Multitoe: High-Precision Interaction with Back-Projected Floors Based on High-Resolution Multi-Touch Input

Thomas Augsten, Konstantin Kaefer, René Meusel, Caroline Fetzer, Dorian Kanitz, Thomas Stoff, Torsten Becker, Christian Holz, and Patrick Baudisch
Hasso Plattner Institute, Potsdam, Germany
{firstname.lastname}@student.hpi.uni-potsdam.de
{christian.holz, patrick.baudisch}@hpi.uni-potsdam.de

ABSTRACT
Tabletop applications cannot display more than a few dozen on-screen objects. The reason is their limited size: tables cannot become larger than arm's length without giving up direct touch. We propose creating direct touch surfaces that are orders of magnitude larger. We approach this challenge by integrating high-resolution multi-touch input into a back-projected floor. As the same time, we maintain the purpose and interaction concepts of tabletop computers, namely direct manipulation.

We base our hardware design on frustrated total internal reflection. Its ability to sense per-pixel pressure allows the floor to locate and analyze users' soles. We demonstrate how this allows the floor to recognize foot postures and identify users. These two functions form the basis of our system. They allow the floor to ignore users unless they interact explicitly, identify and track users based on their shoes, enable high-precision interaction, invoke menus, track heads, and allow users to control high-degree of freedom interactions using their feet. While we base our designs on a series of simple user studies, the primary contribution on this paper is in the engineering domain.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces: Input Devices and Strategies, Interaction Styles.

Keywords: Interactive Floor, Multi-touch, FTIR, Front DI, Direct Manipulation, Tabletop, Projection.

General terms: Design, Human factors.

INTRODUCTION
Direct touch technology has spawned a new generation of display form factors [10], in particular tabletop computers [7]. Unfortunately, the nature of *direct touch* limits the size of these systems: contents can only be touched if located within *arm's reach*. On tables larger than arm's length, users can reach contents only by artificially extending their reach [11]. In order to preserve the direct touch concept, current tabletop makers have opted to create coffee table-sized devices (e.g., 30' display in *Microsoft Surface* [15] and a 27' display in the *Smart Table* [31]).

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.

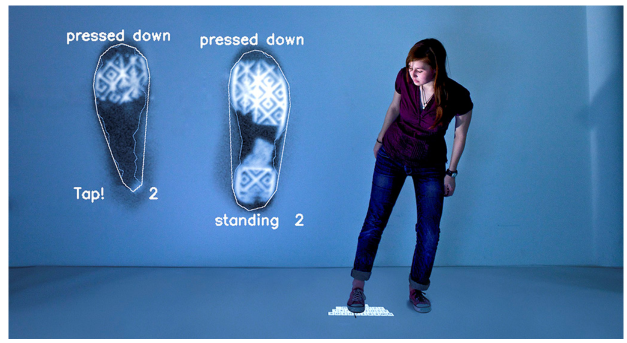

Figure 1: Integrating high-resolution FTIR into a back-projected floor allows the floor to see the pressure distribution of the user's soles (inset top left, as seen from below). In the shown situation, the floor ignores the foot on the right based on its posture, yet allows the foot on the left to interact. By identifying the user based on her sole patterns, the floor has attached a user-specific high-precision pointer to her foot, which allows her to operate tiny controls, here a keyboard.

We argue that the size constraints of tabletops have limited the discussion about what can be done on horizontal surfaces to what fits the format. What about applications were users interact with *thousands or ten-thousands* of on-screen objects, such as complex visual sensemaking applications?

We propose direct touch surfaces that are orders of magnitude larger than tables by integrating high-resolution multi-touch technology into back-projected floors. Unlike table users that stand along the table's *perimeter*, floor users walk *across* these surfaces, allowing them to reach any part of the floor—independent of the size of the installation.

In order to enable direct manipulation on floors, we base our design on *frustrated total internal reflection* (*FTIR*) [10] with high camera resolution (Figure 1). Unlike earlier floor installations that achieved display size at the expense of input resolution (e.g., [5, 13]), we demonstrate how the use of FTIR allows us to maintain the direct manipulation interaction model of tabletop systems, despite the dramatically different size (Figure 1).

The ability of FTIR to sense per-pixel pressure also provides our *multitoe* floor with new interactive capabilities, such as user identification and head tracking interaction.

OVERVIEW: DIRECT MANIPULATION ON FLOORS
As suggested by its name, the interaction concept of tabletop, i.e., direct *mani*-pulation was designed with hands in

mind. The adaptation to interactive floors results in a series of challenges that we tackle in this paper:

1. Users stand on floors. We address inadvertent activation by making the floor ignore but input unless users demonstrate a specific foot posture (Figure 2a).

2. Distances on floors are potentially very large. We address this with location-independent pop-up menus that users invoke by jumping (Figure 2b).

3. To allow for a consistent interaction model, the floor needs to know which parts of their feet/shoes users expect to be "active". We conducted a simple study in which most participants expected not just the contact area, but the entire *projection* of their shoes to actively trigger interactions.

4. Feet are roughly 200x times larger than fingertips and less precise. When necessary, we offer a high-precision mode that condenses a user's foot into a single "hotspot" (Figure 2c). Since users disagree about the location of this hotspot, we allow them to customize its location. To enable personalization, the floor recognizes users based on their sole patterns (Figure 2d).

Finally, we take a closer look at algorithms and at the additional functionality enabled by FTIR floors: how to track users' heads based on the pressure distribution in their soles (Figure 2e), and how to enable high-degree of freedom interaction (Figure 2f).

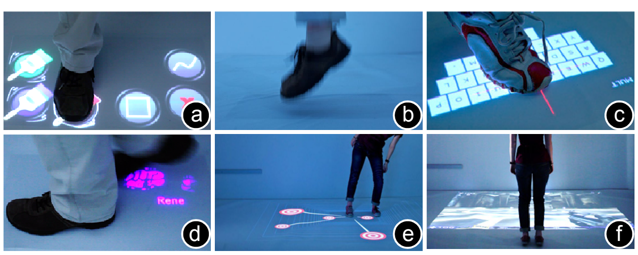

Figure 2: (a) Users trigger interactions with a specific foot posture, (b) invoke menus by jumping, (c) interact precisely using a hotspot, which is (d) enabled by sole-based user recognition. (e) FTIR-based tracking also allows controlling applications using body posture and (f) foot posture.

Contribution

This paper makes two contributions. (1) We enable precise direct manipulation on interactive floors. We present a series of techniques designed to overcome the inherent uncertainties of floors, in particular inadvertent activation, lack of modes, and size of feet. (2) We conduct an exploration of interaction techniques enabled by bringing FTIR to interactive floors. The ability of FTIR to recognize pressure in high spatial resolution allows us to recognize users, determine body postures, and approximate head tracking.

The primary contribution on this paper is therefore in the *engineering* domain. While we conduct a series of simple studies, the purpose of these studies is to help us make *design decisions*, rather than validate the system or test a scientific hypothesis (see also the discussion in [8]).

RELATED WORK

The work presented in this paper builds on interactive floors, techniques and technology from touch and multi-touch, user identification in ubiquitous computing, and high-precision interaction.

Interaction using shoes and floors

Floor interaction can be accomplished by instrumenting either shoes or the floor. The shoes by Choi and Ricci detect walking direction and speed using buttons mounted under users' soles [4]. They used the shoes for artistic performances and movement training. By adding bending, twisting, orientation, acceleration, and pressure sensors, Paradiso et al. gave dance performances extra expressiveness [22]. Shirai at al's *Fantastic Phantom Slipper*s are tracked optically [30]. Visell et al. used optical tracking as well [33]. Paradiso et al. tracked shoes using laser rangefinders [20] and active sonar (*magic carpet* [21]).

Instrumented floors use ceiling-mounted cameras (e.g., *iFloor* [13]), a technique that came out of Ubicomp environments (e.g., *EasyLiving* [2]). *IGameFloor* tracked uses using *front diffuse illumination* [9].

Unlike the work presented in this paper, interactive floors so far have been used for natural walk-up-and-use [13], immersion (also as part of *CAVEs* [5, 14]), gaming [9], and multi-user collaborative applications [10]. Paradiso argues that for many types of floor interaction "…fine-grained information delivered by a video camera is unnecessary or potentially inadequate" [21]. *IFloor* allowed for Text entry, however, via mobile phone [13].

Sensing the pressure between feet and floor

In this paper, we use FTIR to obtain pressure information at a level of resolution high enough to recognize sole patterns. In the past, the pressure abilities of FTIR have been studied in the context of wall displays [6]. A series of floor prototypes have used (low-resolution) pressure sensing. The projection-less *magic carpet* senses pressure using piezoelectric wires and a pair of Doppler radars [21]. *Z-tiles* improved on this by introducing a modular system of interlocking tiles [28]. Pressure sensing has been implemented using force-sensing resistors [33]. In the desktop world, the *UnMousePad* improves on resistive pressure sensing by reducing the number of required wire connections [29].

Identifying users

The majority of large-scale touch technologies, such as *diffuse illumination* (aka *DI* [38]), *front-DI* [9], and *FTIR* [10] are ignorant of who touches. *DiamondTouch* improved on this by mapping users to seat positions [7]. Wang et al. obtained similar information by analyzing finger orientation [36]. Recognizing fingerprints has been envisioned to identify users of touch systems [32]; Holz and Baudisch implemented this by turning a fingerprint scanner into a touch device [12]. User identification has been used for a variety of applications including the management of access privileges [16] and to help kids with Asperger syndrome learn social protocol [26].

The screen-less *Smart Floor* identifies users by observing the forces and timing of the individual phases of walking [19]. While floors so far did not have enough resolution to distinguish soles, footprints have been analyzed as evidence in crime scene investigation [23]. Sole imprints and sole wear has been used to match people either by hand and using semi-automatic techniques based on local feature extractors, such as *MSER* [23, 24].

High-precision touch input

Feet have a comparably large extent, which complicates pointing techniques that expect a single point of contact. In the context of touch, this has been referred to as the *fat finger problem* [35]. It has been alleviated by separating target and finger to avoid occlusion (e.g., *offset cursor* [27], *shift* [35]) and by calibrating touch on per-user basis [12]. Vogel et al proposed avoiding rendering contents in occluded areas in the first place [34]. Instead of reducing fingers to a point, Cao et al. proposed touch interaction based on the entire contact area (*ShapeTouch* [3]).

PROTOTYPE HARDWARE

As mentioned earlier, most of the functionality of our floor design is enabled by FTIR, more specifically FTIR combined with front diffuse illumination. In this section, we explain the technology in additional detail and juxtapose it to other technologies we have tried.

Tracking using Front DI + High-resolution FTIR

We initially experimented with traditional rear-diffuse illumination. While we found it to work well with light soles, it produced no effect when users wore shoes with black soles (Figure 3a). *Front diffuse illumination*, in contrast, is ignorant of shoe color, as it is based on tracking the *shadows* casts by shoes [9]. We therefore included front-DI in our floor design, which gives us a rough outline of the user's shoes (Figure 3a).

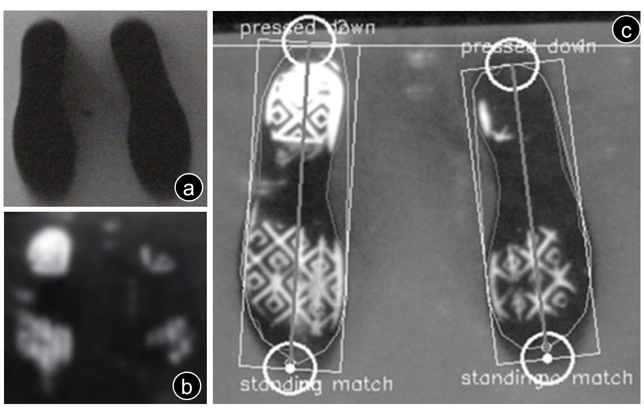

Figure 3: A user wearing shoes with black soles is standing on our floor prototype. (a) Front DI, (b) FTIR, (c) Front DI with FTIR, at 1mm resolution camera.

The key step was to add FTIR. The main benefit of FTIR in our application scenario is that it makes pressure visible, as was explored earlier in the context of wall displays [6]. As illustrated by Figure 3c bringing FTIR to floors reveals weight distributions; here we see that the foot on the left bears most of the weight, while the foot on the right does not, indicating basic properties of the user's posture.

Unlike other floor designs, we use a camera the resolution of which is comparable to the resolution offered by multi-touch *tables*, i.e., a pixel size of 1.0mm (comparable to *Microsoft Surface* [15]). As illustrated by Figure 3c, this reveals the next level of structure inside the sole, such as patterns and logos of the shoe manufacturer. These elements are crucial for recognizing users (see Section "Algorithms").

Materials & prototype

Figure 4 shows the stack-up of our floor surface. A three-layer 3.4cm/1.34" glass pane provides structural support, an 8mm layer of acrylic serves as the waveguide, and a *Rosco* projection screen creates the image (http://rosco.com). Between waveguide and screen we use a layer of *Tectosil 185 500µm* silicone as compliant surface (http://wacker.com). As with all FTIR devices [10], a compliant surface helps obtain a well-defined amount of frustration for a given amount of pressure. *Tectosil 185* is a stiffer silicone, which allows us to distinguish pressure at the upper end of the scale, e.g., to distinguish a user resting the entire weight on the ball of one foot from a user standing straight.

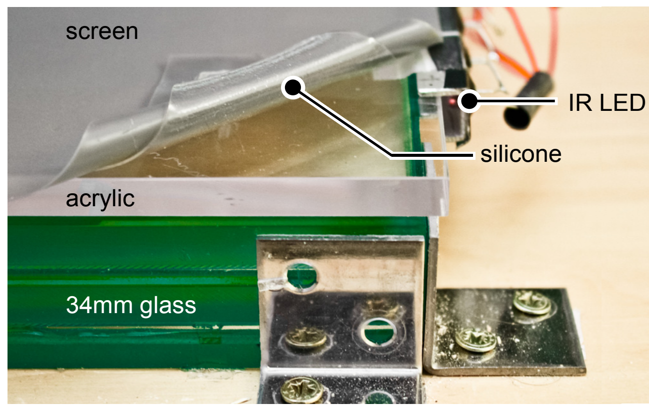

Figure 4: We use 34mm safety glass, 8mm acrylic, *Tectosil 185* silicone, and *Rosco* projection screen.

In order to keep material expenditure reasonable during the exploration phase, FTIR input on this prototype is limited to a small sub-region (70cm x 50cm, see Figure 5). The floor's projection resolution is 0.6mm per pixel; the camera resolution is 1.0mm/pixel.

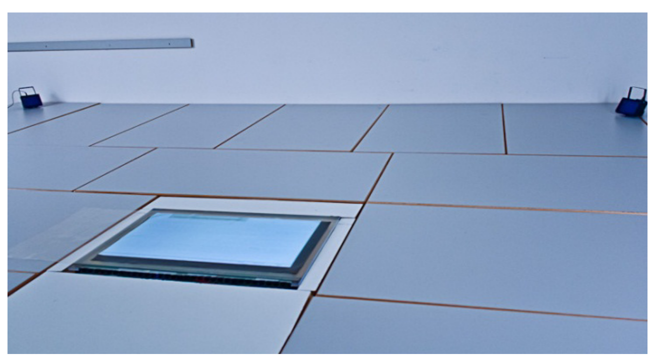

Figure 5: (a) On our current working prototype only a sub-region is interactive (projection screen, compliant surface and one floor tile removed).

1. RESOLVING INADVERTENT ACTIVATION

Unlike tabletop devices, floor users are in contact with the floor at basically all times. To enable direct manipulation, we need a mechanism that distinguishes between intentional action (the analog to touch on tabletop) and standing/walking (the analog to hover on tabletop).

For most foot-operated devices, this distinction is handled spatially. Users step onto the gas to accelerate; in order to not accelerate they rest their foot elsewhere. We can port the same concept to our floor design by inserting pathways of touch-insensitive areas between controls. Unfortunately, this prevents us from using large controls, such as the painting surface of a painting program. We thus need a gesture that allows users to not interact even though they are standing *on* a control.

Several alternative designs seem possible: users could jump onto a button to activate it or stomp on it, etc. Not all of them are equally ergonomic though and it is unclear how intuitive they are. To find out what works for users we conducted a simple user study.

User study: how to *not* activate a button

The purpose of this study was to help design a mechanism that allows the floor to distinguish intentional user action from regular walking and standing and that also matches users' intuition. Participants' task was to walk across four "buttons" such that two of them would be triggered, while the other two would remain in their current state. We observed participants' strategies and interviewed them.

Interfaces & task: As illustrated by Figure 6, the interface was "implemented" using four paper buttons taped to the floor. There were a small and a large button labeled "ok", and a small and a large button labeled "cancel". Large buttons measured 40cm x 60cm, small buttons 10cm x 10cm.

During the study, participants walked across the four buttons. Half of the participants were tasked to "activate" the two 'cancel' buttons and get across the 'ok' buttons without activating them; the other half was instructed to activate the 'ok' buttons instead. An experimenter observed participants' strategies. Finally, participants explained their rationale in a verbal interview.

Participants: We recruited 30 participants (6 female) from our institution; they were between 21 and 29 years old.

Results: Figure 6 shows selected participants performing the task. Together, participants demonstrated nine different strategies (Table 1).

Figure 6: Five participants demonstrate how they activate a button: (a) tapping, (b) jumping, (c) walking on center, (d) dwelling, and (e) stomping.

Strategy	To activate	To *not* activate	#
Part of foot	tap (ball only)	walk	**8**
	walk	*tiptoe***	1
	walk on ball	*walk on heel***	1
Amount of pressure	stomp	walk	**5**
	jump onto	walk	**2**
Temporal	double-tap	walk	**2**
	Dwell (with both feet)	*walk quickly***	5
Left-right	right foot	*left foot***	*1*
Spatial	walk across centre	*walk edge of button**	*5*

Table 1: Strategies and number of participants who employed them. (* does not work with of densely packed controls, ** raises ergonomic concerns)

Discussion: The breadth of strategies emphasizes that there is no widely accepted model for interaction using feet.

Not all demonstrated strategies are applicable to all scenarios. The strategy "walk along edge of button" fails for densely packed arrangements of tiny controls, such as the pixels of a painting program—some pixels are always hit straight on. Four other strategies raise ergonomic concerns (marked ** in Table 1): Walking on heels and tiptoeing can get tiring over time. Activating by dwelling requires users to walk perpetually in order to not activate. Activation with the right foot requires users to hop on their left foot when crossing large controls without activating.

The remaining four strategies (*tap*, *stomp*, *jump*, and *double tap*) seem suitable. The strategy demonstrated by the largest number of participants was *tap* with 8 participants. Based on these findings we implemented *tap* into our system (see Section "Algorithms" for how we implemented this based on FTIR pressure sensing).

2. INVOKING A MENU

We face similar requirements when designing a mechanism for invoking menus. In theory, a spatial strategy is possible, such as a toolbar or the corner buttons that *Microsoft Surface* uses to close applications [15]. However, since interactive floors can become arbitrarily large, so can the distances to a stationary menu. Fixed menus therefore only make sense if replicated at a large number of locations and/or for very infrequent tasks. For the majority of tasks, users will prefer a *location-independent* interface [25].

To invoke such a context menu, we can pick any of the leftover invocation strategies from User Study 1, i.e., *stomp*, *jump*, and *double tap*. We found "jump" to offer the best recognition rate—it also virtually never occurs unintentionally. The implementation of jumping is straightforward. The floor tracks users (see Section "Algorithms") and if both of their feet go out-of-range for more then 200 milliseconds, the system invokes the menu.

3. STEPPING ON OBJECTS, ACTIVE CONTACT AREA

In order to manipulate objects on the floor, users need a basic pointing technique. Since shoes occupy a substantial amount of space, they can hit objects in many different spatial relationships, such as with their edge or arch (Figure 8).

When defining a pointing technique we need to decide which parts of a shoe should be used for hit testing.

Candidates include a point-like *hotspot* [12], the entire *contact area* [3], and the projection of the shoe (outline of the shoe projected onto the floor). In order to understand which of these models matches the users' conceptual model or whether we need an entirely different model we conducted a brief user study.

User study: Conceptual Model of Stepping

The goal of this study was to understand which area of their soles users consider to be active in targeting and should thus be considered in hit testing.

Task & procedure: (a) Participants stepped onto the multi-touch floor with their dominant foot wearing shoes. (b) A honeycomb grid was displayed under the participant's shoe (Figure 8). The cells of the grid were described to the participants as defunct "buttons." (c) For each "button" an experimenter asked the participant if it should be depressed based on the participant's foot position. If the answer was 'yes', the experimenter "set" the respective buttons which caused it to change color (Figure 8). All participants completed the task in 5min or less.

Apparatus: We used the floor prototype shown in Figure 5.

Participants: 20 participants (6 female) aged between 20 and 29 participated.

Results: Figure 7 shows shoes and button states for all 20 participants. 8 of 20 participants matched the *projection* model, i.e., they set every button at least a certain percentage of which was covered by the projection of the participant's shoe. This included tip and arch. Another 7 participants matched the *projection* model, but left occasional omissions along the outline (Figure 7b).

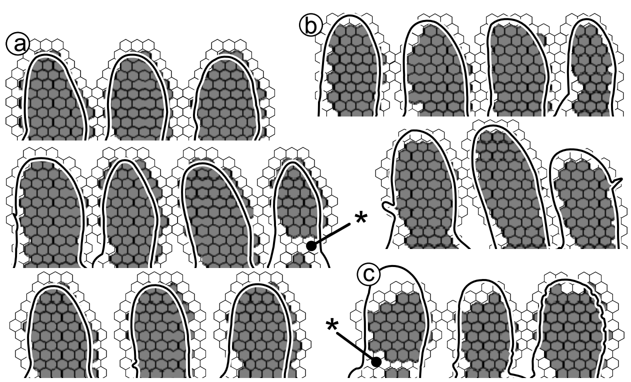

Figure 7: Resulting conceptual models. (a) 10 of 20 participants' model is *projection*; (b) projection with minor omissions (c) 3 excluded the upward curved tip. (*) Only 2 excluded the arch.

Three participants excluded the curved up tip of the shoe (Figure 7c); 2 excluded the arch (Figure 7*). One of them wore 5cm heels, the other, a male participant, wore sneakers. He rationalized that the arch was not touching the floor, suggesting that his conceptual model was based on *contact area*.

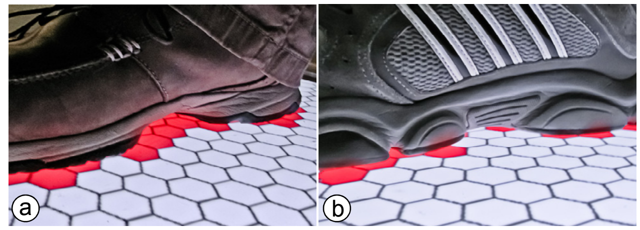

Figure 8: (a) 18 of 20 participants felt that the area under the arch should be included, while (b) the remaining two felt it should be excluded.

Note that when participants referred to contact area they did to in an idealized way. This does not necessarily correspond to the reality on an FTIR floor, where pressure and outlines change as users change body postures over time (Figure 9).

Figure 9: The FTIR contact area changes as the user changes posture over time.

Discussion: These finding suggest that the most common conceptual model of stepping is *projection*, even though some users erode the area a bit. The tracking model of FTIR, i.e., contact area, in contrast does generally *not* match the conceptual model of the majority of users.

We therefore implemented stepping primarily based on the Front DI component of our system (the dark outline in Figure 9). To alleviate front DI's sensitivity to shadows cast by the user's body, we combine the approach with some FTIR support (see Section "Algorithms").

4. HIGH-PRECISION POINTING WITH HOTSPOT

While the *projection model* makes for a good default model of floor interaction, it prevents application designers from packing controls tighter than a foot. As illustrated by Figure 10, huge controls require users to walk between buttons or extend themselves in order to reach. More importantly it prevents applications from using large numbers of objects, which defeats the original purpose of switching from table to floor. In order to allow for complexity, we need to allow applications to create small objects and to pack them densely.

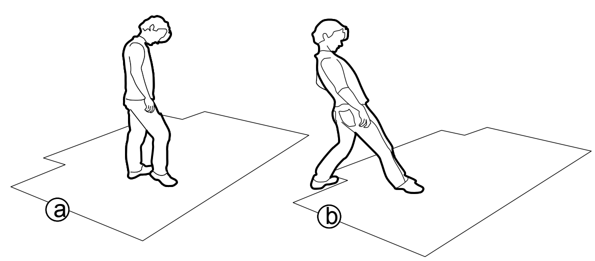

Figure 10: Operating a keyboard with foot-size buttons requires users to (a) walk between buttons or (b) to extend themselves in order to reach.

In order to allow for interaction with dense clusters of on-screen objects, we introduce a high-precision mode in which users' feet are reduced to a single *hotspot*. In analogy to the previous section, we started by investigating users' conceptual model, i.e., which part of their foot they consider to be the hotspot. Is there a single global hotspot or how much variation is there across users?

User study: Conceptual model of the hotspot

The purpose of this study was to survey what point on their shoe (the *hotspot*) users use to interact with point targets. We also wanted to find out how much agreement there was about the location of the hotspot: strong agreement would suggest a single global solution, while little agreement would suggest the need for personalization.

Task: As illustrated by Figure 11, (a) participants wearing shoes stood on a "waiting" position marked with circles. (b) For each trial, a target marked with crosshairs appeared 30cm in front of the user. Participants placed their preferred foot onto the crosshairs, such that the foot's hotspot was located directly over the crosshairs. Participants confirmed their selection by pressing a button on a wireless presenter tool. (c) Pressing the button recorded the floor's FTIR image of the user's foot as well as a photo of the user's foot from above. Finally, participants stepped back into the waiting position.

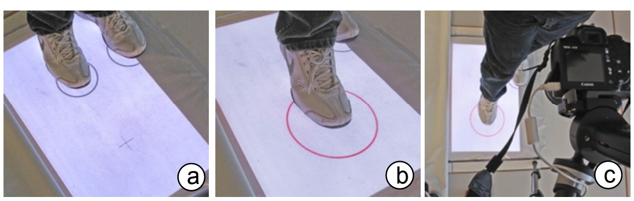

Figure 11: (a) When the crosshair appeared (b) participants stepped onto it. (c) In addition to the FTIR image, trials were recorded from above.

Independent Variables & procedure: Each participant performed the task in four conditions, three repetitions each. The first time, they were not given any further instructions (*free choice* condition). In the other three conditions, participants were instructed to aim using the *ball* or their foot, the *big toe* of their foot, and the *tip of their shoe*. In order to prevent the more specific conditions from influencing the *free choice* condition, the order of conditions was *not* counterbalanced—participants always performed the free choice condition first.

Participants We recruited 24 participants (8 female) from our institution; they were between 20 and 29 years old. Two participants were left-footed and thus performed all trials with their left foot.

Apparatus: We again used our FTIR floor prototype, as well as a *Canon EOS 1000D SLR* camera.

Results: Figure 12 shows participants' hotspots mapped onto outlines of their shoes. Black dots denote contacts made during *free choice* trials. The three triplets of white dots belong to shoe tip, big toe, and ball from tip downwards.

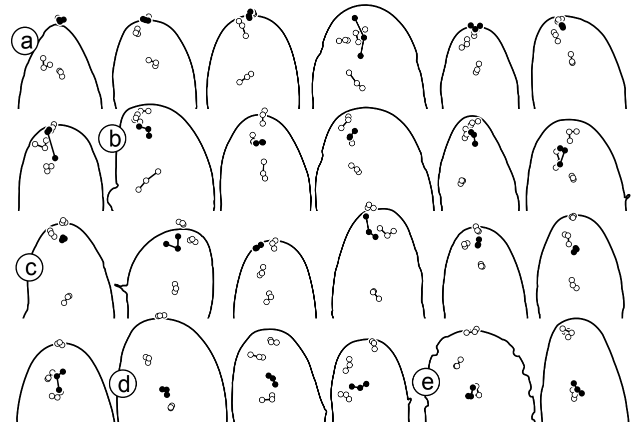

Figure 12: Participants acquire the target using these points on their soles (black dots: free choice; white dots are shoe tip, big toe, ball from tip downwards). Participants aimed using (a) tip, (b) big toe, (c) Offset from toe, (d) offset from ball, and (e) ball.

Free choice condition: As illustrated by Figure 12**,** we classified *free choice* point triplets according to which of the other triplets they were closest to. Based on this, 7 participants seemed to aim using the tip of their foot, 6 using their big toe and another 6 at an offset from the toe, and 2 using their ball and another 3 with a point slightly above the ball.

Figure 13 illustrates how participants' hotspots relate to each other by overlaying shoe outlines, so that the respective hotspots (centroid of all three trials) align. The spread of 8.4cm in Figure 13a suggests substantial disagreement between the *free choices* of participants.

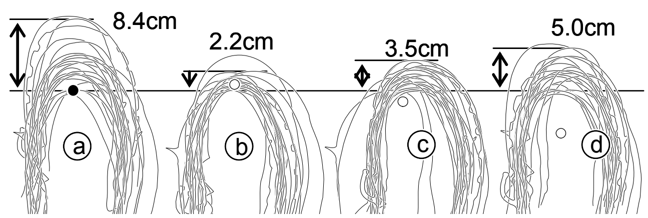

Figure 13: Outlines of participants' soles centered on the centroid of contact points for (a) free choice (b) tip, (c) big toe, and (d) ball. The dot indicates the position of the cross.

Discussion and resulting implementation: The substantial spread among *free choice* hotspots implies that the use of a global hotspot for all users would incur a large targeting error. This error can be reduced by instructing users to aim with a specific part of their shoe, in particular the tip (e.g., to 2.2cm in the case of the sample shown in Figure 13b). However, such an approach would fit the conceptual model of only 7 of the 24 participants.

To eliminate the necessity to train users, we allow users to customize their hotspot. When users step on the floor for the first time or with a new pair of shoes, a dialog with crosshairs is pops up. Stepping onto the crosshairs defines the user's hotspot and assigns it permanently to the respective location in the FTIR sole pattern of this pair of shoes (see Section "Identifying Users").

User study 4: Targeting precision with custom hotspot

As discussed earlier, our purpose for including FTIR into floors is to allow for direct manipulation of complex applications with large numbers of objects. This implies the necessity to support interaction with small objects. In order to inform application design, we conducted another study to determine the lower bound on the size of such objects. The participants' task was to enter text using on-screen foot keyboards of three different sizes.

Interfaces: All on-screen keyboards offered 28-keys (a-z, <space> and '.') in a localized QWERTY layout (Figure 14a). All three interfaces were identical except for scale. We picked a range of sizes that would capture a wide range of error rates. Overall the three keyboards measured 52.0 × 23.2cm, 31.0 × 14.0cm, and 15.0 × 6.8cm. Figure 14b illustrates key sizes; note that the keys on the *small* keyboard were smaller than keys on a physical QWERTY keyboard (Figure 14). Space bars were 3 times wider than regular keys.

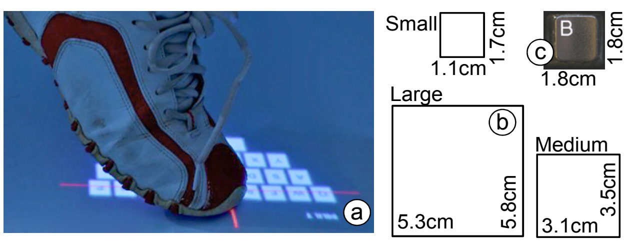

Figure 14: (a) The small keyboard. (b) The sizes of the keys on the large, the medium, and the small keyboard (c) Key of a physical keyboard for scale.

Participants wore their own shoes and targeted using a self-selected hotspot. Since our goal was to study the limits of *user* abilities, we minimized tracking-related inaccuracy by attaching an extruded dot (a Ø11mm nut) to the user's hotspot, which eliminated remaining tracking errors.

Task: For every trial, participants entered the sentence " the lazy brown dog." (including the leading blank and the trailing period), which was shown above the keyboard. Participants typed by tapping keys with one foot, while standing on the other foot. Typing the first character started the timer. Correct key presses turned the letter in the display green, incorrect key presses red. In addition, a brief sound indicated whether a key press was correct or incorrect. Participants had to retype erroneously entered letters until they got it right, but did not have to delete erroneous entries using backspace. Typing the trailing period stopped the timer.

Procedure: Participants typed the sentence twice on each of three keyboards for an overall number of six repetitions. The order of keyboard sizes was counterbalanced. Finally, participants filled in a questionnaire. All participants completed the study in less than 10min.

Participants: 26 participants (9 female) between 19 and 29 years old participated.

Apparatus: We used the same setup as in previous study.

Results: Figure 15 summarizes error rates and task times for the regular buttons of the three keyboards. Error rates for space bars were comparable (2.0%, 8.6%, and 23.8%). As expected, error rate and task time increased with decreasing button sizes. Note that about half of the error rate came from tapping outside the keyboard, a strategy we saw participants employ to avoid tapping incorrect letters. Task time mirrors the trends seen in error rate.

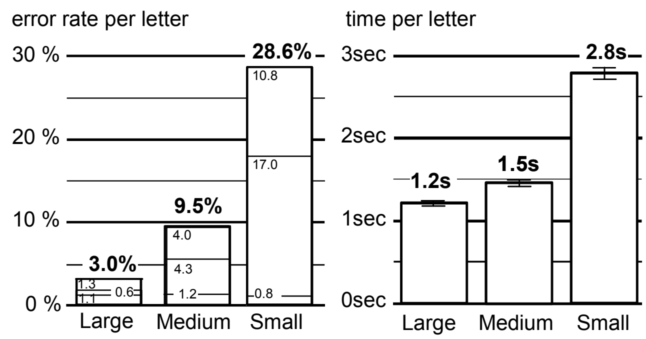

Figure 15: (a) error rate per letter (error types top-to-bottom: outside keyboard, neighboring key, wrong key) (b) time per letter.

Figure 16 shows the complete targeting data of all trials. The fact that contact point cluster centroids are centered on button centers suggests that all remaining error is indeed noise, rather than a systematic effect, such as ergonomic issues or head parallax.

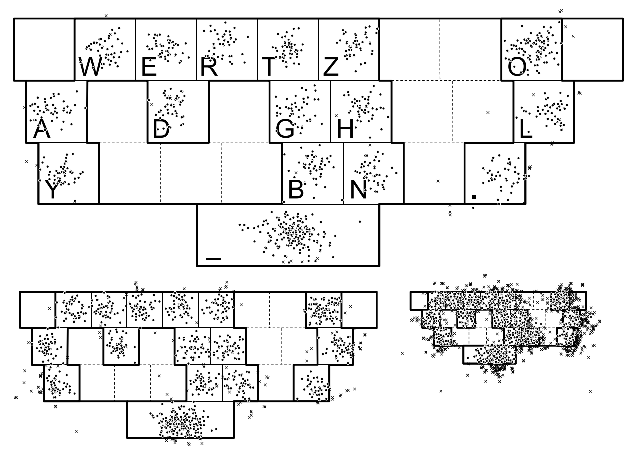

Figure 16: The contact points for all trials of all participants.

In the final questionnaire, half of the participants selected the *large* keyboard as their favorite. Interestingly, 10 of 26 participants picked the *medium* keyboard, where they found buttons easier to *reach*. The remaining 3 participants were indifferent between the large and the medium keyboard.

Discussion: With error rates close to 30%, the 1.1cm keys on the *small* keyboard were clearly too small. The 3.1cm buttons of the medium keyboard, however, might be acceptable for some applications where packing density is more important than error rate. Using a tiled layout, a 3x4m floor could pack 10,000 of such interactive objects, supporting our goal of bringing highly complex applications to multi-touch surfaces.

The large keyboard, finally, offers very good error rates below 2.9% and fully comparable to error rates on interactive tabletops. Note that at 5.3cm, buttons on the large keyboard are still quite compact and an order of magnitude smaller than the 1"+ buttons required by the projection model.

This completes the first part of the paper. We looked at enabling the basic techniques required for direct manipulation on our interactive floor and in particular at retrofitting the tabletop interaction model to the floor. FTIR and High-resolution camera input played a key role here, e.g., because they allow the floor to distinguish users and thus personalize the interaction. High-res FTIR enables a range of other possibilities and we will inspect them in the remainder of this paper.

ALGORITHMS AND NEW FTIR FLOOR INTERACTIONS

We now discuss the underlying algorithms that enable user identification and tracking of foot postures and balance. We also demonstrate how the same algorithms can be used to implement a simple type of head tracking and to enable foot interaction with high-degrees of freedom.

General processing

All processing on the FTIR floor starts with the following steps implemented in the open computer vision toolkit *OpenCV* [18], as illustrated by Figure 17.

1. By using higher illumination intensity for FTIR than for front DI (Figure 3c), we can extract the *FTIR image* from the raw image by thresholding (Figure 17b).
2. Extract the *DI image* by replacing the FTIR portion in the raw image with shoe color, i.e. black (Figure 17c).
3. Thresholding the DI image to allow finding connected components (Figure 17d)
4. Determine the main axis of the sole by fitting an ellipse onto the blob.
5. Determine which end of the main axis is the front of the shoe by testing which half of the convex hull of its contour is wider (Figure 17d).
6. Obtain rotation value from step 4 and orientation from step 5 and use them to rotate the FTIR image to a standard rotation.
7. Obtain shoe height and width from the oriented bounding rectangle (Figure 18e).

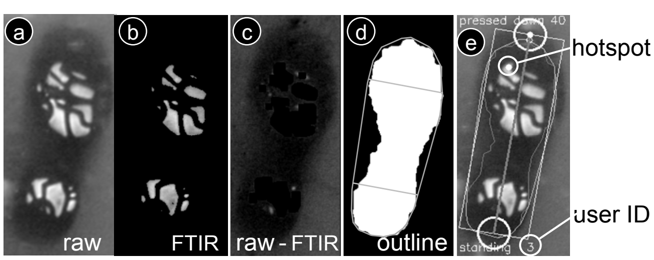

Figure 17: Sole processing (a) raw image, (b) thresholding extracts FTIR, (c) DI = raw minus FTIR, (d) Thresholding (blur(DI)) with convex hull and widths, and (e) annotated.

Additional processing is done based on the requirements of the application, as we discuss in the following.

Identifying users

User identification determines whether an observed sole is contained in the floor's sole database and if so pick's the most likely candidates. Our algorithm starts by pre-selecting candidates from the database based on similarity in sole length, width, and surface area, as well as the sole's grayscale histogram. We then perform a series of comparisons by sliding template images over the observed image to find the position with the best match in the image (*OpenCV*'s matchTemplate( ) function [18]). If the absolute difference between the two images is below a threshold, the footprint in the database is considered a match. FTIR brings out the unique line patterns and logos that shoe makers embed into their soles, which helps recognition.

If a foot print is not recognized for several frames, it is added as a new pair of shoes to the database. At this point it is labeled *anonymous* and assigned a random ID. In addition, the system brings up a dialog that allows users to identify themselves (Figure 18). This allows the floor to assign the user's name to the new shoes.

Figure 18: When the floor sees a pair of soles for the first time, it asks for identification.

Analysis of pressure distribution

All other functions, such as walking vs. tapping and head tracking are computed based on the *pressure distribution* on soles. All functions have in common that they partition the FTIR image of each foot into one or more cells, estimate the physical weight resting on each cell, and then compare this cell pressure with other cells.

In slightly more detail, the algorithm proceeds as follows. (1) Mask the FTIR image with the DI blob. (2) Partition the FTIR image into a set of cells. (3) Translate pixel color into pressure. It is important compensate for the non-linear pressure response of FTIR by applying the inverse of the pressure to pixel brightness function shown in Figure 19. We created this function by sampling the material-specific pressure response of our floor. (4) Sum up the pressure per pixel per cell. (5) Compare cell pressure with other cells.

We can implement the aforementioned functions using appropriate cells partitioning.

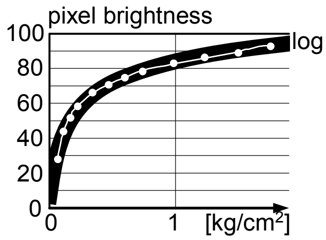

Figure 19: We found the brightness response of our stack-up to be roughly logarithmic with pressure (acrylic waveguide with *Tectosil 500* silicone and *Rosco* projection screen)

Classifying tapping vs. walking

In order to distinguish walking from taping we partition the users sole into a front cell ("ball") and a back cell ("heel"). Now we port the algorithm by Choi and Ricci [4] to FTIR: The floor observes the pressure patterns of the two cells over time and when it sees "nothing, ball, entireFoot, heel, nothing" it classifies the user as walking (Figure 20); if it sees "nothing, ball, nothing" as tapping.

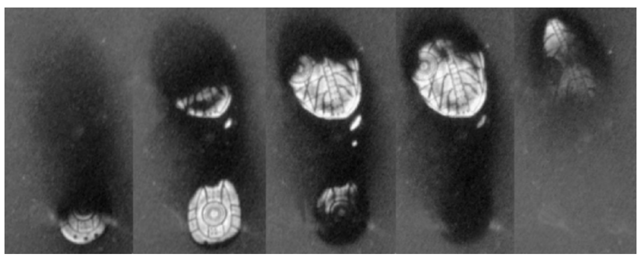

Figure 20: Using FTIR, "walking forward" is identified as a heel-ball pressure sequence.

Tracking the user's balance and head position

Unlike immersive and stereoscopic installations, such as *CAVEs* [5] or smart rooms [2], the position of body or head plays only a subordinate role in the context of direct manipulation scenarios. Nonetheless, FTIR-based pressure sensing allows us obtain a simple approximation of the user's posture.

Again, we partition users' soles into front cells and back cells, which gives us four cells whenever both feet are in contact with the floor. We determine the user's left-right balance as the pressure difference between the cells of each foot; we determine the user's front-back balance as the pressure difference between front cells and back cells.

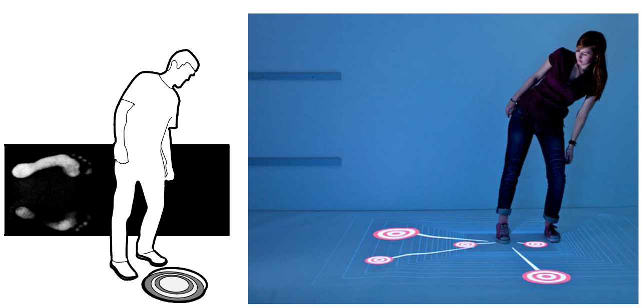

Figure 21: Sensing pressure using FTIR enables *fish tank VR* (here uses front projection).

To enable the *fish tank virtual reality* [36] demo shown in Figure 21, we calibrated our pressure input. We recorded pairs of user head position and pressure distribution for the centered and extreme forward, backwards, left, and right positions. This allows us to compensate for posture biases and the natural 60:40 pressure ratio between ball and heel. During the fish tank VR experience, we interpolate angles linearly.

Additional degrees of freedom

Finally, we can create additional degrees of freedom by subdividing soles further. Figure 22 shows a user playing a first person shooter on our prototype, hands-free, by controlling the game using her feet alone. We obtained 10 degrees of freedom by subdividing each foot into five zones; we then use a subset of them to implement functions for moving, strafing, and shooting, a subset of which is reproduced in Figure 22b. Note that users fire and alt-fire using their left and right large toes. Surprisingly, this continues to work inside of shoes.

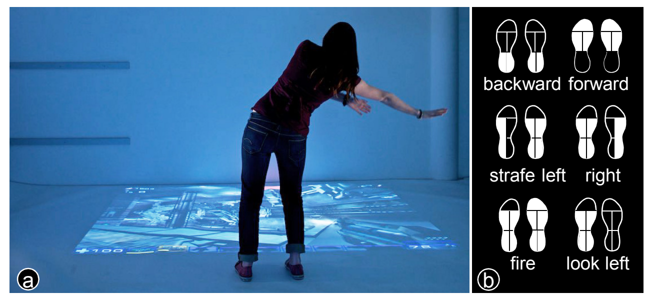

Figure 22: (a) Playing a first person shooter based on balance and foot posture (*Unreal Tournament 2004*) (b) subset of the mapping (front projection)

CONCLUSIONS

In this paper, we have presented high-resolution FTIR-based floors as a means to create interactive multi-touch surfaces beyond the size of tabletop computers. We made first steps toward enabling the interaction model of touch, i.e., direct manipulation, on such a floor. FTIR played the key role in this process, as it allows us to reliably tell interacting from walking and because it provides the high accuracy required to acquire and manipulate small objects. We argue that the combination of small objects (3-6cm, comparable to objects on tabletops) with the dramatically larger scale of interactive floors forms an interesting platform for enabling complex applications that deal with ten-thousands of objects.

Combining high-resolution FTIR with a projected floor also resulted in the design of additional interactions. One of them is user identification based on sole patterns, which works in part because users are bound to floors by gravity—very different from tabletops. Finally, we showed that we can extract more degrees of freedom from feet that at least we had expected.

As future work, we plan to explore the use of back-projected FTIR floors as part of smart rooms that monitor the well-being of people inside, yet respect privacy by only observing the contact area. In order to explore these effects we are currently creating the larger prototype shown in Figure 23.

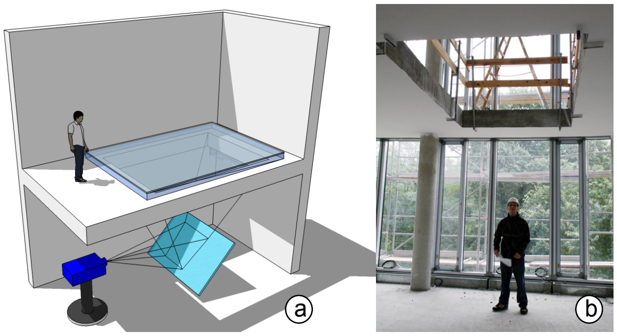

Figure 23: The next prototype, currently under construction

ACKNOWLEDGEMENTS

We thank Gerald Morrison for his input and compliant surface, Sebastian Boring, Florian Echtler, Peter Brandl, Christian Mueller-Tomfelde and Raphael Wimmer for their insights on multitouch technology and Frank Priester, Timm Krohn, and Christoph Meinel for their support.

REFERENCES

1. Bay, H., Tuytelaars, T., and Van Gool, L. Surf: Speeded up robust features. *Computer Vision and Image Understanding (CVIU),* Vol. 110, No. 3, 346–359, 2008.
2. Brumitt, B., Meyers, B., Krumm, J., Kern, A., and Shafer, S. A. EasyLiving: Technologies for Intelligent Environments. In *Proc. Ubicomp 2000*, 12-29.
3. Cao, X., Wilson A., Balakrishnan, R., Hinckley, K., and Hudson, S. ShapeTouch: Leveraging Contact Shape on Interactive Surfaces. In *Proc. TABLETOP'08*, 129-136.
4. Choi, I., Ricci, C. Foot-mounted gesture detection and its application in virtual environments. *IEEE International Conference on Systems, Man and Cybernetics. '97*, 4248-4253.
5. 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.
6. Davidson, P., Han, J. Extending 2D object arrangement with pressure-sensitive layering cues. In *Proc. UIST'08*, 87-90.
7. Dietz, P. and Leigh, D. DiamondTouch: a multi-user touch technology. In *Proc. UIST'01*, 219-226.
8. Greenberg, S. and Buxton, B. Usability evaluation considered harmful (some of the time). In *Proc. CHI '08*, 111-120.
9. Grønbæk, K., Iversen, O. S., Kortbek, K. J., Nielsen, K. R., and Aagaard, L. IGameFloor: a platform for co-located collaborative games. In *Proc. ACE '07*, 64 – 71.
10. Han, J. Y. Low-cost multi-touch sensing through frustrated total internal reflection. In *Proc. UIST '05*, 115-118.
11. Hartmann, Björn, Morris, M.R., Benko, H., and Wilson, A. Augmenting Interactive Tables with Mice & Keyboards. In *Proc. UIST'09*, 149-152.
12. Holz, C., and Baudisch, P. The Generalized Perceived Input Point Model and How to Double Touch Accuracy by Extracting Fingerprints. In *Proc. CHI'10,* pp. 581–590.
13. Krogh, P. G., Ludvigsen, M., Lykke-Olesen, A. Help me pull that cursor. A Collaborative Interactive Floor Enhancing Community Interaction. In *Proc. OZCH'04*, 22-24.
14. LaViola, J. J., Feliz, D. A., Keefe, D. F., and Zeleznik, R. C. Hands-free multi-scale navigation in virtual environments. In *Proc. I3D '01*, pp. 9-15.
15. Microsoft Surface http://www.microsoft.com/surface
16. Morris, M.R. Designing Tabletop Groupware. In *UIST'05 Doctoral Symposium.*
17. Mota, S. and Picard, R. Automated posture analysis for detecting learner's interest level. In *Proc. CVPR'03*, 49-56.
18. *OpenCV* http://opencv.willowgarage.com
19. Orr, R. J. and Abowd, G. D. The smart floor: a mechanism for natural user identification and tracking. In *CHI'00 Extended Abstracts*, 275 – 276.
20. Paradiso, J. A., Hu, E. Expressive Footwear for Computer-Augmented Dance Performance. In *Proc. ISWC '97*, 165-166.
21. Paradiso, J., Abler, C., Hsiao, K., Reynolds, M. The Magic Carpet - Physical Sensing for Immersive Environments. In *Proc. CHI' 97,* 277-278.
22. Paradiso, J., Hu, E., Hsiao, K.Y. Instrumented Footwear for Interactive Dance. In *Proc. IDAT '99*, 89-92.
23. Pavlou, M. and Allinson, N.M. Automatic extraction and classification of footwear patterns. In *Proc. IDEAL'06*, 721–728
24. Pavlou, M. and Allinson, N.M. Footwear Recognition. In *Proc. Encyclopedia of Biometrics'09*, 557-562
25. Pier, K. and Landay, J. Issues for Location-Independent Interfaces. Tech. Rep. ISTL92-4, Xerox PARC, Dec 1992.
26. Piper, A., O'Brien, Ringel Morris, M., and Winograd, T. SIDES: a cooperative tabletop computer game for social skills development. In *Proc. CSCW'06,* 1-10.
27. Potter, R. L., Weldon, L. J., Shneiderman, B. Improving the accuracy of touch screens: an experimental evaluation of three strategies. In *Proc. CHI '88*, 27-32.
28. Richardson, B., Leydon, K., Fernstrom, M., and Paradiso, J. A. Z-Tiles: building blocks for modular, pressure-sensing floorspaces. In *CHI '04 Extended Abstracts*, 1529 - 1532
29. Rosenberg, I. and Perlin, K. The UnMousePad: an interpolating multi-touch force-sensing input pad. In *Proc. SIGGRAPH '09/ACM Trans. Graph.* (Vol. 28, No. 3, Article 65), 1-9.
30. Shirai, A., Sato, M., Kume, Y., and Kusahara, M. Foot Interface: Fantastic PhantomSlipper. Report ER14, Tokyo Institute of Polytechnics, 1998; see also *SIGGRAPH 98Conference abstracts and applications,* 114.
31. Smart Table http://smarttech.com
32. Sugiura, A. and Koseki, Y. A user interface using fingerprint recognition: holding commands and data objects on fingers. In *Proc. UIST '98*, 71-79.
33. Visell, Y., Law, A., and Cooperstock, J. R. Touch Is Everywhere: Floor Surfaces as Ambient Haptic Interfaces. *EEE Trans. Haptics 2, 3 (Jul. 2009)*, 148-159
34. Vogel D. and Balakrishnan R. Occlusion-Aware Interfaces. In *Proc. CHI '10*, 10 pages.
35. Vogel, D. and Baudisch, P. Shift: a technique for operating pen-based interfaces using touch. In *Proc. CHI '07*, 657-666.
36. Wang, F., Cao, X., Ren, X., Irani, P. Detecting and leveraging finger orientation for interaction with direct-touch surfaces. In *Proc. UIST '09*, 23-32.
37. Ware, C., Arthur, K., and Booth, K. S. Fish tank Virtual Reality. In *Proc. INTERCHI '93*, 37-42.
38. Matsushita, N., and Rekimoto, J. *HoloWall*: Designing a Finger, Hand, Body, and Object Sensitive Wall. In Proc UIST'97, 209-210.