Ergonomic Interaction on Touch Floors

Dominik Schmidt[1], Johannes Frohnhofen[2], Sven Knebel[2], Florian Meinel[2], Mariya Perchyk[2], Julian Risch[2], Jonathan Striebel[2], Julia Wachtel[2], Patrick Baudisch[1]
Hasso Plattner Institute
Potsdam, Germany
[1]{first.last}@hpi.de, [2]{first.last}.student@hpi.de

Figure 1: We argue that touch floors have bad ergonomics as they are designed for being used while standing, which causes fatigue, especially in combination with looking down. We thus propose allowing users to operate touch floors in other poses. Based on a series of studies, we have created a simple view manager that supports users in switching poses by re-layouting screen content.

ABSTRACT
The main appeal of touch floors is that they are the only direct touch form factor that scales to arbitrary size, therefore allowing direct touch to scale to very large numbers of display objects. In this paper, however, we argue that the price for this benefit is bad physical ergonomics: prolonged standing, especially in combination with looking down, quickly causes fatigue and repetitive strain. We propose addressing this issue by allowing users to operate touch floors in any pose they like, including sitting and lying. To allow users to transition between poses seamlessly, we present a simple pose-aware view manager that supports users by adjusting the entire view to the new pose. We support the main assumption behind the work with a simple study that shows that several poses are indeed more ergonomic for touch floor interaction than standing. We ground the design of our view manager by analyzing, which screen regions users can see and touch in each of the respective poses.

CHI 2015, April 18 - 23 2015, Seoul, Republic of Korea

ACM 978-1-4503-3145-6/15/04…$15.00
http://dx.doi.org/10.1145/2702123.2702254

Author Keywords
Ubicomp; Smart Rooms; Interactive Floor; GUI; Multi-touch; Ergonomics.

ACM Classification Keywords
H.5.2. [Information Interfaces and Presentation]: User Interfaces–Input devices and strategies, interaction styles.

INTRODUCTION
The main appeal of touch floors (e.g., [1,11,26]) is that they allow creating interaction spaces of *arbitrary* size, while maintaining the affordance of direct touch. This makes touch floors different from touch tables, tablets, etc. whose size is limited by arm's reach—even interactive walls, which can be arbitrarily long, are limited in height. Touch floors thus open new opportunities as they can be designed to scale direct touch to handle substantially bigger problems [1]. Working on bigger problems, however, not only requires space—it may also require users to interact for longer periods of time.

In this paper, we argue that interaction for longer periods or time can be an issue, because of the particular form factor of touch floors: prolonged standing, especially in combination with looking down, quickly causes fatigue and repetitive strain. The hydrostatic blood pressure in the legs increases over time, causing discomfort [10]. Even worse, users find themselves continuously looking down, increasing the risk of neck pain, a milder form of which is also known to notebook computer users, as their screen position also makes users look down [5].

We propose addressing floor ergonomics by allowing users to operate touch floors in poses that are less prone to fatigue, such as sitting or lying. As illustrated by Figure 2, we argue that floors naturally afford multiple poses—unlike most existing direct touch form factors (e.g., tabletops require sitting, interactive walls require standing).

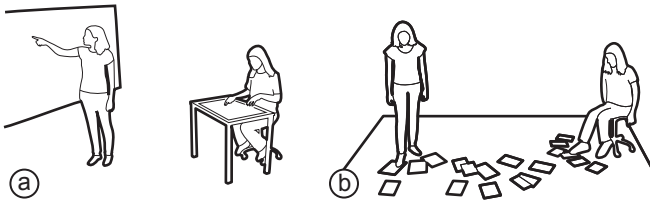

Figure 2: Unlike (a) other touch devices, such as wall displays or tabletops, (b) we argue that touch floors allow users to assume a range of poses.

CONSTRUCTING A POSE-AWARE VIEW MANAGER

To help users work in any pose, we propose the concept of a "pose-aware" view manager with the following functionality. (1) Such a system should support users in accessing their content in their current pose. (2) Since any pose gets tiring after some time (see Study 1) users will *change* poses periodically, relieving whatever tension the previous pose has caused. The pose-aware view manager should allow users to continue their interaction seamlessly whenever such a pose switch occurs.

We construct our pose-aware view manager in three steps:

Step 1: We verify that at least some of the other poses afforded by touch floors are indeed less fatiguing than standing. We do this in the form of a simple user study. The study also tells us *which* out of many possible poses work best, thus should be supported by our view manager.

We use the same study to also extract the first set of parameters we need for constructing our pose-aware view manager, which is the *scale* at which users would like to view content for each possible pose.

Step 2: We determine the remaining parameters we need for our view manager, which is *where* to place content. Direct touch interaction requires users to see *and* touch content. By intersecting the two we determine candidate regions for each pose.

Step 3: Based on these findings, we present our pose-aware view manager, which we termed *PoseUI* (Figure 1).

CONTRIBUTION

In this paper, we make two main contributions. (1) We argue that touch floors afford interaction in various other poses than standing. We verify this assumption in a simple study in which we find that *several* of the poses afforded by touch floors are less fatiguing than standing. (2) We present a simple pose-aware view manager that allows users to continue their interaction seamlessly when they switch between poses. We build this view manager on a careful analysis of the necessary parameters, in particular scale and placement of content.

RELATED WORK

This work builds on interactive floors, research in ergonomics, direct-touch tabletop UIs, and automatic UI layouting.

Interactive Floors

Interactive floors are part of the ubiquitous computing vision, that is, computers being integrated into the environment, thus becoming invisible [24]. Researchers have explored floors from different perspectives, for example, as part of immersive environments for virtual [7] and augmented realities [18], as unobtrusive sensing platform (e.g., for pose detection [4]), and for interactive installations (e.g., to support collaboration in public space [15] or for gaming [11]). In addition, recent work has proposed precise direct touch manipulation for interactive floors [1] and floors as platform for tangible UIs [19].

Ergonomics

Research in ergonomics on other devices and form factors suggests that changing poses relieves fatigue (e.g., [10,21]) and recommends work equipment that encourages frequent pose change [16].

Direct-Touch Tabletop UIs

Interactive floors share major UI design challenges with direct-touch tabletops, particularly with respect to multi-user collaboration, content orientation, occlusion, and reach [20].

To allow for interacting with objects beyond a user's physical reach, researchers have proposed a range of techniques, such as temporarily moving target icons closer to the user [3], switching between absolute and cursor-based input [8], or interacting with a scaled-down world-in-miniature view [22].

To avoid occluding information, *Display Bubbles* dynamically wrap content around physical objects placed on a table [6]. Vogel and Balakrishnan automatically show callouts to make information visible that would otherwise be obscured by the user's hand and arm [23].

Automatic UI Layouting

A simple layout adaptation is implemented by *Lean and Zoom*, which detects the user in front of a laptop computer leaning forward and zooms the UI accordingly to meet halfway [12]. *SUPPLE*, in contrast, automatically pre-renders UI layouts for devices with varying constraints, using models of the user and the device and treating interface generation as an optimization problem [9]. *Cassowary* is a linear solver that maintains layout constraints of interactive UIs [2].

STEP 1: TOUCH FLOORS AFFORD OTHER POSES THAT ARE LESS FATIGUING THAN STANDING

The primary objective of this user study was to validate our assumption that some of the poses afforded by touch floors are indeed more ergonomic than standing.

At the same time, this study provides a first basis for our pose-aware view manager in that (1) it tells us which poses are worth supporting (2) it allows us to determine at what scale users would like to view content in each pose.

Task

Participants performed a repetitive direct touch task (i.e., the game "whack-a-mole" shown in Figure 3). Each participant performed this task in each of the eight different poses shown in Figure 4. Note how some poses allowed users to interact using their hands, others using their feet.

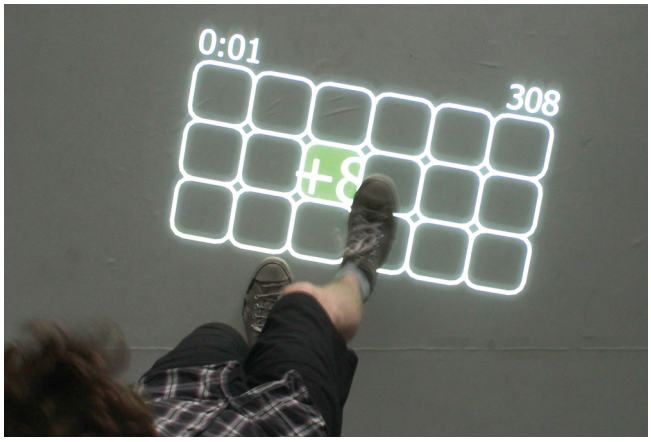

Figure 3: Participants performed a repetitive "whack-a-mole" direct touch task, i.e., they tried to touch/tap highlighted buttons as soon as they appeared.

For each trial, participants made themselves comfortable in the respective pose; they were allowed to use pillows if they preferred. Participants then rotated and scaled the game board so that they could reach it comfortably; we logged these settings. We intentionally initialized interfaces to too small a scale, forcing participants to tediously scale up their interface by repeatedly pushing a button. This approach, however, allows us to interpret the resulting scales as a lower bound for scale, which is one of the required parameters for our pose-aware view manager.

Participants then performed the whack-a-mole task for 60 s. At the end of a trial, participants rated the pose using a questionnaire (two ratings: one referring to the first seconds and one rating referring to the end of the 60-second period).

We used a within-subjects design (i.e., each participant interacted in all eight poses). The order of these poses was counterbalanced.

The study was designed to establish *lower bounds* on ergonomic issues. While 60 s are a comparably short amount of time, one would expect any fatigue effects found to be amplified during prolonged use.

Participants

We recruited eight participants (two female) from our institution. They were between 21 and 26 years old ($M = 22.1$ yr, $SD = 1.7$ yr).

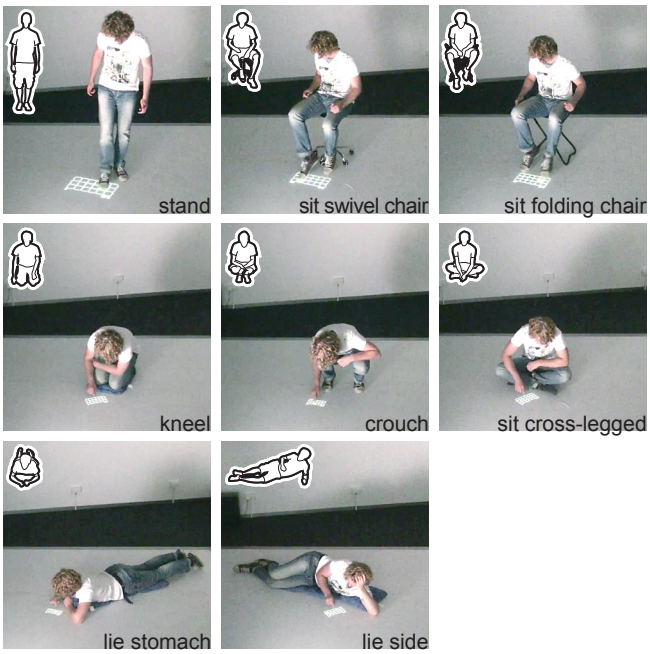

Figure 4: Every participant performed the study task in each of these eight poses.

Results: Fatigue

Figure 5 shows the median fatigue rating for each pose. For the beginning of each trial, participants rated only one pose as less tiresome than standing (pose "sitting cross-legged", bold outline) on a scale from 1 (effortless) to 7 (tiresome). For the end of the 60 s trial, participants rated *four* poses as less tiresome (bold outlines). A Friedman test showed significant differences in pose ratings for the beginning of a trial ($\chi^2(7) = 18.63$, $p < 0.01$) as well as for the end of a trial after 60 s ($\chi^2(7) = 15.27$, $p < 0.05$).

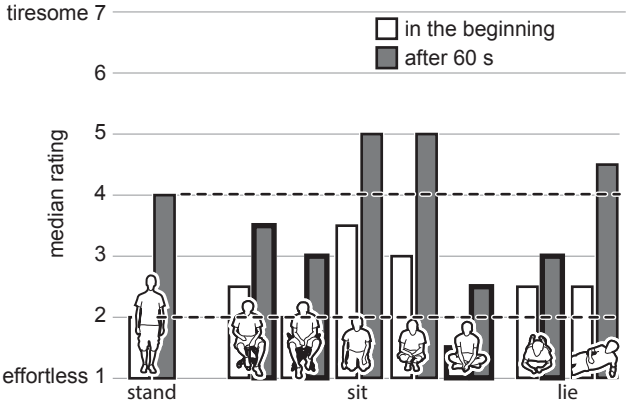

Figure 5: Participants rated "sitting cross-legged" as less tiresome than standing upright in the beginning of a task. After 60 s, four poses had better rating than standing (bold outlines).

For four participants, standing was their highest rated pose *in the beginning*. However, standing was rated highest only for one participant after 60 s. As illustrated by Figure 6, all but one pose received the highest rating from at least one participant. We assigned fractional points for draws.

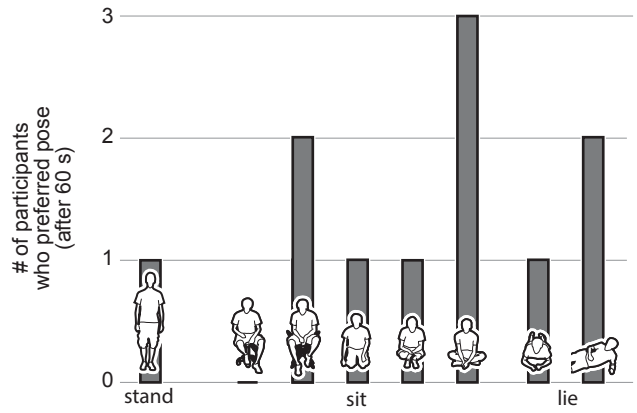

Figure 6: All but one pose received the highest rating from at least one participant (we assigned fractional points for draws).

Discussion: These findings support our main hypothesis: touch floors indeed afford other poses that are more ergonomic than standing.

The findings also inform our design of a pose-aware view manager, in that they tell us which poses should be supported. If we choose to support the poses that cause less fatigue than standing (which is four of them) as well as poses favored by at least one user (which surprisingly is all but one, Figure 6), the answer is that *all* tested poses should be supported by a pose-aware view manager.

Results: Scale

Figure 7 shows how participants scaled their interfaces for the respective pose. As expected there was a significant main effect of pose onto scale ($F(7,42) = 5.73$, $p < 0.001$).

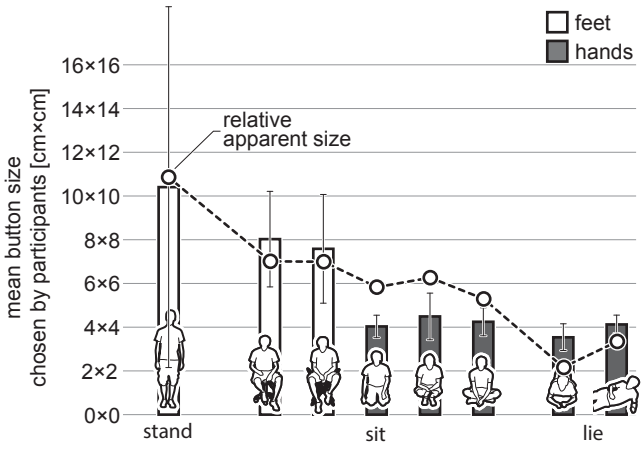

Figure 7: Mean button sizes chosen by participants (error bars: ±1 std. deviation): they seemed to maintain the apparent UI size, i.e., scaled the interface relative to standing.

Discussion: As expected, there was a positive relationship between participants' head height and scale of the interface. A correlation of 0.87 ($p < 0.01$) suggests that participants might try to scale content so as to *appear* to them at a certain size across pose. We calculated the apparent size as 1 / distance (between eyes and floor).

Poses that let participants interact using their feet resulted in larger buttons than those that let participants interact using their fingers. This is expected, based on the different accuracy levels of hands and feet in direct touch [1,13].

Figure 7 shows a reasonably good fit for the three poses that let users interact using their feet as well as for the two lying poses, in which participants used their hands to interact. For sitting poses, in contrast, it seems that the higher accuracy of finger touch allowed participants to make the interface smaller than suggested by apparent size. While this worked well for our study task, scaling below the apparent size might not be desirable for other tasks, such as reading text or performing visual search.

These findings suggest that document scale may be approximated by means of head height and whether users use feet or hands to interact. This suggests that a pose-aware view manager may not have to fully capture the user's pose; instead, we might obtain a good approximation by only tracking the vertical position of the user's head and determining whether a pose affords foot or finger touch.

STEP 2: WHERE TO PLACE CONTENT ACROSS POSES

Now that we know how to scale content across poses, our next step in constructing our pose-aware view manager is to determine *where* to place content for each pose.

Direct touch interaction requires users to see *and* touch content. In this section we model both for each of the eight poses used earlier. For each pose, we then intersect visible and touchable regions, obtaining candidate regions for direct touch.

What Users Can See

Figure 8 shows how we approximately model what users can see in a given pose. We use the simple metric of how long it would take a user to see a documents placed at a certain location. Our model distinguishes three areas.

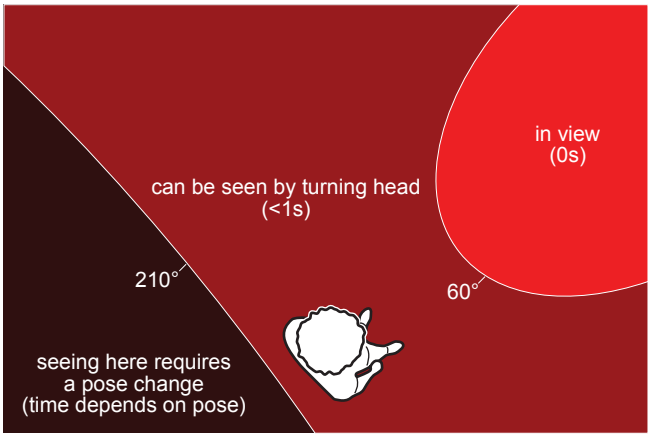

Figure 8: Our simple model of what users can see

1. "In view" We compute this area by casting a cone-shaped 60° spotlight [14] from the point between the user's eyes.

2. "Can be seen in 1s by turning head" We determine this area using a 210° cone (60º maximal eye + 150º lateral neck rotation [17]).

3. "Seeing here requires a pose change." The time it takes to see a document here depends on the user's pose (e.g., is longer for sitting vs. standing).

In addition, some areas are occluded by the user's body and thus not visible. We will handle these later when we determine what users can touch.

To get a sense for what areas can be seen in each pose, we created the simple real-time visualization shown in Figure 9. It tracks the user using an optical motion capture system (*OptiTrack,* naturalpoint.com), then uses the GPU to calculate the visibility of each pixel based on shoulder orientation and incident angle of gaze.

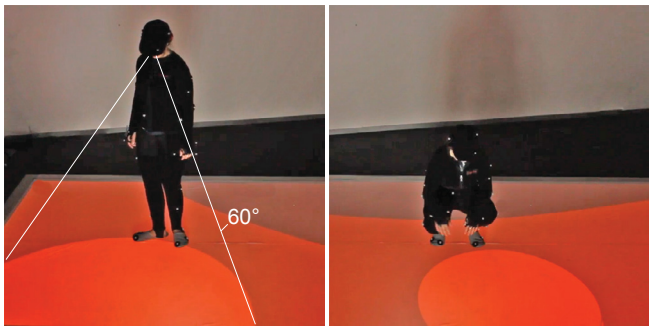

Figure 9: Our simple real-time visualization allows us to get a sense for what users can see from which pose.

For our pose-aware view manager, we want content to be visible without requiring a pose change. Figure 10 shows the resulting models, which we obtain by simply removing the area that users can only see by means of a pose change.

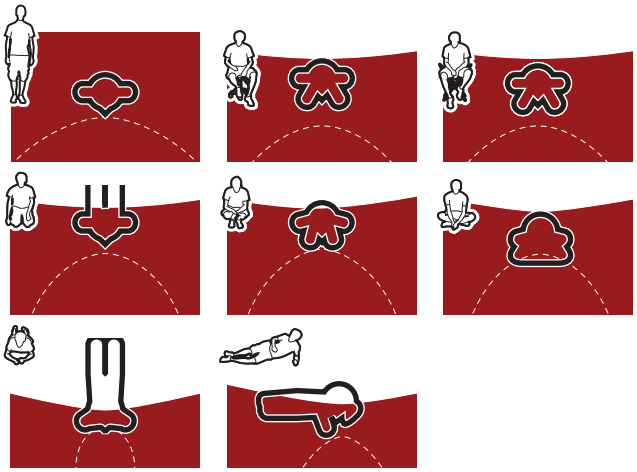

Figure 10: What users can see for each pose. Standing users can turn and walk around without changing pose. Thus, we do not prune the visible space for standing.

We are making one exception though: standing users can turn around and walk around without changing their pose, making it easy to access content anywhere—a big strength of this pose (and probably the reason it has been so popular in the related work, e.g., [1,11,26]). We reflect this in that we do *not* prune the space for standing and instead consider all screen space to be acceptable for placing content.

What Users Can Touch

To determine what users can touch in each pose, we conducted another simple study. We recruited eight participants (two female) from our institution between 22 and 32 years old ($M = 26$ yr, $SD = 3.7$ yr) and between 165 and 180 cm tall ($M = 173.4$ cm, $SD = 5.9$ cm). Their arms were between 52 and 66 cm ($M = 59$ cm, $SD = 4.5$ cm), their legs between 92 and 105 cm ($M = 99.8$ cm, $SD = 4.3$ cm) long. As shown in Figure 11, we placed them on a study interface consisting of a 5 cm × 5 cm button grid and asked them to show what areas they could touch, using their left, their right hand, their left foot, and their right foot. We had them repeat this for each of the eight poses in counterbalanced order.

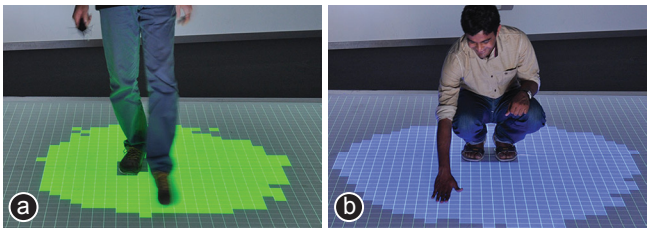

Figure 11: We asked participants to indicate areas they could touch in different poses using their hands and feet, for example, (a) left foot (standing) and (b) right hand (crouching).

Figure 12 shows the participants' touch outlines for feet and hands. Areas that could be reached by at least *two thirds* of participants are filled green for feet and blue for hands. As before, we make an exception for standing and include the entire screen space for placing content.

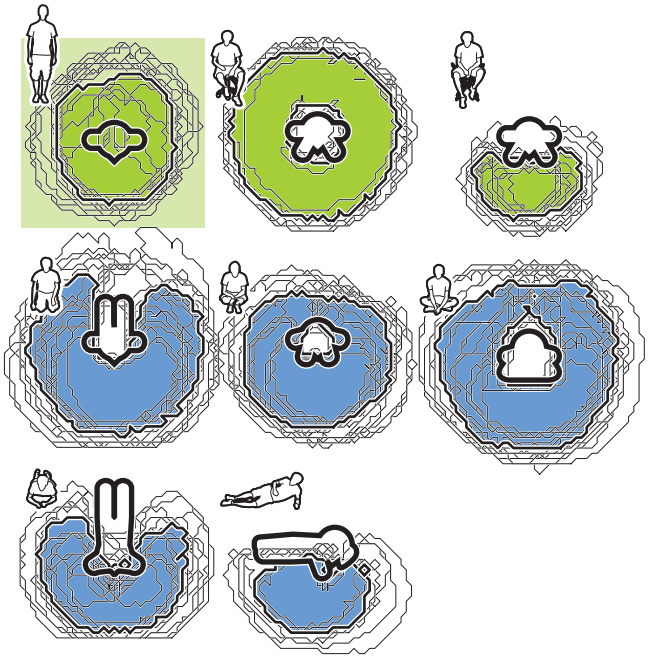

Figure 12: Areas participants could touch in different poses using feet (green) and hands (blue). Standing allows users to reach any location on the floor without changing pose.

Direct Touch: What Users Can See *and* Touch

Figure 13 shows the resulting direct touch areas for each pose as the intersection of what users can see with what they can touch.

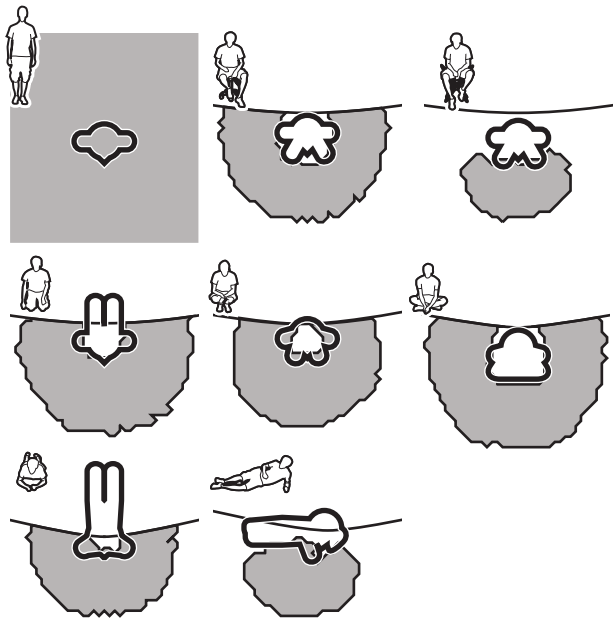

Figure 13: Direct touch is possible where what a user can see intersects with what a user can touch. Standing takes on a special role as users can move without additional effort.

We now have all necessary information (scale and placement) for placing content with respect to any of the eight poses.

STEP 3: A POSE-AWARE VIEW MANAGER

Based on the findings from Steps 1 and 2, we created a simple pose-aware view manager. It rearranges the user's documents to match the new pose after a pose change.

Prototype Framework

When the user changes pose, the system's objective is to select all relevant documents, scale them to the target scale, and arrange them into the target direct touch area with the goal to make all documents that users could see and touch in the previous pose visible and reachable in the new pose.

To enable this, we created the following hardware setup. In order to detect pose changes, a pose-aware system has to continuously monitor the users' pose. Pose recognition on interactive floors is a well-understood problem (e.g., FootSee [25], GravitySpace [4]); we thus simply used an optical motion capture system. For our first prototype we used a full-body tracking suit; based on insights from Study 1, we simplified the tracking set-up to a single rigid body marker on the users' shoulder for our second prototype. All prototypes run on a 360 cm × 260 cm back projected touch floor [1] with software written in C++ and some components running on the GPU.

V1: Optimized Layout at the Expense of Spatial Memory

Based on this hardware setup, we created a first prototype. This version used a "per-document" approach to pose change as illustrated by Figure 14: when the user changed pose, the system selected only those documents currently in reach (determined based on Figure 13; for standing, we chose to include all documents at most two steps away). The system then not only scaled documents according to the scales determined in Study 1 (Figure 7) but also rearranged them to best fit into the direct touch area of the target pose. It did so by moving documents individually.

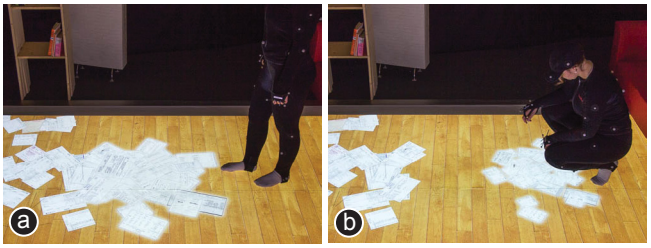

Figure 14: First version re-laid out documents individually, invalidating users' spatial memory by breaking up groups.

This approach worked well in the sense that it maximized the number of documents users could reach in the new pose. On the other hand, the approach performed poorly in that the re-layout severely affected the neighborhood relationships between formerly adjacent documents, even breaking up clusters. This severely affected users' spatial memory. In addition, and arguably more important, the rearranged layout offers no visual cue *how* it was rearranged. As a matter of fact, most neighborhood relationships are still intact—there is just no obvious way for users to tell.

Based on this experience, we redefined our design objectives by complementing our objective of an optimal layout with a second objective reflecting the desire to minimize impact on spatial memory. Any re-layout has the potential to disorient users [3]. We thus created a second, improved version of our view manager that minimizes change.

V2: Preserving Spatial Memory

Figure 15 illustrates the redesigned version of our view manager, which we termed PoseUI.

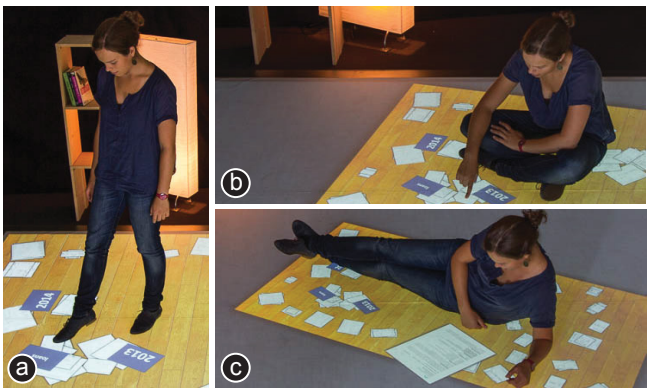

Figure 15: (a) This user is sorting documents for her tax return. Initially, she stands. (b-c) As she sits or lies down, PoseUI positions documents optimally for these poses.

Like the previous version, PoseUI arranges documents so that the user can still see and reach the same documents in the new pose. Compared to its predecessor, however, we

made eight changes in order to better preserve users' spatial memory. We describe these changes in the following.

1. Simplified Transformation

Where the previous prototype moved documents individually, PoseUI applies a single global transformation (translation, scale, rotation) to all documents at once. This obviously limits the level of optimization the document layout can achieve, especially when the shape of the direct touch area of the target pose differs from the shape of the previous pose (e.g., sitting cross-legged vs. on folding chair). However, most direct touch layouts look similar in that they form an abstract crescent shape, so in the typical case the simplified transformation works well.

Figure 16 shows how PoseUI places content for a rectangular region building on the parameters determined earlier in this paper. If we use our 360 cm × 260 cm touch floor as an arbitrary reference, we see that some poses can almost accommodate the space as a whole, trivializing the layout question even with our simple transform-as-a-whole approach. After this scale-adaptation, however, some of the poses offer access to much fewer documents. Particularly the two chair-based poses offer access to very few, thus may be less suitable for manipulating large collections of documents.

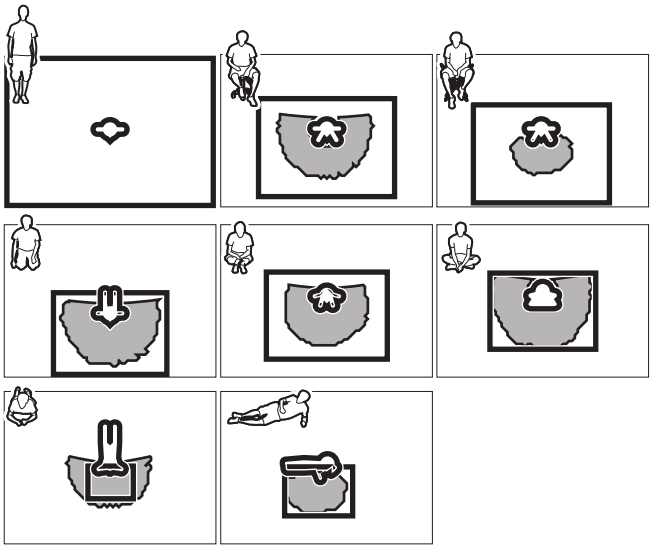

Figure 16: PoseUI applies a single global transformation including the entire floor area with all its content.

2. Visually Explained Transformation

The approach of transforming the layout already makes the layout in the new pose much easier to understand. In order to allow users to *instantly* grasp how the layout was adapted, PoseUI transforms the background image behind the documents along with the documents (here we gave it the texture of hardwood floor); we will refer to it as *backdrop*.

The backdrop in the version shown in Figure 17 was chosen to be of rectangular shape—a particularly good choice in that the rectangular outline of the backdrop already contains the transformation's three parameters translation, rotation, and scale (minus symmetries), so that it allows users to read these transformation parameters intuitively from the outline of the backdrop alone.

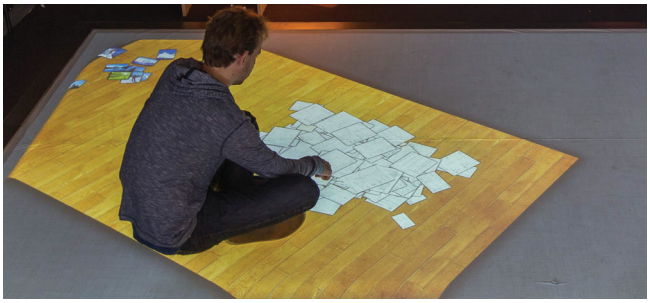

Figure 17: PoseUI transforms the entire floor imagery alongside with the documents, which helps users to understand the current layout information.

By scaling the backdrop along with the documents, all documents now stay on top of the same spot of the backdrop throughout all transformations. The backdrop and especially its frame thereby serve as landmarks, allowing users to locate familiar content at intuitively named locations such as "the top right corner of the hardwood floor", leveraging spatial memory.

3. Resolving Clipping at the Edge of the Screen

The simplified translation, rotation, and scale transformation is obviously less flexible and thus can lead to parts of the backdrop ending up off-screen. To prevent documents from becoming inaccessible, PoseUI pushes documents together at the edge of the screen. To help users understand the effect, PoseUI *bends* the backdrop in these border regions, as illustrated by Figure 18. This way, documents again remain positioned at their spot in the backdrop, helping the backdrop serve as landmark, leveraging spatial memory.

Figure 18: PoseUI bends the backdrop where pushed outside the screen.

4. Resolving Occlusion by Physical Objects

Similarly, transforming the space during pose change can make it clash with objects placed on the floor, such as furniture. To prevent documents from becoming inaccessible, PoseUI pushes out content from underneath such objects.

Early versions of our algorithm resolved occlusion by pushing out each document towards the closest edge of the occluding object (similar to *Display Bubbles* [6]). Unfortunately, this locally stretched the layout by an infinite factor, piling up documents very densely *towards* the occluding objects, while it spread documents out only very loosely *along* the perimeter. This distortion obfuscated neighborhood relationships and once again made it hard to tell which documents belong together.

We therefore redesigned our approach to occlusion resolution. As shown in Figure 19, the improved version produces space for the occluding object in the form of a *cut*. Along the edges of this cut, the floor bends up, similar to how it behaves at the edges of the screen. If the cut runs through a document, the document stays whole and moves to the closest sides, which keeps the documents readable. If the occluding object moves or is moved, the effect animates with the object slicing up the backdrop, resealing the cut behind it.

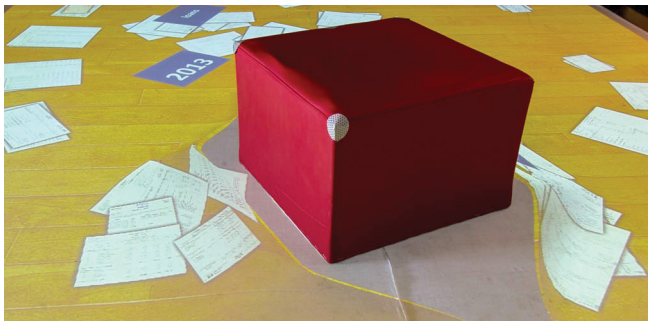

Figure 19: PoseUI prevents documents from being occluded by physical object by cutting the virtual floor and bending it up on both sides.

The main benefit of the cut is that it better preserves neighborhood relationships. It fully preserves neighborhood relationships along the cut; neighborhood relationships across the cut can still be reconstructed to a certain extent by measuring along the cut.

The underlying algorithm creates a raster of vertices surrounding the cut-to-be, which are moved outwards following a function parameterized by the size of the occluding object; this function defines the shape of the cut. We implemented our algorithm as shader to allow for fast update rates.

5. Using "Borrowed" Space

If content and thus the backdrop are scaled down, additional space is created that is void of documents. PoseUI marks this "borrowed" space by showing a pattern visual distinct from the backdrop (here gray linen cloth).

In order to maximize users' performance, PoseUI allows users to use borrowed space for the task at hand, i.e., users can place and manipulate documents in that space (Figure 20a). At the same time, the visual distinction is designed to clarify the nature of borrowed space: when the user stands up, this space will go away and documents placed here will be "scooped up" against the edge of the screen (Figure 20bc). Borrowed space thus serves as a temporary cache.

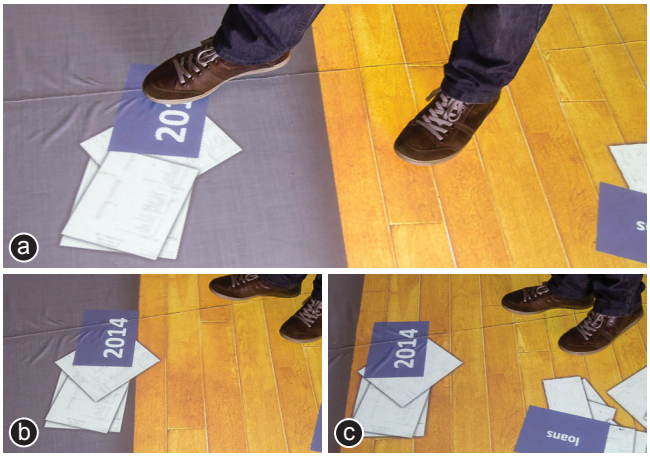

Figure 20: (a) This user places documents on "borrowed" space. These documents do not follow layout transformations, but are (b, c) "scooped up" once the backdrop moves across.

6. Layout Consistency Over Time

PoseUI uses the following two-step algorithm to prevent distortions from accumulating and to keep the layout consistent across any series of pose changes. Whenever the user changes pose, PoseUI computes the target layout by first restoring the backdrop to its rectangular shape, which undoes any distortion caused by screen edges and occluding objects. It then applies the transformation of the target layout and renders it. This approach keeps documents anchored with respect to the backdrop at all times.

PoseUI uses any return to the standing pose as an opportunity to reset the backdrop to its default position, i.e., to scale and rotate the backdrop such that it aligns with the physical screen. This resolves any backdrop clipping from previous poses and maximizes the amount of visible content.

7. Supporting *Navigational* Pose Changes

So far, we talked about pose changes exclusively as a means to optimize ergonomics. However, pose changes obviously also are users' primary means for navigating towards desired content. Similar to how laptop users lean forwards to see content in detail and backwards to get an overview [12], touch floor users may walk towards content or squat in order to view specific content in detail (Figure 21a); they may get up in order to obtain an overview.

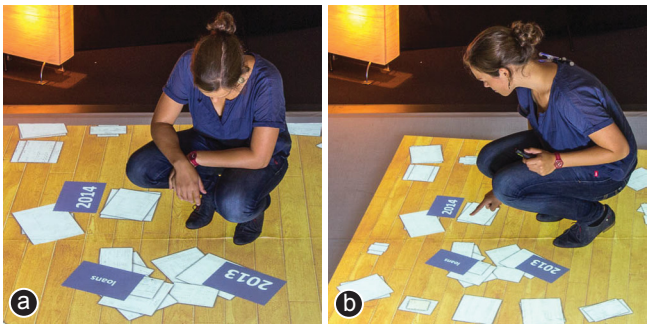

Figure 21: When a user (a) squats the system assumes the purpose is to inspect content in detail, thus not adjusts the view. (b) By marking the pose change as an ergonomic change (here using a wireless presenter), screen content adapts.

Such *navigational* pose changes occur more frequently than ergonomic pose changes. PoseUI therefore handles the distinction by letting users manually mark when a pose change is an *ergonomic* pose change. The problem of triggering events on touch interfaces is well understood (e.g., trailing widget [8]); in the shown version, we simply used the button on a wireless presenter. As a side effect, the explicit mechanism allows users to make content follow them as they pace across the floor (Figure 22b).

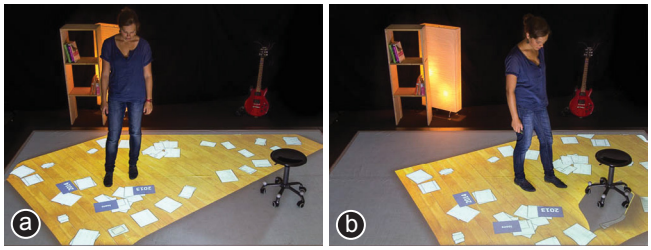

Figure 22: This user enjoys wandering around while interacting, which is supported by PoseUI.

8. Supporting Multiple Concurrent Users

When PoseUI sees multiple users, its default assumption is that these users collaborate on a single big task—as afforded by the size of touch floors. It therefore continues to provide a single shared backdrop with all documents.

Sometimes, however, users are more effective working in parallel on separate subtasks. PoseUI supports this by creating multiple backdrops, each of which forms a view onto the same set of documents (Figure 23). Moving a document in one of these views thus also moves that document in the other view.

In the particular implementation of PoseUI, prototype, we invoke this "forking" process manually: when one of multiple users indicates a pose change, PoseUI provides a separate view for this user.

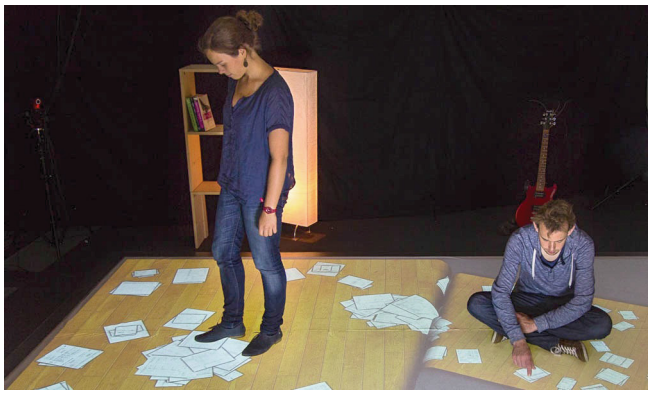

Figure 23: As the user on the right sits down, PoseUI clones the view, allowing each user to work independently.

As illustrated by Figure 1, PoseUI partitions space in the form of a Voronoi tessellation, i.e., all users are assigned the screen area that is located closest to them. This oftentimes generates overlap between the views, which PoseUI resolves the same way it handles clipping at the end of the screen, i.e., by bending the involved backdrops. Note how the concept of backdrops and borrowed space naturally extends to multiple users.

CONCLUSION

In this paper, we proposed improving the ergonomics of touch floor interaction by allowing users to interact in and across multiple poses. We used a three-step design process: (1) Our first study challenged the status quo, which is that touch floors are used almost exclusively for standing users. (2) Our first and second study determined the parameters for touch interaction in different poses. (3) Based on these results we presented a simple view management system called PoseUI that allows users to seamlessly continuing their task across poses. The main design objective behind PoseUI was to minimize the impact of pose change on users' spatial memory.

As future work, we plan on creating a complete GUI/NUI framework for touch floors, replacing the "documents" we used in this paper with the complexity of GUI/NUI components they are supposed to stand in for.

REFERENCES

1. 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.
2. Badros, G.J., Borning, A., and Stuckey, P.J. The Cassowary Linear Arithmetic Constraint Solving Algorithm. *ACM Trans. Comput.-Hum. Interact. 8*, 4 (2001), 267–306.
3. Baudisch, P., Cutrell, E., Robbins, D., et al. Drag-and-Pop and Drag-and-Pick: Techniques for Accessing Remote Screen Content on Touch- and Pen-Operated Systems. In *Proc. INTERACT'03*, 64–57.
4. 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.
5. Breu, F.X., Guggenbichler, S., and Wollmann, J.C. A Survey of Ergonomic Issues Associated with a University Laptop Program. *Journal of Education and Human Development 1*, 2 (2008), 1–15.
6. Cotting, D. and Gross, M. Interactive Environment-Aware Display Bubbles. In *Proc. UIST'06*, 245–254.
7. 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.
8. Forlines, C., Vogel, D., and Balakrishnan, R. HybridPointing: Fluid Switching between Absolute and Relative Pointing with a Direct Input Device. In *Proc. UIST'06*, 211–220.
9. Gajos, K.Z. and Weld, D.S. SUPPLE: Automatically Generating User Interfaces. *IUI*, (2004), 93–100.
10. Grandjean, E. *Fitting the Task to the Man*. Taylor & Francis/Hemisphere, 1989.

11. Grønbæk, K., Iversen, O.S., Kortbek, K.J., et al. iGameFloor - A Platform for Co-Located Collaborative Games. In *Proc. ACE'07*, 64–71.
12. Harrison, C. and Dey, A.K. Lean and Zoom: Proximity-Aware User Interface and Content Magnification. In *Proc. CHI'08*, 8–11.
13. Holz, C. and Baudisch, P. The Generalized Perceived Input Point Model and How to Double Touch Accuracy by Extracting Fingerprints. In *Proc. CHI'10*, 581–590.
14. Karmakar, S., Pal, M.S., and Majumdar, D. Application of Digital Human Modeling and Simulation for Vision Analysis of Pilots in a Jet Aircraft: a Case Study. *Work 41*, (2012), 3412–3418.
15. 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*.
16. Lueder, R. and Noro, K., eds. *Hard Facts about Soft Machines: the Ergonomics of Seating*. CRC Press, 1994.
17. NASA. Anthropometry and Biomechanics. In *Man-Systems Integration Standards (NASA-STD-3001)*. 1995.
18. 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.
19. Schmidt, D., Ramakers, R., Pedersen, E., et al. Kick-ables: Tangibles for Feet. In *Proc. CHI'14*, 3143–3152.
20. 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.
21. Sjogaard, G. and Jensen, B.R. Low-level Static Exertions. In *The Occupational Ergonomics Handbook*. 1999, 247–259.
22. Stoakley, R., Conway, M.J., and Pausch, R. Virtual Reality on a WIM. In *Proc. CHI'95*, 265–272.
23. Vogel, D. and Balakrishnan, R. Occlusion-Aware Interfaces. In *Proc. CHI'10*, 263–272.
24. Weiser, M. The Computer for the 21st Century. *Scientific American 265*, 3 (1991), 94–104.
25. Yin, K. and Pai, D.K. FootSee: An Interactive Animation System. In *Proc. SCA'03*, 329–338.
26. Epidémik. http://www.cite-sciences.fr/epidemik, Aug. 14th, 2013.