts, height displacement, or arbitrary PWM signals. Changing the physical resistance of a mechanical sliding knob, or the strength of a vibration feedback, could then become a matter of a few clicks.

ACKNOWLEDGMENTS

This work was funded in part by the German B-IT Foundation. We thank Helga Isenberg, Lucas Braun, Christian Remy, and Simon Voelker for their help to build the system. Special thanks to Patrick Baudisch for his valuable input to the paper.

REFERENCES

1. P. Baudisch, T. Becker, and F. Rudeck. Lumino: Tangible blocks for tabletop computers based on glass fiber bundles. In *Proc. of SIGCHI '10*. ACM, 2010.
2. S. Brave, H. Ishii, and A. Dahley. Tangible interfaces for remote collaboration and communication. In *Proc. of CSCW '98*, pp. 169–178. ACM, 1998.
3. R. Gabriel, J. Sandsj, A. Shahrokni, and M. Fjeld. BounceSlider: actuated sliders for music performance

- and composition. In *Proc. of TEI '08*, pp. 127–130. ACM, 2008.
4. S. C. Goldstein, T. C. Mowry, J. D. Campbell, M. P. Ashley-Rollman, M. D. Rosa, S. Funiak, J. F. Hoburg, M. E. Karagozler, B. Kirby, P. Lee, P. Pillai, J. R. Reid, D. D. Stancil, and M. P. Weller. Beyond audio and video: Using claytronics to enable pario. *AI Magazine*, 30(2), 2009.
 5. C. Harrison and S. E. Hudson. Providing dynamically changeable physical buttons on a visual display. In *Proc. of SIGCHI '09*, pp. 299–308. ACM, 2009.
 6. O. Hilliges, S. Izadi, A. D. Wilson, S. Hodges, A. Garcia-Mendoza, and A. Butz. Interactions in the air: adding further depth to interactive tabletops. In *Proc. of UIST '09*, pp. 139–148. ACM, 2009.
 7. J. Hook, S. Taylor, A. Butler, N. Villar, and S. Izadi. A reconfigurable ferromagnetic input device. In *Proc. of UIST '09*, pp. 51–54. ACM, 2009.
 8. H. Ishii. Tangible bits: Beyond pixels. In *Proc. of TEI '08*, pp. xv–xxv. ACM, 2008.
 9. H. Iwata, H. Yano, F. Nakaizumi, and R. Kawamura. Project FEELEX: adding haptic surface to graphics. In *Proc. of SIGGRAPH '01*, pp. 469–476. ACM, 2001.
 10. S. Izadi, S. Hodges, A. Butler, A. Rrustemi, and B. Buxton. ThinSight: integrated optical multi-touch sensing through thin form-factor displays. In *Proc. of the 2007 workshop on Emerging displays technologies: images and beyond: the future of displays and interaction*, p. 6, San Diego, California, 2007. ACM.
 11. D. Jackson, T. Bartindale, and P. Oliver. FiberBoard - compact multi-touch display using channeled light. In *Proc. of ITS '09*, pp. 25–28. ACM, 2009.
 12. Y. Kakehi, T. Hosomi, M. Iida, T. Naemura, and M. Matsushita. Transparent tabletop interface for multiple users on lumisight table. In *Proc. of TABLETOP '06*, pp. 143–150. IEEE Computer Society, 2006.
 13. H. Koike, W. Nishikawa, and K. Fukuchi. Transparent 2-D markers on an LCD tabletop system. In *Proc. of SIGCHI '09*, pp. 163–172. ACM, 2009.
 14. M. Kojima, M. Sugimoto, A. Nakamura, M. Tomita, M. Inami, and H. Nii. Augmented coliseum: An augmented game environment with small vehicles. In *Proc. of TABLETOP '06*, pp. 3–8. IEEE Computer Society, 2006.
 15. D. Leithinger and H. Ishii. Relief: A scalable actuated shape display. In *Proc. of TEI '10*, pp. 221–222. ACM, 2010.
 16. M. A. Nacenta, D. Aliakseyeu, S. Subramanian, and C. Gutwin. A comparison of techniques for multi-display reaching. In *Proc. of SIGCHI '05*, pp. 371–380. ACM, 2005.
 17. G. Pangaro, D. Maynes-Aminzade, and H. Ishii. The actuated workbench: Computer-controlled actuation in tabletop tangible interfaces. In *Proc. of UIST '02*, pp. 181–190. ACM, 2002.
 18. J. Patten and H. Ishii. Mechanical constraints as computational constraints in tabletop tangible interfaces. In *Proc. of SIGCHI '07*, pp. 809–818. ACM, 2007.
 19. J. Patten, B. Recht, and H. Ishii. Audiopad: A tag-based interface for musical performance. In *Proc. of NIME '02*, pp. 1–6. National University of Singapore, 2002.
 20. I. Poupyrev, T. Nashida, S. Maruyama, J. Rekimoto, and Y. Yamaji. Lumen: Interactive visual and shape display for calm computing. In *Proc. of SIGGRAPH '04*, p. 17. ACM, 2004.
 21. I. Poupyrev, T. Nashida, and M. Okabe. Actuation and tangible user interfaces: The vaucanson duck, robots, and shape displays. In *Proc. of TEI '07*, pp. 205–212. ACM, 2007.
 22. J. Rekimoto, B. Ullmer, and H. Oba. DataTiles: a modular platform for mixed physical and graphical interactions. In *Proc. of SIGCHI '01*, pp. 269–276. ACM, 2001.
 23. D. Reznik and J. Canny. C'mon part, do the local motion! In *Proc. of ICRA '01*, pp. 2235–2242, Seoul, South Korea, 2001.
 24. I. Rosenberg and K. Perlin. The UnMousePad: an interpolating multi-touch force-sensing input pad. In *Proc. of SIGGRAPH '09*, pp. 1–9. ACM, 2009.
 25. D. Rosenfeld, M. Zawadzki, J. Sudol, and K. Perlin. Physical objects as bidirectional user interface elements. *IEEE Computer Graphics and Applications*, 24(1):44–49, 2004.
 26. A. Shahrokni, J. Jenaro, T. Gustafsson, A. Vinnberg, J. Sandsj, and M. Fjeld. One-dimensional force feedback slider: Going from an analogue to a digital platform. In *Proc. of NordiCHI '06*, pp. 453–456. ACM, 2006.
 27. M. Weiser. The computer for the 21st century. *Scientific American*, 265(3):94–104, 1991.
 28. M. Weiss, J. Wagner, Y. Jansen, R. Jennings, R. Khoshabeh, J. Hollan, and J. Borchers. SLAP widgets: Bridging the gap between virtual and physical controls on tabletops. In *Proc. of SIGCHI '09*, pp. 481–490. ACM, 2009.
 29. R. Wimmer. FlyEye: grasp-sensitive surfaces using optical fiber. In *Proc. of TEI '10*, pp. 245–248. ACM, 2010.