Kickables: Tangibles for Feet

Dominik Schmidt, Raf Ramakers[1], Esben W. Pedersen[2], Johannes Jasper, Sven Köhler, Aileen Pohl, Hannes Rantzsch, Andreas Rau, Patrick Schmidt, Christoph Sterz, Yanina Yurchenko, Patrick Baudisch

Hasso Plattner Institute, Potsdam, Germany {first.last}@hpi.uni-potsdam.de

[1]Hasselt University, iMinds Diepenbeek, Belgium raf.ramakers@uhasselt.be

[2]University of Copenhagen, Copenhagen, Denmark esbenwp@diku.dk

Figure 1: This example of a museum exhibit on basic molecules allows visitors to interact by kicking physical objects around—which we call *kickables*. (a) This visitor starts a tutorial video by pushing a kickable from pause to play. (b) Another visitor scrubs through a different video. (c) This visitor assembles a water molecule by moving a red hydrogen atom towards a blue oxygen atom.

ABSTRACT

We introduce the concept of tangibles that users manipulate with their feet. We call them *kickables*. Unlike traditional tangibles, kickables allow for very large interaction surfaces as kickables reside on the ground. The main benefit of kickables over other foot-based modalities (e.g., foot touch), is their strong affordance, which we validate in two user studies. This affordance makes kickables well-suited for walk-up installations, such as tradeshows or museum exhibits.

We present a custom design as well as five sets of standard kickables to help application designers create kickable applications faster. Each set supports multiple standard controls, such as push buttons, switches, dials, and sliders. In doing so, each set explores a different design principle, in particular different mechanical constraints. We demonstrate an implementation on our pressure-sensing floor.

Author Keywords: Tangibles; Interactive Floor; Foot-Based Interaction; Affordance.

ACM Classification Keywords: H.5.2 [Information interfaces and presentation]: User Interfaces: Input Devices and Strategies; Interaction Styles.

INTRODUCTION

Tabletop computers have been successfully deployed in public settings, such as museum exhibits, art galleries, or tradeshows, because of their ability to engage visitors (e.g., FlowBlocks [5]). The key is that they offer strong affordance and sufficient functionality to allow for walk-up use [4].

CHI 2014, April 26 - May 01 2014, Toronto, ON, Canada

ACM 978-1-4503-2473-1/14/04…$15.00.
http://dx.doi.org/10.1145/2556288.2557016

On the flipside, tabletops only accommodate a few users at a time (e.g., two to eight [27]), as the size of the interaction surface is limited by users' arm length [3]. This prevents tabletops from scaling to the dozens or hundreds of visitors that museums and tradeshow exhibits tend to attract. The limited size of the interaction surface also limits tabletops to display objects that fit on a table. Thus, for instance, no life-size cars, dinosaur skeletons, or large multi-user interaction spaces, such as the chemistry simulation in Figure 1c.

In order to accommodate dozens or hundreds of visitors at a time, researchers have proposed interactive floors (e.g., iGameFloor [12] or Multitoe [3]). For instance, the Epidemik [36] exposition in Paris' Cité des Sciences museum used 37 ceiling mounted projectors and 31 cameras for tracking to create a 450 m² seamless interactive floor, allowing about 100 collocated visitors to *simultaneously* share a single interactive experience.

Unfortunately, current floor installations lack the key benefit that made interactive tabletops attractive for the task in the first place, i.e., they either fail to offer sufficient interactive functionality or they lack affordance. The reason for the lack of functionality is that users are in constant physical contact with the floor, limiting users to interact with items by *crossing* them (e.g., Epidemik [36]). Augsten et al. [3] demonstrated how to address this shortcoming (they used gait recognition to allow users to operate buttons by *tapping*) but at the expense of affordance, as 22 of 30 participants of their guessability study proposed a *different* mechanism (e.g., jumping onto the button, use of a specific foot, etc.).

Since affordance *and* functionality are mission critical in public installations, we propose a different approach: We move all input into a separate physical "layer" above the floor surface. We implement this concept using *kickables*—tangible objects that users operate using their feet.

KICKABLES

Figure 1 shows kickables enabling an example science exhibit on basic molecular chemistry. The exhibit allows visitors to explore the composition of different molecules, such as water or carbon dioxide. Visitors do so by moving atoms around to engage in chemical bonds with other atoms. These atoms are implemented as kickables.

In analogy to tangible user interfaces [16], kickables consist of a physical representation (the *knob*) coupled to a digital representation, which provides visual feedback (the *screenlet*). While we designed kickables with feet in mind, users may manipulate kickables using their hands as well. Either way, they obtain the benefit offered by kickables, i.e., a large interaction area.

As with tangibles, kickable application designers will typically create their own *custom* kickables. In the shown chemistry exhibit, for example, kickables are used to represent different types of atomic nuclei. Figure 2a shows how to operate them. Here, a visitor pushes a red hydrogen towards a blue oxygen atom, creating hydroxyl. Another visitor tries to add oxygen, causing an invalid configuration (2b), before rolling over another hydrogen atom to form the water molecule (2c) shown in Figure 1c.

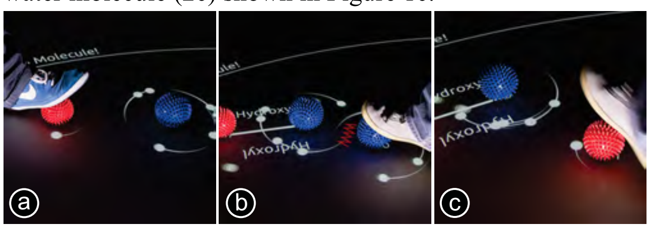

Figure 2: A custom kickable: A visitor (a) pushes a hydrogen towards an oxygen atom (b) forming hydroxyl; another visitor tries adding a second oxygen atom, but the floor indicates that this is an invalid configuration. (c) Rolling over another hydrogen atom forms the water molecule shown in Figure 1c.

At the same time, many applications will benefit from *reusable* kickables. The visitor in Figure 1a, for example, starts a tutorial video by kicking the knob of a *kickable switch* from play to pause. Another visitor (1b) scrubs a video using a *kickable slider*, pushing the knob for precise control. To support application designers in creating kickable applications, we dedicate a substantial part of this paper to the design of standard kickable widgets.

Tracking

The specific kickable designs shown in this paper were optimized for use with a pressure sensing, back-projected floor, such as the one illustrated in Figure 3 (see [6]). To allow for this, all knobs are designed to create discernible imprints on the floor to facilitate recognizing and tracking them (see *implementation* section for details). A key benefit of this implementation is that kickable knobs are passive (no batteries, no electronics) and thus easy to maintain.

The concept of kickables, however, goes beyond the specifics of our floor-based implementation. They may be implemented on a variety of floor displays (e.g., using steerable projectors [22]) and tracked using a variety of tracking solutions (e.g., 2D or depth cameras, optical trackers like Vicon, or inertial measurement units (IMU) inside knobs). We demonstrate one such solution (based on overhead IR cameras) in section *alternative top-camera tracking*.

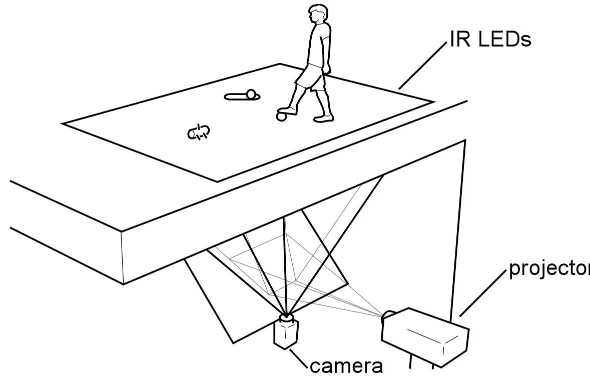

Figure 3: While kickables can be implemented with different tracking hardware, the specific designs presented in this paper are optimized for a pressure-sensitive floor based on FTIR [6].

Design Challenges

The design of kickables raises a series of questions around which we have created our five sets of standard kickables: How to engineer knobs in terms of shape, weight, and materials to allow for precise, controlled movements? How to keep knob and screenlet aligned without leaving the range of valid values? How to allow users to acquire frequently used values despite humans' limited precision with their feet? At the same time, we need to design kickables to allow us to track them on our specific floor. Figure 4 shows some of the resulting designs, one representative per set.

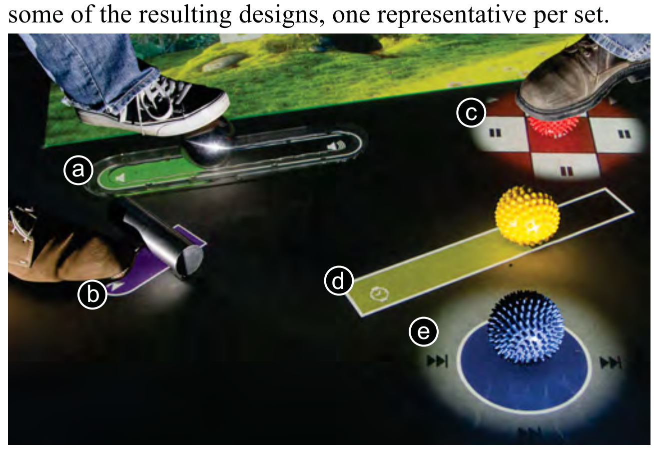

Figure 4: Sample widgets from the five sets of standard kickables: (a) *bento* slider, (b) *tumbler* button, (c) *tessellation* toggle switch (d) *tracking* slider, and (e) *crossing* button.

CONTRIBUTION

The primary contribution of this paper is the concept of kickables, that is, tangibles designed to be manipulated using feet. The biggest strength of kickables is that they can be used with much larger interaction spaces than tangibles. They also provide good affordance, which we validated in two user studies. While feet are much more limited than the hands we usually design for (e.g., they offer limited accuracy and are generally unable to grasp, lift, or carry), we think of this limitation as a strength, as it leaves little ambiguity as to how to operate a kickable.

On the flipside, kickables—like any type of tangible—are susceptible to misplacement or theft.

We present a simple set of custom kickables, then focus most of this paper on designing sets of standard kickables: five sets, each of which consists of multiple individual widgets for an overall number of 20 different kickables. Kickable sets differ in how they implement constraints and detents, and in how they are being tracked.

Two user studies, one lab-based and one in the wild, showed (1) that kickables suggest how to interact and (2) that kickables offer the affordance that floors with high-precision touch input lack [3].

As a side effect, our work on tracking kickables on a pressure floor contributes a general system of pressure markers, which should be useful beyond the specifics of kickables.

RELATED WORK

This work builds on interactive floors, foot-based interaction techniques, and tangible user interfaces.

Interactive Floors

Interactive floors are part of the ubiquitous computing vision, that is, computers being integrated into the environment, thus becoming invisible [32]. Researchers have explored floors from different perspectives, for example, as part of immersive environments for virtual [9] and augmented realities [23], as unobtrusive sensing platform (e.g., for pose detection [6]), and for interactive installations (e.g., to support collaboration in public space [18] or for gaming [12]). In addition, recent work has proposed precise direct touch manipulation for interactive floors [3].

Input technologies for interactive floors include camera-based tracking [12,18], depth sensing [34], laser range finders and sonar [20], and pressure sensing [3,35]. An alternative to these infrastructure-based sensing approaches is to augment shoes directly [7]. On the output side, systems use steerable [22] or stationary projectors [6,12,18], multi-projector arrays [34], or non-visual, vibrotactile feedback [31]. Our specific kickables are designed for a high-resolution pressure sensing, back-projected floor [6].

Foot-Based Interaction Techniques

Using feet-operated input devices for cursor control goes back to the work by Pearson and Weiser [21]. While feet allow for a wide range of motion, studies reported on their relative coarseness compared to hands and fingers [19]. With the advent of interactive screens, foot-based input has been used to zoom and pan a map [25], or to rotate a canvas [24], complementing hand interaction on the screen.

Researchers have also explored foot-based gestures for hands-free interaction with mobile devices, for example using foot taps [8], kick gestures [13], or rotation and flexion gestures [26]. Similarly, Alexander et al. studied real-world mappings of foot gestures for mobile command invocation, including continuous gestures [1]. In contrast, kickables are designed for foot-based interaction as primary input modality in direct manipulation tasks.

Tangible User Interfaces

Kickables add to the area of tangible user interface [16] by designing for interaction with feet. Just like tangibles for hands, kickables give physical form to digital information to take advantage of haptic interaction skills [16], and to provide tactile feedback, which has been shown to outperform virtual controls [33]. What is more, kickables alleviate limiting factors of direct manipulation with feet (e.g., lower precision [19], lack of multi-digit use, or lack of ergonomic dragging, particularly beyond a leg's reach).

As other tangibles, kickables afford spatial interaction [15]. We designed different kickable sets for different purposes: Bento and tumbler sets provide haptic feedback and decrease demands for visual attention, the former following a token+constraint approach [29]. The remaining three sets are closest in spirit to bricks [11] or general-purpose physical handles [30].

DESIGNING KICKABLE KNOBS

All kickables have a knob—a physical object that forms the only part of the interface that users directly manipulate.

Despite the presumed simplicity of a kickable knob, getting all properties right turned out to be non-trivial, as a knob has to offer not only optimal affordance, but also allow for reliable tracking. We thus went through a series of prototypes and exploratory designs, some of which are shown in Figure 5. A comparison to these initial designs shows that the criteria below are not as obvious as they may appear in hindsight. For example, none of our initial designs offered the now postulated large contact area.

Figure 5: Selected initial designs of kickable knobs we created in the initial stage of the kickables project

Knob Shape

Large contact area. To ease acquisition, kickable knobs should always offer the largest possible contact area—ideally, the entire surface of a kickable knob is thus kickable.

Omni-directional access. Floors allow users to operate contents from any position and any orientation—even more so than tabletop computers. Kickables should therefore expose at least some contact area to all directions.

Resulting knob shapes. We use kickable shapes to afford coarse motion, precise motion, or both. A pointy top, for example, prevents the foot from making careful contact (Figure 6c), while a flat knob cannot be kicked with force (6d). Spherical and cylindrical shapes, in contrast allow for any type of foot interaction (6a-b).

Kicking a non-round kickable can lead to unexpected movement. We will therefore give preference to rotationally symmetric knobs, as shown in Figure 6.

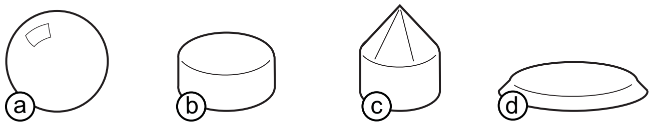

Figure 6: (a-b) Cylindrical and spherical knob shapes afford any interaction. (c) A pointy top discourages use with the foot on top of the kickable. (d) A flat design discourages use with the foot attacking the kickable from the side.

Knob Weight

Knobs have to be designed to offer the right amount of mass (1) to result in a good amount of resistance to the kicking foot to travel the intended distance, and (2) to provide users with a satisfying haptic experience. We found weights between 1kg and 4kg to work well. As a side effect, masses in this range produce sufficient pressure to make them visible to our pressure-sensitive floor.

Friction and Physical Constraints

To allow for precise manipulation, a kickable knob needs to come to rest within the intended motion range. This requires an appropriate amount of friction between knob and ground.

Round kickables roll and are thus typically subject to very low friction. While this makes them suitable for large outdoor installations, on smaller to medium-size installations we need to take measures to increase their friction to a useful range. We achieve this by using a compliant material that causes internal friction as it is constantly deforming while rolling (Figure 7). In order to make it deform, we add weight (1.6kg of lead shot).

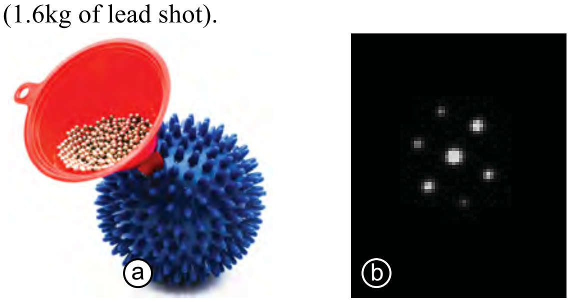

Figure 7: (a) We make spikeball knobs by filling a rubber ball of diameter 10cm with 1.6kg of lead shot using a funnel. (b) The floor sees each kickable as characteristic pressure pattern as a result of the spikes.

Alternatively, we use low-friction materials, adding physical constraints to limit the range of motion (Figure 8a-b).

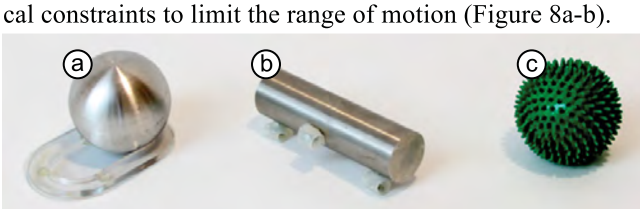

Figure 8: (a-b) Physical constraints limit range of motion for low-friction material such as steel. (c) Compliant material causes internal friction by constantly deforming when rolling.

Physical constraints can serve two purposes:

1. *To limit the motion range* to the represented value range. This provides users with haptic feedback and thus facilitates eyes-free use. We may accomplish this using external objects as constraints (Figure 8a) or by adjusting the shape of the knob itself (8b).
2. *Detents* allow for precise manipulation and limit ranges to discrete values, while again helping with eyes-free use. Detents also avoid ambiguities as they prevent a knob from settling halfway between two states. Again, we may implement this using external constraints objects or by adjusting the shape of the knob itself.

Alternatively, we may implement constraint effects as *soft-constraints*, i.e., by moving screen contents rather than knobs. The benefit of soft-constraints is that they are easier to adapt to different needs by modifying them in software.

In the view of the above design considerations, we now present our five sets of kickable designs. We present them in two groups: soft- and hard-constrained kickables. The soft-constrained kickables use virtual constraints and detents, while the hard-constrained kickables add external objects for physical constraints and detents.

SOFT-CONSTRAINED KICKABLES

We have implemented three sets of what we call *soft-constrained* kickables. The knob design we use for these soft-constrained sets is what we call *spikeball knobs*. As shown in Figure 7a, spikeball knobs are made from a soft but sturdy rubber ball with elastic spikes. We used massage balls of diameter 10cm. As mentioned earlier, we filled them with approximately 1.6kg of lead shot.

The key design element behind spikeball knobs is the spikes. Combined with the extra weight, the spikes create the right amount of deformability that results in the desired amount of friction. This allows users to kick lead-filled spikeballs towards any target in the intended distance range. Unlike other compliant materials we experimented with, such as fabrics, the rubber material maintains its spherical shape, assuring a straight trajectory when kicked. In addition, the properly weighted spikes also produce the characteristic pattern shown in Figure 7b that allows us to reliably recognize spikeball knobs on our pressure-sensing floor.

Spikeball knobs have *x* and *y* location, but no rotation and only limited ID (density of spikes). This allow them to serve as physical handles (similar to bricks [11]), which is sufficient for a wide range of tasks.

Set 1: Tessellation Kickables

Figure 9 shows three tessellation kickables we have implemented based on spikeball knobs. The main design rationale behind tessellation kickables is to create a single basic screenlet that contains all values—and to then repeat it infinitely in a space-filling fashion. In geometry, this process of filling space by repetition is known as tessellation. Tessellation causes the knob to move into another instance of the same basic screenlet pattern. Therefore, input remains valid at all times—also when users overshoot.

Since tessellation makes screenlets infinitely large, each individual screenlet would normally fill the entire screen. To allow multiple tessellation kickables to co-exist, the screenlet renders only the most relevant area, that is, the area surrounding the knob. This generates a "spotlight" effect (inspired by the *spotlight* [17]).

Figure 10 shows how we add soft detents to tessellation kickables, namely by enlarging the respective values. The shown audio slider, for example, helps users turning the audio volume to 100%, 50%, or 0%.

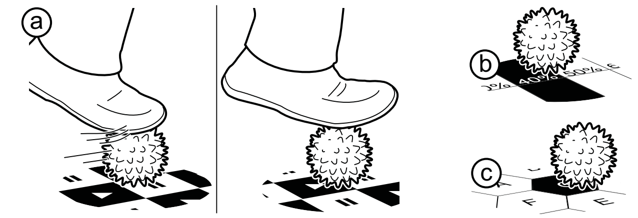

Figure 9: Tessellation kickables indefinitely repeat all values framed by a spotlight. (a) Switching from pause to play by dragging the knob, (b) slider, and (c) choice widget.

The main strength of tessellation kickables is that users can operate them in any direction. This is a desirable property, because users on an interactive floor may be facing *any* direction at any given time, even more so than on tabletops [28].

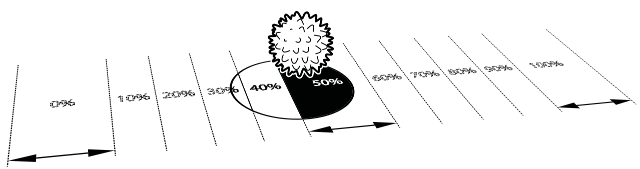

Figure 10: Soft detents help users setting this audio volume slider to 0%, to 50%, or 100% by widening the corresponding regions (indicated by arrows).

Still the tessellation approach has limitations in that it requires careful targeting when switching states to avoid overshooting (i.e., ending up in the original state again), or when aiming at minimum and maximum values—values that tend to occur frequently. Oftentimes, it would be desirable to instead interpret overshooting as a shortcut for switching or "maxing out" rather than wrapping around. We accomplish this with tracking kickables.

Set 2: Tracking Kickables

Tracking kickables are also based on spikeball knobs. Unlike tessellation kickables, the screenlet tracks with the knob whenever the knob reaches the edge of the screenlet (similar to tracking *menus* [10]) as shown in Figure 11.

Tracking kickables subdivide their finite screenlet space and place frequently used values at either end for quick selection by kicking. For example as shown in Figure 12b, users can select either "A" and "E" by overshooting, just as muting or maxing out the volume do not require careful targeting when using the volume slider (12a).

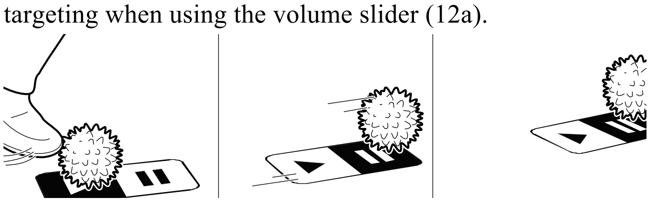

Figure 11: Tracking kickables: The play-pause toggle button tracks with the spikeball when reaching the edge.

Tessellation and tracking kickables are stateful in that their knob encodes the current state through its position. As there is no hovering (i.e., knob positions always indicate values), they do not lend themselves to triggering actions (e.g., "next track"). In contrast, this is the essence of crossing kickables.

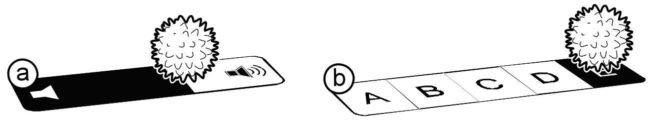

Figure 12: Tracking kickables (a) slider and (b) radio button allow for quickly selecting values at either end by kicking.

Set 3: Crossing Kickables

As shown in Figure 13, crossing kickables are based on the *crossing* interaction technique [2]: not the absolute position of the knob matters, but the act of crossing a specific line.

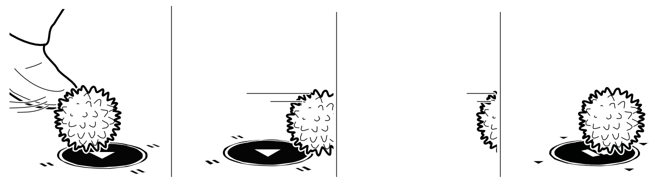

Figure 13: Kicking the knob to cross the line between play and pause; the screenlet resets when the knob stops.

Crossing kickables enable users to operate them in any direction. They consist of one or more infinitely large regions separated by crossing lines (Figure 14). Just as for tessellation kickables, we also use the spotlight metaphor to allow for multiple kickables at the same time.

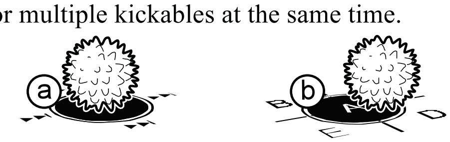

Figure 14: Crossing kickables (a) push and (b) radio button.

As crossing screenlets move the crossing lines to the new knob location after every event, users have to refresh their spatial memory. In addition, this does not allow for adjusting continuous values, as users are required to step through one option at a time.

HARD-CONSTRAINED KICKABLES

Hard-constrained kickables employ physical constraints that prevent their knobs from leaving the space of valid widget values [29]. We use mechanical constraints also to create detents. While the number of physical detents is always fixed, one of the main benefits of hard-constrained kickables is that they provide haptic feedback, which provides better affordance and the potential for eyes-free use. We present two designs: bento and tumbler.

Set 4: Bento Kickables

Bento kickables create physical constraints using an *external* frame, which we call *underlay* (Figure 15a), to prevent leaving the space of valid values. The shown underlays were laser-cut from 8mm transparent acrylic. The actual knob is a featureless solid steel sphere of diameter 10cm (4.1kg), which uses the underlay as a sort of rail track. Figure 15b shows a complete bento kickable with sphere, underlay, and screenlet. We have created four bento kickables as shown in Figure 16.

When our system detects a bento underlay and knob, it creates the corresponding screenlet. Bento kickables allow for determining position and orientation of underlay and knob to align the screenlet with the underlay and to update its state.

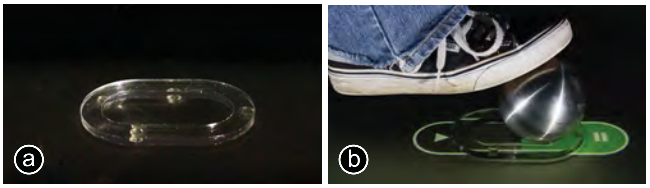

Figure 15: (a) Bento kickables add physical constrains using a system of "rail tracks", which we call *underlays.* (b) The knob is a featureless steel sphere, here shown as part of a switch.

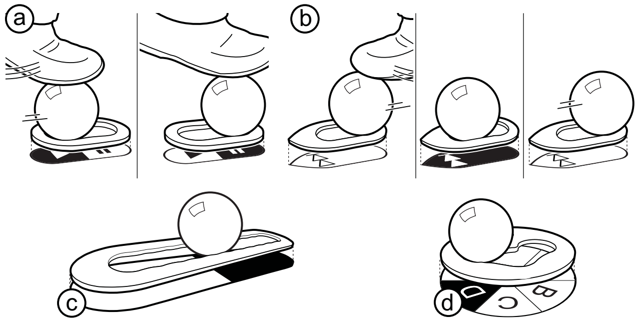

Figure 16: Bento kickables: (a) Pushing to switch to pause (toggle button), (b) kicking to skip to the next track (push button), (c) slider, and (d) radio button.

The floor tracks bento kickables based on a system of point pressure markers arranged in triangles and located at the bottom of the underlays as shown in Figure 17a. The underlay is designed to keep the knob off the ground at all times. Since the knob is resting on the underlay and is sufficiently heavy, it provides each of three contact points with sufficient pressure to make it visible to the pressure-sensing floor. The underlay is compliant enough to limit this pressure to the triangle of contact points surrounding the knob's center of mass; all other marker points remain invisible to the floor (17b). We describe our marker design in detail in section *implementation*.

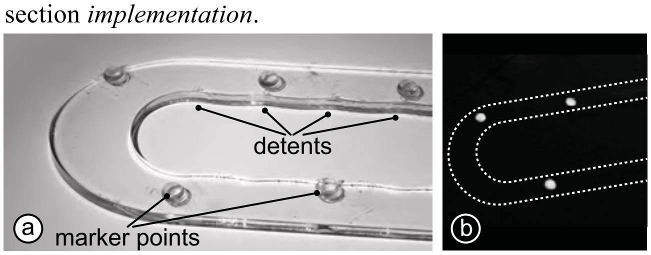

Figure 17: (a) The bottom side of each bento underlay contains pressure marker points, which form a row of triangles. Locally widening the tracks implements detents. (b) Only three marker points support the sphere at a time (outline added).

Underlays also implement detents by locally widening the tracks as shown in Figure 17a. The widening allows the knob to drop a tiny bit, so that gravity pulls the knob into the detent. Since the featureless sphere does not provide sufficient friction against the acrylic underlay, it bounces back when reaching the resting state of a push button. We thus widened end states for the sphere to go deeper and stop.

The main limitation of bento kickables is the underlays; they add clutter to the setup and need to be transported separately from the knob. We address this with our *tumbler* design.

Set 5: Tumbler Kickables

A somewhat simplistic way of describing the rationale of tumbler kickables is to see them as the result of wrapping a bento underlay around a bento knob. As a result, the marker points, constraints, and detents are now part of the knob itself, eliminating the necessity for a separate underlay. Figure 18 shows a user pushing a slider from the tumbler kickable set.

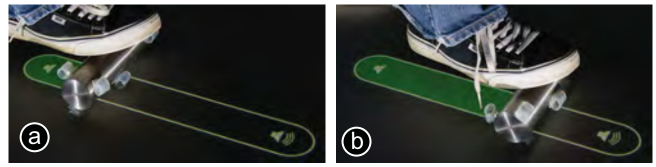

Figure 18: Adjusting volume using a tumbler slider: Marker points, constraints, and detents are part of the knob itself.

Tumbler kickables can be made in a wide variety of form factors. The design shown in Figure 19 reduces the knob to the bare minimum: It uses spheres of diameter 8cm (2kg) as well as 18cm long cylinders of diameter 5cm (2.8kg) for *cores*, and M6 threaded rods with silicone caps for *spokes*. The spoke's role is twofold: (1) they form marker triangles and (2) implement constraints and detents. For example, adding a longer spoke to the slider implements a constraint that prevents infinite rolling, limiting its range.

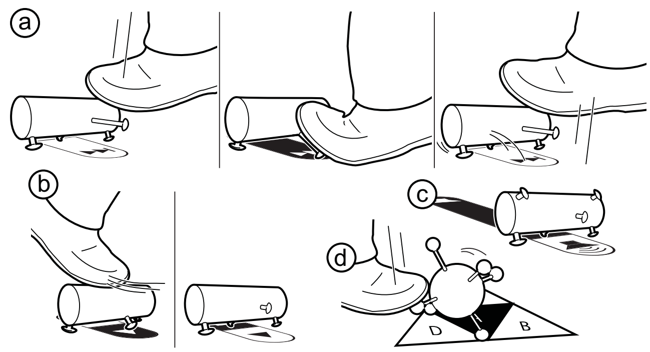

Figure 19: Tumbler kickables: (a) Skipping a track (push button), (b) switching from play to pause (toggle button), (c) volume slider with detents, and (d) 2D radio button.

As with bento kickables, each tumbler is designed to render state transitions as weight shifts of the center of gravity across different marker triangles. With the tumbler push button shown in Figure 20, for example, the core is by default supported by a triangle of spokes. Users pushing down on the long activation spoke lift up marker point M1 and instead cause marker point M2 to touch the floor, resulting in a different triangle touching the ground.

The main limitation of tumbler kickables is that detents are inherent to their design. However, packing spokes tighter to increase the number of sides, and hence detents, alleviates this to some extent. In contrast to bento kickables, tumbler kickables can move without limitations on the floor.

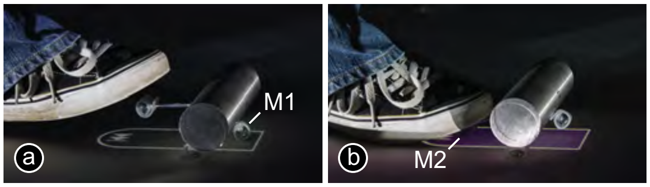

Figure 20: Operation of a tumbler push button kickable: Pushing down causes M1 to disappear and M2 to appear instead, resulting in a different triangle touching the ground.

Summary of Kickable Sets

In order to help application designers create interfaces based on kickables, we have presented five sets of standard kickables, each of which addresses our design requirements in a different way, as summarized by Table 1.

	soft-constrained			hard-constrained	
	tessel.	**tracking**	**crossing**	**bento**	**tumbler**
knob	lead-filled spikeball			steel ball	steel cylinder or ball
constraints	space-filling screenlet repetition	screenlet tracks along with knob	knob is positioned relative to line	underlays as tracks	long spokes
detents	enlarging regions	enlarging regions	inherent to design	widening tracks	short spokes

Table 1: Summarizing properties of the five sets

IMPLEMENTATION

In this section, we focus on our specific implementation of kickables made for pressure-sensing floors. We then introduce an alternative tracking approach for spikeballs.

System Architecture

Figure 21 shows the three layers of our system, composed of (1) floor hardware, (2) tracking framework, and (3) widget toolkit. Using computer vision algorithms, the tracking framework recognizes and tracks kickable knobs based on their pressure imprints. Based on this, it outputs events to notify the widget toolkit of user interactions, such as moving a knob. The toolkit, in turn, provides a set of kickable screenlets and manages their coupling with knobs.

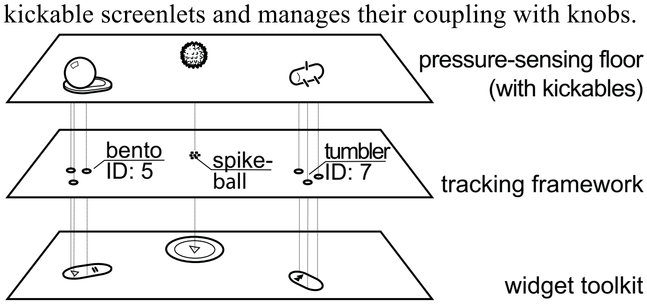

Figure 21: Kickables system layers

The floor consists of an 8m² interaction surface in a single seamless piece (see Figure 5). It uses FTIR for pressure sensing at a resolution of 1×1mm, and a 4K back projection for output. We implemented two orthogonal tracking approaches: (1) using dedicated pressure markers that encode identifier and rotation for bentos and tumblers, and (2) a simpler approach solely based on imprint textures for spikeballs.

Pressure-Marker System for Bentos and Tumblers

Pressure markers consist of individual marker points arranged in triangles, and are defined through the distances between these points.

Marker Layout. Pressure markers have been previously explored, for example to track furniture [6] or detect the amount of pressure exerted by users on tangibles [14]. We have the additional requirement of detecting the quasi-continuous movement of the knob. Figure 22 shows how we address this by creating marker sequences in the form of triangle strips using the example of bentos.

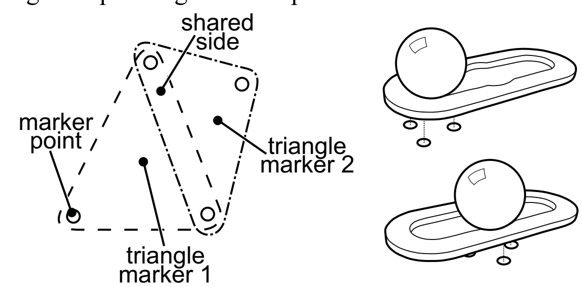

Figure 22: Marker sequences (i.e., triangle strips) encode widget states; adjacent markers share two marker points.

Each triangle represents one state in a widget (e.g., the value of a slider or the state of a switch). Any two adjacent markers share a pair of marker points. As the bento knob moves across the underlay, its center of pressure moves from one triangle to the next, causing one marker point to disappear and another marker point to appear. In case of tumblers, turning over the knob produces the same effect. This approach requires $n + 2$ marker points in order to distinguish n locations.

Marker Space. We generate unique markers by representing each triangle marker of points A, B, and C as a triplet of distances (i.e., AB, BC, CA). To identify markers independent of their rotation, we discard triangles that can be created by rotating another triangle. We also discard equilateral triangles, because they do not allow determining the rotation of the kickable.

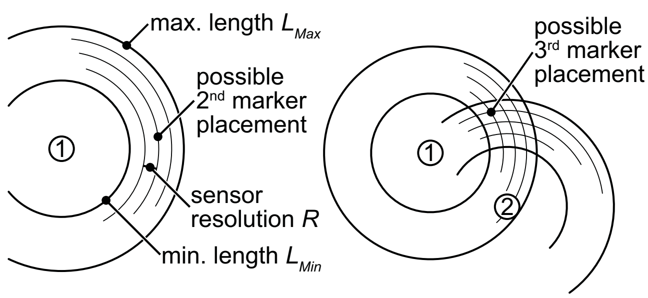

Figure 23: Three factors determine the available marker space: (1) minimum and (2) maximum side lengths L_{Min} and L_{Max}, and (3) sensor resolution R.

Figure 23 illustrates the three factors that determine the available marker space: (1) minimum and (2) maximum triangle side lengths, L_{Min} and L_{Max}, and (3) sensing resolution R. L_{Min} and L_{Max} are governed by the kickables' shape and size, while R affects the minimum difference between lengths to still be distinguishable.

It follows that there are $l = [(L_{Min} - L_{Max}) / R] + 1$ available triangle side lengths. Consequently, the number of unambiguously identifiable triangle markers is given by $m = (l^3 - l) / 3$, where l^3 stands for all combinations of triangle lengths, $-l$ removes equilateral triangles, and dividing by 3 eliminates circular shifts. Our bento implementation, for example, uses L_{Min} = 5.5cm, L_{Max} = 10.5cm, and R = 0.5cm, hence l = 11, allowing for 440 rotation-invariant markers.

Marker Identification. Triangle markers are registered in a database with their defining side lengths as well as the corresponding knob and state. For identification, the tracking framework first detects individual marker points based on their distinct shape, and clusters them into potential marker triangles. Next, for each triangle, it calculates the squared difference of side lengths compared to all markers stored in the database, and picks the marker with the lowest difference (i.e., only if the difference stays below a predefined threshold, to reject unknown markers or noise).

Tracking Spikeballs Based on Imprint Textures

To track spikeballs, we use an approach solely based on their discernible imprint texture (see Figure 7b), which allows for localizing the knob, but does not encode rotation or any unique identifier. The tracking framework extracts 16 fast-to-compute image features as input to a feed-forward neural network for classification, an approach successfully applied to classifying body contacts [6].

Widget Toolkit

Our widget toolkit is event-driven and abstracts from the underlying tracking approach. It provides a set of kickable widgets (e.g., switches or sliders), handles the coupling with knobs, and updates internal states (e.g., slider values) based on knob movements. In doing so, it adopts the appearance of screenlets depending on the type of knob placed on top.

This implementation is based on the Qt GraphicsView framework to allow for fast, cross-platform prototyping of applications. All screenlets are represented as items in a 2D scene graph and can be customized, decorated, or animated using QML. Application developers may add new screenlets by providing designs and defining behaviors.

Alternative Top-Camera Tracking for Spikeballs

Figure 24 shows an alternative tracking solution we created in order to track spikeballs on arbitrary surfaces. It is based on overhead IR cameras and projectors. It uses the same widget toolkit as described above, so that applications written once run without modification. However, we surrounded the spikeballs with thin retro-reflective stripes in order to make them visible to the IR camera. We apply standard computer vision algorithms to identify and track spikeballs.

As spikeballs always stay in the same 2D plane, i.e., just above the plane of the floor, a homography suffices to calibrate cameras against the projection. The fact that kickables move in a 2D plane also makes it easy to add additional cameras, which we may do in order to reduce occlusion and cover larger floor areas by fusing observations from multiple angles.

Figure 24: The IR camera tracks spikeballs with retro-reflective tape using standard computer vision algorithms.

USER STUDY I: KICKABLES SUGGEST HOW TO INTERACT

To validate our approach we conducted two user studies. The purpose of this first, lab-based study was to validate that kickables indeed suggest how to interact and that it frees users up to walk across the display area. We recruited 20 participants (4 female) between 18 and 28 years old. They participated in both parts of this two-task lab-based study. Given the nature of the two tasks, we had *all* participants perform the walk-up task first.

Walk-Up Task

We asked participants to enter certain values without prior training into the study interface shown in Figure 25a. It consists of a video player and three kickable controls from the tracking set: a play-pause switch (left), a volume slider (center), and a playlist with four items (right). In counterbalanced order, we asked participants to pause and resume playback, set the volume to 75%, and pick a specific track from the playlist—using any interaction they saw fit.

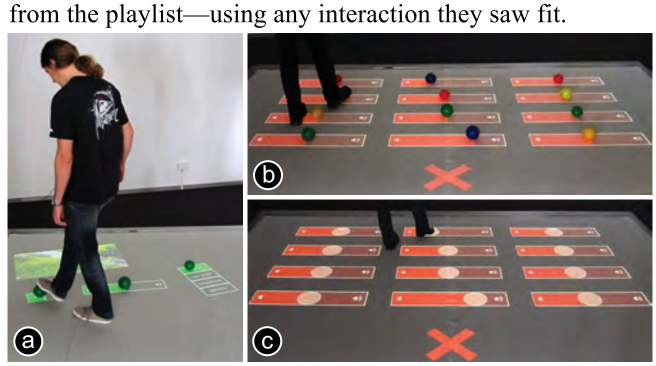

Figure 25: (a) ***Walk-up task:*** **We asked participants to enter certain values into this interface without prior training. (b)** ***Non-interaction task:*** **Participants realized that stepping on kickable sliders was ok, (c) while they had to avoid stepping on sliders in the touch condition.**

Results: All participants successfully completed the three study tasks. 90% did so at the first attempt, i.e., the first operation was to correctly operate the right kickable. Two participants, however, tried touch input first, and only after this failed they tried to (correctly) operate the knobs.

To our surprise, only nine participants operated kickables using their feet, while eleven participants used their hands. While our designs support both, we did not expect that many to use their hands first. When we asked participants about their preferences, three unsurprisingly explained that they were more familiar with hand interaction; an aspect any new technology has to face. Another two participants were worried about touching lab equipment using their dirty shoes. Besides, intuitively using hands instead of feet may also be rooted in cultural norms.

Non-Interaction Task

Touch interfaces of reasonable complexity can make it hard for users to find empty space to navigate [3]. Kickables are less susceptible to this as users are free to walk on the screenlet itself; only the knobs are to be avoided. The purpose of this second task was to investigate if participants understood this aspect and if it helped them moving around more freely.

We did so by making participants cross an interface that was densely populated with controls—either kickable controls (Figure 25b) or touch controls (25c). We then observed whether they dared walking across the controls.

To this end, we first demonstrated the interface by adjusting a slider in the bottom-most row, which we could without stepping on the interface, and instructed participants to try a different slider in the same row. We then asked participants to adjust two specific sliders, all of which were located in the top or middle rows. We logged how much their steps overlapped the sliders as they performed the task.

We tested kickable and touch conditions in counterbalanced order. To reduce sequence effects, we presented participants with a distractor task between conditions (i.e., we asked participants to answer questions about an unrelated video).

Results: In the touch condition, participants placed their feet very carefully to avoid stepping on sliders (average overlap of M=28.26cm^2, SD=25.98). In the kickable condition, in contrast, participants stepped on the sliders with much more confidence—causing their feet to intrude on significantly more slider area (M=74.88cm^2, SD=62.44; $t(18)=3.285$, $p<0.05$). These results suggest that users had understood the underlying interaction concept of kickables, which allowed them to roam around more freely.

USER STUDY II: KICKABLES DEPLOYED IN THE WILD

The goal of this study was to assess kickables' ability to attract users in settings outside the lab. We set up a simple installation with a single widget and counted how many passersby tried to interact with it. To help interpreting the results, we added a touch condition as a baseline. However, given the higher familiarity with touch, we suspected kickables to attract fewer people than touch.

Interface and Task: Figure 26 shows the interactive floor installation that we set up in the entrance halls of two on-campus cafeterias. The single interactive widget they offered was a volume slider located at the bottom of the interface. To justify its presence, we added a music video player above. The widget was either touch or kickable, depending on condition. We tracked kickables using the top-camera tracking setup described in the *implementation* section. We tracked touch using a Kinect camera. The installation was set up for a total of 2:10h for kickables and 2:10h for touch. No instructions were provided.

Results: Out of 355 passersby, 23 tried to interact with kickables (6.5%); out of 341 passersby, 19 tried to interact with the touch widget (5.6%). A χ^2 test found no significant difference.

Figure 26: In-the-wild study installation in the entrance hall of the on-campus cafeteria using (a) kickables or (b) foot touch

Considering the confound between familiarity and affordance, one way of interpreting the results is that kickables were making up for their lack of familiarity with better affordance.

Encouragingly, all passersby who interacted with kickables did so correctly right away (i.e., by *moving* the spikeball along the volume slider). Again, we found only 70% of participants to use their feet to interact, while the remaining seven used their hands. When we asked them about their reason, they again pointed to concerns about their dirty shoes and worry about damaging the equipment.

To extend this initial understanding of in-the-wild kickable interaction, we suggest studying large-scale deployments with dozens of simultaneous users next. This will shed light on issues possibly arising from prolonged multi-user interaction, such as the disarrangement of kickable knobs or accidental activation. Surrounding exhibits with low barriers, for example, addresses the problem of disarranging spikeballs. While we have not observed accidental activation in the presented studies, we expect users to recover quickly, as the state of any kickable is permanently visible.

DISCUSSION & CONCLUSIONS

In this paper, we presented kickables, tangibles for feet. The biggest strength of kickables is their potential to bring functionality *and* affordance to very large interaction spaces, such as exhibition halls or outdoor installations, as we validated in two user studies.

We presented a custom design, which we had made for a hypothetical chemistry exhibit, but focused primarily on creating standard kickables to support the developers of kickable application. Each member of our five sets of kickables addressed the design requirements in a different way, for an overall number of 20 individual kickable designs. While the presented kickables were engineered to work with a pressure-sensing floor, we also demonstrated an alternative top-tracking solution for arbitrary surfaces.

While we focused on general-purpose designs for generic UI controls, the general concept of kickables comprises a much larger design space. In the future, as the past has shown for hand tangibles, we expect to see a large number of *special-purpose* designs. These designs will naturally come in a wide variety of shapes and materials. The kickable concept does allow for a wide range of embodiments, from the basic geometric shapes we presented in this paper to self-actuating robotic knobs. We also plan to explore a wider range of tracking and output technologies, from sensors embedded in the knobs to augmented reality displays.

ACKNOWLEDGEMENTS

We thank Microsoft Research Cambridge for their support.

REFERENCES

1. Alexander, J., Han, T., Judd, W., et al. Putting Your Best Foot Forward: Investigating Real-World Mappings for Foot-Based Gestures. In *Proc. CHI'12*, 1229–1238.
2. Apitz, G. and Guimbretière, F. CrossY: A Crossing-Based Drawing Application. In *Proc. UIST'04*, 3–12.
3. Augsten, T., Kaefer, K., Meusel, R., et al. Multitoe: High-Precision Interaction with Back-Projected Floors Based on High-Resolution Multi-Touch Input. In *Proc. UIST'10*, 209–218.
4. Bannon, L., Benford, S., Bowers, J., et al. Hybrid Design Creates Innovative Museum Experiences. *Comm. ACM 48*, 3 (2005), 62–65.
5. Block, F., Wigdor, D., Phillips, B.C., et al. FlowBlocks: A Multi-Touch UI for Crowd Interaction. In *Proc. UIST'12*, 497–508.
6. Bränzel, A., Holz, C., Hoffmann, D., et al. GravitySpace: Tracking Users and Their Poses in a Smart Room Using a Pressure-Sensing Floor. In *Proc. CHI'13*, 725–734.
7. Choi, I. and Ricci, C. Foot-Mounted Gesture Detection and Its Application in Virtual Environments. In *Proc. Systems, Man, and Cybernetics'97*, 4248–4253.
8. Crossan, A., Brewster, S.A., and Ng, A. Foot Tapping for Mobile Interaction. In *Proc. BCS HCI'10*, 418–422.
9. Cruz-Neira, C., Sandin, D.J., and DeFanti, T.A. Surround-Screen Projection-Based Virtual Reality: The Design and Implementation of the CAVE. In *Proc. SIGGRAPH'93*, 135–142.
10. Fitzmaurice, G., Khan, A., Pieké, R., et al. Tracking Menus. In *Proc. UIST'03*, 71–79.
11. Fitzmaurice, G.W., Ishii, H., and Buxton, W.A.S. Bricks: Laying the Foundations for Graspable User Interfaces. In *Proc. CHI'95*, 442–449.
12. Grønbæk, K., Iversen, O.S., Kortbek, K.J., et al. iGameFloor - A Platform for Co-Located Collaborative Games. In *Proc. ACE'07*, 64–71.
13. Han, T., Alexander, J., Karnik, A., et al. Kick: Investigating the Use of Kick Gestures for Mobile Interactions. In *Proc. MobileHCI'11*, 29–32.
14. Hennecke, F., Berwein, F., and Butz, A. Optical Pressure Sensing for Tangible User Interfaces. In *Proc. ITS'11*, 45–48.
15. Hornecker, E. and Buur, J. Getting a Grip on Tangible Interaction: A Framework on Physical Space and Social Interaction. In *Proc. CHI'06*, 437–446.
16. Ishii, H. and Ullmer, B. Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms. In *Proc. CHI'97*, 234–241.
17. Khan, A., Matejka, J., Fitzmaurice, G., et al. Spotlight: Directing Users' Attention on Large Displays. In *Proc. CHI'05*, 791–798.
18. Krogh, P.G., Ludvigsen, M., and Lykke-Olesen, A. "Help Me Pull that Cursor" - A Collaborative Interactive Floor Enhancing Community Interaction. In *Proc. OZCHI'04*.
19. Pakkanen, T. and Raisamo, R. Appropriateness of Foot Interaction for Non-Accurate Spatial Tasks. In *Proc. CHI EA'04*, 1123–1126.
20. Paradiso, J.A. and Hu, E. Expressive Footwear for Computer-Augmented Dance Performance. In *Proc. ISWC'97*, 165–166.
21. Pearson, G. and Weiser, M. Of Moles and Men: The Design of Foot Control for Workstations. In *Proc. CHI'86*, 333–339.
22. Pinhanez, C. The Everywhere Displays Projector: A Device to Create Ubiquitous Graphical Interfaces. In *Proc. UbiComp'01*, 315–331.
23. Raskar, R., Welch, G., Cutts, M., et al. The Office of the Future: A Unified Approach to Image-Based Modeling and Spatially Immersive Displays. In *Proc. SIGGRAPH'98*, 179–188.
24. Sangsuriyachot, N., Mi, H., and Sugimoto, M. Novel Interaction Techniques by Combining Hand and Foot Gestures on Tabletop Environments. In *Proc. ITS'11*, 268–269.
25. Schöning, J., Daiber, F., Krüger, A., et al. Using Hands and Feet to Navigate and Manipulate Spatial Data. In *Proc. CHI EA'09*, 4663–4668.
26. Scott, J., Dearman, D., Yatani, K., et al. Sensing Foot Gestures from the Pocket. In *Proc. UIST'10*, 199–208.
27. Seto, M., Scott, S., and Hancock, M. Investigating Menu Discoverability on a Digital Tabletop in a Public Setting. In *Proc. ITS'12*, 71–80.
28. Shen, C., Ryall, K., Forlines, C., et al. Informing the Design of Direct-Touch Tabletops. *IEEE Comp. Graph. and App. 26*, 5 (2006), 36–46.
29. Ullmer, B., Ishii, H., and Jacob, R.J.K. Token+Constraint Systems for Tangible Interaction with Digital Information. *ACM TOCHI 12*, 1 (2005), 81–118.
30. Ullmer, B. and Ishii, H. The metaDESK: Models and Prototypes for Tangible User Interfaces. In *Proc. UIST'97*, 223–232.
31. Visell, Y., Law, A., and Cooperstock, J.R. Touch Is Everywhere: Floor Surfaces as Ambient Haptic Interfaces. *IEEE Trans. Haptics 2*, 3 (2009), 148–159.
32. Weiser, M. The Computer for the 21st Century. *Scientific American 265*, 3 (1991), 94–104.
33. Weiss, M., Wagner, J., Jansen, Y., et al. SLAP Widgets: Bridging the Gap Between Virtual and Physical Controls on Tabletops. In *Proc. UIST'09*, 481–490.
34. Wilson, A.D. and Benko, H. Combining Multiple Depth Cameras and Projectors for Interactions On, Above and Between Surfaces. In *Proc. UIST'10*, 273–282.
35. Yin, K. and Pai, D.K. FootSee: An Interactive Animation System. In *Proc. SCA'03*, 329–338.
36. Epidémik. http://www.cite-sciences.fr/epidemik, Aug. 14th, 2013.