SurfacePhone: A Mobile Projection Device for Single- and Multiuser Everywhere Tabletop Interaction

Christian Winkler[1], Markus Löchtefeld[2], David Dobbelstein[1], Antonio Krüger[2], Enrico Rukzio[1]

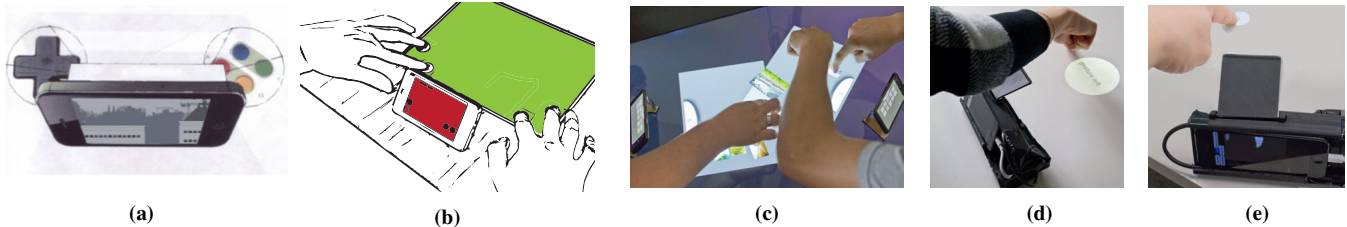| [1]Ulm University	[2]DFKI
Ulm, Germany	Saarbrücken, Germany
{firstname.name}@uni-ulm.de	{firstname.name}@dfki.de

Figure 1: The evolution of the SurfacePhone design. Starting with design sketches (a, b) over concept prototypes for multi-user interaction (c) to a fully implemented phone case for iPhone5 that supports gesture tracking (d) and touch recognition (e) using the phone's camera and accelerometer.

ABSTRACT

To maintain a mobile form factor, the screen real estate of a mobile device canIn this paper we present SurfacePhone; a novel configuration of a projector phone which aligns the projector to project onto a physical surface to allow tabletop-like interaction in a mobile setup. The projection is created behind the upright standing phone and is touch and gesture-enabled. Multiple projections can be merged to create shared spaces for multi-user collaboration. We investigate this new setup, starting with the concept that we evaluated with a concept prototype. Furthermore we present our technical prototype, a mobile phone case with integrated projector that allows for the aforementioned interaction. We discuss its technical requirements and evaluate the accuracy of interaction in a second user study. We conclude with lessons learned and design guidelines.

Author Keywords

Projector Phone; Interactive Surfaces; Mobile Multi Display Environment

ACM Classification Keywords

H.5.3 Information Interfaces and Presentation: Computer-supported cooperative work

INTRODUCTION

Mobile computing devices have caught up with their desktop counterparts in terms of computing power and even display resolution. But in terms of display real estate they are limited to maintain a small mobile form factor. The more

power and content is made available on our mobile companions the higher become users' needs to perform more complex tasks. This creates demand for bigger displays which inevitably leads to increased form factors. To overcome this problem, an increasing number of devices focuses on multi-display solutions. Besides devices based on multiple physical displays such as [8], projector phones set out to enable the exploration of large-scale content and support collaboration in a mobile setting.

In such mobile multi-display environments (MMDEs) the information is displayed across different screens, that are often spatially seperated and not in the same field of view. While a MMDE consisting of multiple physical displays easily allows for the extension of the screen real estate, it is still limited by the maximum size of the device. A MMDE consisting of a combination of physical and projected displays, however, allows an increase in screen estate to a much higher degree while still keeping the small form-factor, and can be used for collaboration and sharing. However the display setup of current projector phones such as the Samsung Galaxy Beam is often characterized by two displays being visually separated. This setup precludes many of the prevalent sharing and collaboration techniques that are well known and investigated for example in today's tabletop systems. In contrast, our setup allows to recreate such tabletop-like interactions in mobile scenarios with a private and a public display. In this paper we present SurfacePhone, a novel configuration of a MMDE that consists of a physical and a projected display. It is able to project a second display right behind itself, while it is standing on a surface (compare Figure 1d).

The investigated setup allows for collaboration and sharing as well as advanced single-user interactions. The projected display is touch- and gesture-enabled and additionally orientation aware. This allows connecting multiple projected displays into one combined display (see Figure 1c). In this paper we present the design process of the SurfacePhone. Starting with the considerations for such a system and the envi-

sioned usage concepts, we then present two prototypes that have been evaluated in user studies. The initial concept prototype allowed us to easily evaluate the concepts and ideas. The technical mobile phone case prototype (hard- and software open sourced) was developed to show and evaluate the technical feasibility of the SurfacePhone concept. We conclude with design guidelines based on our study results.

RELATED WORK
There has been an ever increasing interest in mobile projection interfaces in the last five years [5, 6, 15, 21]. Initial investigation of the advantage that a mobile phone with integrated projector can deliver has been done by Hang et al. [5]. Harrison et al. demonstrated the capabilities of a body-worn system [6]. All these systems demonstrate that pico projectors integrated into a mobile device allow for an increased display size and that the explored applications and interaction benefit from it significantly. Therefore the SurfacePhone integrates a mobile projector instead of a secondary physical display. This allows to maintain a small form factor while generating a significantly bigger secondary display.

MMDE devices have been not only explored in research [10, 8, 3, 21, 14], there are also commercially available devices such as the Nintendo DS or the Kyocera Echo. With Codex [8], Hinckley et al. created a dual-display device that allows for re-orientation of two physical hinged displays. Besides, they also explored different application scenarios for different configurations as well as novel interaction techniques. Nevertheless the display arrangement of the here presented SurfacePhone has not been mentioned nor explored in the related work.

Kane et al. explored with Bonfire a laptop equipped with a pico-projector that allows to create a secondary display right next to the notebook [10]. Even though the SurfacePhone also makes use of a similar configuration, one of its main aspects is to explore collaboration on a potentially mutual interactive surface with multiple devices. As a consequence, we present new sharing techniques between displays and devices and dynamic merging of their projections which lead to new public/private scenarios. Furthermore, the SurfacePhone improves on touch accuracy and portability and for the first time, evaluates touch and gesture recognition which could not have been carried out before.

The ad-hoc combination of multiple mobile devices is another possibility [9, 11]. For instance, the work of Lyons et al. [13] would allow to create such a setting. While there is evidence that different spatial arrangement of displays can have an impact in a mobile setting [2], it has been ignored in most MMDE investigations [3, 8]. A large body of literature also investigated partially mobile configurations with one fixed display and one mobile [4, 16, 17]. Those configurations mainly have been employed to create a personal display which interacts with a shared public display [4, 18]. In those settings the orientation of the displays to each other can be fluently changed since they are not coupled. The SurfacePhone takes a very similar approach, it combines a public and personal display that keep their spatial alignments.

The most related work to SurfacePhone is PlayAnywhere [19]. Just like PlayAnywhere the SurfacePhone can create an ad-hoc display on, e.g., a table and just like PlayAnywhere it allows for touch interaction. The main difference between those two systems is that beside SurfacePhone being a truly mobile device, it also provides an additional personal display to manage private content.

SURFACEPHONE CONCEPT
The design of the SurfacePhone concept encompasses the position of the projected surface in relation to the phone, the position and orientation of one SurfacePhone to other SurfacePhones in the environment, and the modalities to interact with screen and projected display in either scenarios. Further, we distinguish between *single device/single user* (SDSU), *single device/multi user* (SDMU), and *multi device/multi user* (MDMU) scenario groups.

Position and size of projection
Hinckley et al. [8] showed that a range of very private to very public and collaborative application scenarios can be supported, depending on the spatial relation of dual-screen postures. The projection in front of the mobile device would resemble the laptop posture or the display configuration, it is a very private setup. This is because the projection is mainly visible to the user facing the device. In such a configuration the projection could show a soft keyboard. A projection to either sides of the phone would imitate the setup of Bonfire [10], where the projected surface is still within easy reach of the user, but more public than in the laptop scenario.

These two configurations have been explored intensively but a projection behind an upright standing phone has been neglected so far. This setup – the SurfacePhone – consists of a public projected display and a private display (as can be seen in Figure 2a) and presents a more collaboration oriented setup. To some extent, it resembles the Battleship setup of Codex [8], albeit the difference that the primary user is able to see both the phone display and the projected display. In this setup, there is a clear separation between the private phone and the public projection that is visible and within reach to people in the near vicinity. This comes at the expense of a slightly more difficult interaction with the projection as the user has to circumvent the phone to touch the projection.

Additionally, this MMDE setup is in line with the findings of Cauchard et al. [2]. When the user is sitting in front of the upright standing phone, the phone's display as well as the projection are in the same field of view. This allows the SurfacePhone to split the information between these two displays without risking visual separation effects.

Details of our technical prototype can be found in section *Implementation*. But to give an idea of the size and position of the projection early-on: through experimentation we found an optimal (undistorted) projection behind the phone to be around 17cm $\times$ 14cm in size, 14cm behind the phone and 4cm to the left of the center of the device. The projection, thus, is three times as big as the 4" screen of the iPhone 5.

Configurations
The SurfacePhone can be used alone, or by multiple users using one or multiple SurfacePhones.
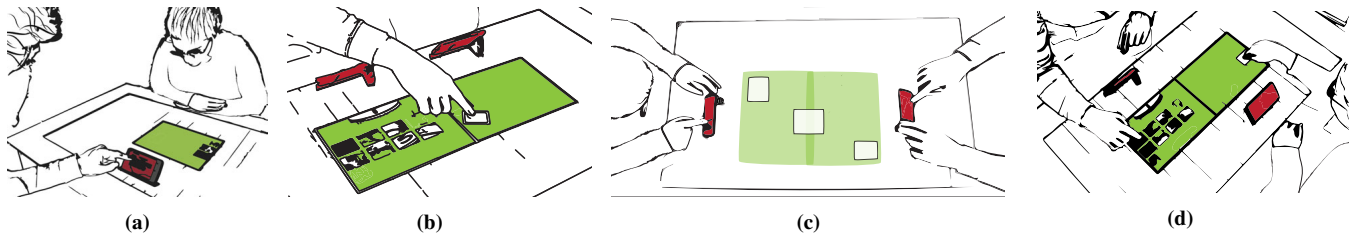
Figure 2: SDMU: (a) One user is presenting pictures to another user. MDMU: (b) Sitting next to each other on the same side with the projections merged at the short side. (c) Two users sitting face to face with the projections merged at the long side. (d) Two users sitting face to face with the projection merged at the short side.

Single-device, single user (SDSU)
This configuration can be used, for instance, to overcome the fat-finger problem on mobile devices by outsourcing e.g. controls of a game (compare Figure 1b) to the projection or showing the main view of the game on the projection. Apart from that, the projected display could be used as general secondary display, for instance, showing a task manager or notifications of applications currently running on the device. Finally, phone screens are very useful for augmenting the reality of the user, but cannot serve publicly visible augmentation. The projection on the other hand could be used to augment a real playboard with projected tokens. For example it could project chess tokens on a real board to play against the computer or a human opponent.

Single-device, multi-user interaction (SDMU)
Leveraging the inherent differences in publicity of the displays, the SurfacePhone can be used for several sharing tasks in small groups (compare Figure 2a). For instance, the projection of the phone can be used to present pictures or slides to a small group of people. The screen of the SurfacePhone can be used to browse the content and decide which content should be shown on the projection. Advantages of using SurfacePhone in this scenario include that users do not have to give out their phone to other people; that the content can be presented to all people simultaneously; and that only specific pictures or slides for presentation can be selected to address time or privacy constraints. Finally, the projection can also be touch- or gesture enabled, giving the viewers the possibility to interact with the pictures or slides. Similarly, the setup is also suitable for games such as blackjack: The person playing the bank controls the game from the screen. Other players sit in front of the projection and use touch interaction.

Multi-device, multi user interaction (MDMU)
Finally, when more than one user brings their SurfacePhone to the table, projections can be merged at different sides forming larger shared surfaces. These can be used for collaboration, e.g. data sharing, as well as competitive scenarios such as gaming. Depending on the scenario and the familiarity of the participants, different setups support different degrees of collaboration.

Sitting next to each other on the SAME SIDE (Figure 2b) is the most intimate setup as both the projections as well as the phone screens are visible to both users. This setup, for example, could be useful to collaboratively search for holiday trips. Users can first explore offers on their personal devices, then share it to the surface. Being able to also see other users'

phone screens may significantly improve communication in collaborative planning.

On the opposite, sitting FACE TO FACE (Figure 2c) merging the long side of the projections is the most distant setup. It suits users unfamiliar with each other, as well as competing opponents in a game for instance. In both cases, users have private interaction on their mobile display, using it to selectively share content on the projected surface. Also, the own projected display is likely not within easy reach of other parties making it more personal for each user.

Sitting face to face, but at the same time NEXT TO each other (Figure 2d) combines properties of both aforementioned setups. In this setup, users keep their private view on their mobile screens, but expose their projected surface to be easily reachable by the other party. Therefore, the setup particularly emphasizes familiar use cases of interactive surfaces, encouraging participants to manipulate all objects on the surface. Two users may also sit round the corner of the table which is in general equivalent to the previous case, but allows more easily to come round and take a look on the other user's private display when both users desire so.

Finally, groups > 2 merge projections at ARBITRARY SIDES in their center. Obviously, no general rule for the visibility of phone screens or reachability of projections can be determined. However, like people do when playing games involving hand cards, users can arrange to ensure the required visibility and privacy.

Interaction Techniques
In the following we will discuss required interaction techniques for the SurfacePhone that suit aforementioned application and usage scenarios. Here we draw from users' experience and familiarity with smartphones and tabletop systems. The technical feasibility of the here described techniques will be addressed in the implementation section of the technical prototype.

With today's prevalence of multi-touch interaction users would expect to be able to interact with the projected content using *direct touch*. This includes long touches and double-touches, to allow for a richer input set through different touch modalities. Furthermore, gestures like directional swipes are common on tabletops and should be supported as well. As the phone camera is watching the scene from above, mid-air gestures above the projection could also be considered.

Another interesting space of interaction lies *around* the projection. As the phone camera is seeing an up to ten times

larger space around the projection, invisible buttons around the projection are possible. Similarly, gestures that cross the edges of the projection could be supported, for example, to move content to another user's projection that is currently not merged.

As the SurfacePhone is a mobile device, movement of the device can be measured using the built-in motion sensors and the optical flow of the camera's video stream. The projection could, for example, be changed from showing display-fixed content that moves with the device to showing a dynamic peephole into world-fixed content. Any table could thus become the personal virtual desktop that is explorable by moving the SurfacePhone like a flashlight over the table.

A regularly occurring task when using the SurfacePhone is to transfer content from the screen to the projection and vice-versa. Following on [1] we can distinguish between three main categories of transfer-techniques that can be supported: *direct*, *binned*, and *mediated* transfer.

- *Direct transfer* is used to transfer an item from a specific position on the phone to a specific position on the projection or vice-versa. For this category we propose to use *Human Link*. The body of the user is conceptually used as a medium to transfer the content between the two displays (cf. [20]). The user touches the content that they want to transfer on the phone and then, simultaneously or in quick succession, touches the point in the projection where they want to place it or vice versa (Figure 3a).
- *Binned transfer* uses a bin element on one or either displays that is used to place content items in the bin that then can be transferred using a form of *direct transfer*. For instance, to place a whole word in the scrabble game, users can position the letters on the bench (the bin) on their phone screen in correct order and then transfer them altogether by swiping over the target positions on the projection (Figure 3b). Similarly, users could select pictures on their phone to a bin and then fan them out on the projection with a finger swipe.
- *Mediated transfer* uses a proxy or gate element through which content is transfered. To transfer a object simply drag&drop it on the *proxy* (Figure 3c).

CONCEPT PROTOTYPE

To explore and evaluate the SurfacePhone concept, we built a prototype to validate that the proposed display configuration is actually desirable and usable. Through the placement of a standard mobile phone on a multi-touch surface it is easily possible to simulate the projection behind the phone. This allows us to test users' experiences providing a more robust, responsive, and clearer multitouch surface than would have been possible through developing a technical prototype (which we present later on) in the same time.

Implementation

The hardware setup consists of a Samsung PixelSense table running Microsoft's Windows 7 and Surface SDK; further two HTC HD 7 running Windows Phone 7.5 which offer a stand to arrange the phone on a table more easily. Markers placed below the phones allow them to be tracked by the table. Our software framework creates a 23cm × 18.5cm sized virtual projection 9cm behind and 3cm to the left of the phone. This size exceeds the projection size that is supported by our technical prototype by 33%. As phone manufacturers surely are able to build devices that support projections of these dimensions by using short-throw lenses or curved mirrors, we assume the projection size fits a realistic usage scenario. The devices communicate over Wi-Fi. As soon as phones are moved such that projections intersect, a merged projection is created. This merged projection can either be a graphically highlighted union of the individual projections, or something different like a shared playboard within the concave hull of the projections' corners.

Applications

Single-user game "escape" (SDSU)

The "escape" game represents the SDSU category by supporting external controls on the projection in a single-user game. The task of the game is to escape monsters by moving the character horizontally and vertically on a play field without other obstacles. When playing the game on the mobile phone, the on-screen controls and finger of the user cover parts of the play field on the phone. By "outsourcing" the controls to the projection behind the phone, thus providing free sight on the whole play field, we assume users will perform better in the projected mode (Figure 4a).

Multi-user presentation (SDMU)

In this application the SurfacePhone is used to present pictures or slides to a small group of people in two different ways: Either the user publishes thumbnails to the projection by dragging the thumbnail on the *proxy* at the top of the phone screen. The audience can then use standard multi-touch techniques for rotating and enlarging the pictures to their will (Figure 3c). The other possibility is that users browse their content on the projection and present items fullscreen on the
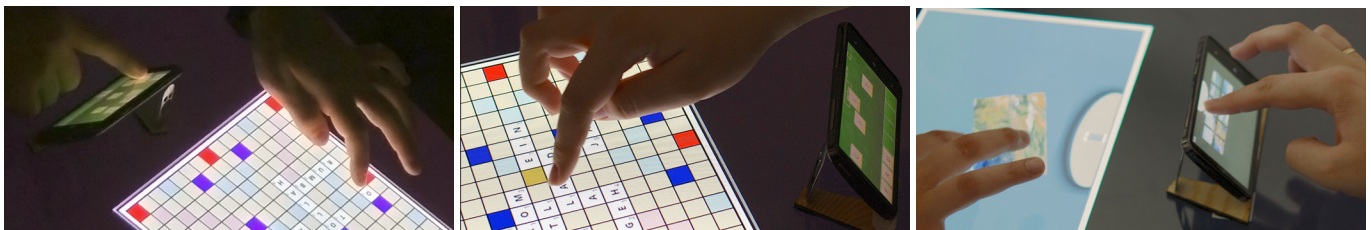

(a) Direct transfer: A user places a scrabble piece at a precise position on the board through simultaneous touch.

(b) Binned transfer: Elements from the bin element (here the bench) on the phone are placed on the projection using *touch-swipe*.
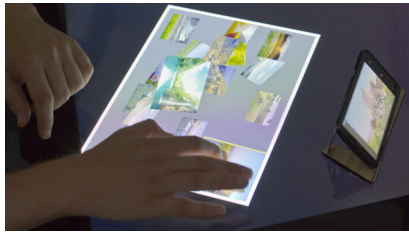
(c) Mediated transfer: The presenter drags another picture on the proxy element at the top of the phone.

Figure 3: Three techniques for content transfer between displays.
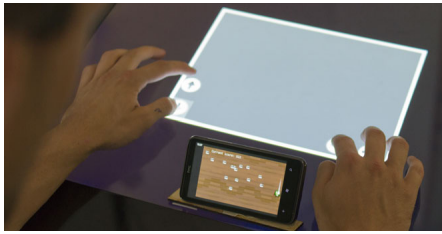
(a) Single-user "Escape" game: The game controls are "outsourced" to address the fat-finger problem with on-screen controls.

(b) Multi-user sharing from projection: the presenter selects images to be displayed on the phone screen.

(c) Multi-device: Both users have pieces for a collaborative puzzle they are supposed to solve on the merged projection.

Figure 4: Three of the five apps used in the concept study (see Figure 3 for the remaining two apps).

phone by double tapping them (Figure 4b). Different to the first way, the user gives up their privacy for the benefit of having a larger space themselves that can be quicker explored.

Multi-device picture sharing (MDMU)
The exemplary picture sharing – which would similarly work with other content types – is very similar to the SDMU presentation application. Users publish their thumbnails to the surface by using the *proxy* or *Human Link* techniques as in the presentation application. As soon as more than one device and user merge their projections by intersecting them, the merged space can be used to share all sorts of personal data. Thumbnails then belong to the joint surface, allowing all participants to explore pictures through multitouch operations and transfer them to their phone using one of the aforementioned techniques. When one of the participating users withdraws from the merged state the merged view is split and the separate projections retain prior items and positions on their side. If items have not been moved to the phone, these items are moved back to the projection of the owner. This feature shall give users a simple means of privacy control as they can withdraw with items that they only want to present but not give away.

Multi-device scrabble game (MDMU)
The scrabble application (Figures 3a and 3b) particularly emphasizes the private display on the mobile phones. It shows a standard scrabble playboard on the merged projections. The phone screens show the letters available to the users and a virtual bench on the bottom where words can be arranged with the letters using drag&drop. On their turn, users either use the *Human Link* technique to place any letter, no matter if on the bench or not, by touching the letter and the target position on the playboard. Alternatively, they first put the letters to place in correct order on the bench and then swipe over the empty fields on the board to place these letters. Depending on whose turn currently is, the board changes its orientation to face the corresponding user. Letters can be taken back to a precise position on the phone using *Human Link* or to a random position by double tapping them.

User Study
With the first user study we assess the quality of the overall SurfacePhone concept and its several components using the presented concept prototype. Using the four aforementioned applications we assess input techniques (e.g. *Human Link* and *proxy*), output (e.g. size and visibility of displays) and possibly occurring problems such as undesired occlusions of the projection and physical demands of the MMDE.

We follow a qualitative approach using think aloud, structured interviews, and video analysis as no similar system is available for comparison. We had 16 participants who took part in pairs to create a more realistic collaborative environment. Their average age was 26 years, (ranging from 23 to 31 years) and six of them were female. All participants except one owned a mobile touchscreen phone and three of the participants had prior experience with multi-touch tables.

Procedure
First we explained the concept of the SurfacePhone by showing them a concept design (similar to Figure 5a) of the technical prototype and to convince them that similar devices can be built we demonstrated a Samsung Galaxy Beam projector phone. Finally, the experimenter briefly explained the prototype, how it works, and the different configurations (SDSU, SDMU, MDMU) which also represented the different phases of the study.

After that, both participants tried all four applications (one each for SDSU and SDMU, two for MDMU) for approximately eight minutes each. Before each application participants were given time to test the concepts relevant in that phase, for instance, merging of projections and different transfer techniques, until they had no further questions. In single-device applications they took turns in acting as user or audience/spectator. In multi-device applications both users operated their own device simultaneously. To ensure a constant learning curve, the order of applications was always the same, going from single-device and single-user to multi-device and multi-user applications, thereby constantly gaining in complexity. Before each multi-user application, users were allowed to choose device positions (see MDMU before) that fit the task according to their opinion.

For the study the participants had to use all aforementioned applications. For the picture presentation applications (SDMU) both participants acted as presenter and observer in turns. For the picture presentation in MDMU mode we added two tasks. One task was to share pictures that contained Waldo with your partner and the other was to solve a 3×3 puzzle collaboratively on the merged projection space.

While participants were continuously motivated to share their experiences aloud, after each configuration (SDSU, SDMU, MDMU) they filled out a questionnaire regarding the configuration and contained tasks. The questionnaire asked for experience with the applications as well as physical demand, fatigue, visibility of content, feelings regarding privacy, etc.

After the study we let participants fill out the Post-Study System Usability Questionnaire (PSSUQ [12]). The study was video captured and later qualitatively analyzed. The textual answers were later analyzed using axial coding.

Results
In terms of preferred interaction technique, to transfer information between the two screens, most participants favored *touch-swipe*. Comparing *touch-swipe* and *Human Link* in the scrabble game, 15 of the 16 participants preferred *touch-swipe*. When comparing *touch-swipe*, *Human Link* and *proxy*, nine participants preferred *touch-swipe* and five would rather use the *proxy* technique. This is also reflected in the physical demand. Ten participants stated that *Human Link* has the highest physical demand and four found *proxy* to have the highest demand. The same was reflected in their rating of success, 12 participants said they were most successful with the *touch-swipe* and two thought they would be better with either *Human Link* or *proxy*. In their comments the poor performance of *Human Link* became pretty obvious. A majority (9 out of 16) stated that when using the *Human Link* they had the problem that when they touched the phone's screen and tried to touch the projection the phone would slide away. This was especially seen as an advantage of *touch-swipe*.

We asked the participants whether they developed a strategy to solve the puzzle and image tasks. All participants agreed that they followed a certain strategy. From the video analysis and the comments two strategies were particular promising. Three couples would actually change their original sitting position so that they would sit next to each other, allowing both participants to view each others phone displays, helping them to identify the correct pictures before putting them on the surface. Three other couples divided the work between in each other so that one participant would move the pictures from the phone to the projection and the other would arrange the puzzle parts in the projection.

To evaluate a possible adoption of such a device we asked the participants whether they would recommend such a device to their friends and if so what would be the necessary circumstances. All but one participant answered that they would recommend it. Most of the participants found fulfillment of hardware constrains such as reasonable size and battery life to be mandatory. Privacy was desired by all participants as a needed feature and most of them thought that presenting pictures by selecting them on the phone screen and have the projection show them is more reasonable than the other way round. Ten participants stated that they like that they don't need to pass their device around when presenting to groups.

The preferred combination of devices and users was MDMU. Participants found the possibilities that arise from having a mobile device that ad-hoc can create a complex mobile multi-display environment very attracting. Besides games such as Battle Ships, Poker and Black Jack the collaborative editing of documents, e.g. layouts of newspapers, was seen as possible application scenarios. Two participants mentioned the case of ad-hoc meetings for example to collaboratively investigate construction plans on a construction site.

The results of the PSSUQ are underlining these results. In the overall usability rating the SurfacePhone scored 84.8% as a mean of system usefulness (87.3%) and interface quality (81.9%). Overall, the results of the PSSUQ indicate that the SurfacePhone is a useful new device to extend screen space in single- and multi-user applications.

TECHNICAL PROTOTYPE
The positive results of the first study motivated us to build a technical SurfacePhone prototype that can support aforementioned interactions. Our aim with this prototype is twofold: Firstly, we want to investigate the technical requirements and challenges for such a device and find solutions for them. Secondly, we want to conduct another user study that delivers quantitative measures how well touches and gestures can be performed by users and detected by the system. Additionally we want to provide everything needed to create a SurfacePhone prototype to the community. From this we hope to spark further discussions.

Hardware Design
As no similar device configuration has been presented so far, we started from scratch. We chose the iPhone platform, since it was the only mobile platform that allowed two different outputs on screen and projection at that time. After we tested several different projection engines (e.g. TI DLP 2 or Microvision PicoP) attached to the backside of the device, it became obvious that without a fitting short-throw lens that is not available, the size of the projection would become too small. We solved the problem by attaching a mirror to the top of the phone and the projector to its bottom. This way the distance from projector to surface is more than doubled and sufficient to create a projection much larger than the phone screen.

Our prototype consists of an iPhone 5 plus projector case as depicted in Figure 5a. Both projector and mirror are 4cm to the right of the iPhone camera which is the minimum distance required for projector and mirror not to appear in the wide-angle view of the phone camera. In the camera image that is sampled at 640×480 the projection appears between $P_{TopLeft}\{183, 238\}$, $P_{BottomRight}\{590, 316\}$, thus takes up 407px in X and 78px in Y direction. Obviously, specifically the resolution in Y direction is quite small and the projection is not centred in the image. Nonetheless, this is the best compromise we found between maximizing the size of the projection and still completely seeing the projection in the camera.

Implementation
Following our previous design considerations, the prototype should support direct touch on the projection, different touch modalities, gestures, and tracking of other nearby SurfacePhones.

The software of the SurfacePhone is implemented in Objective-C and C++ on iOS with the help of OpenCV and openFrameworks modules. First we calibrate intrinsic and extrinsic camera parameters using printed chessboard and projected chessboard patterns respectively. Having the parameters we can map the projected area from object space to an interpolated orthogonal view of the projected region (see Figure 6a) and use this for tracking. In a final SurfacePhone device this would only have to be performed once.
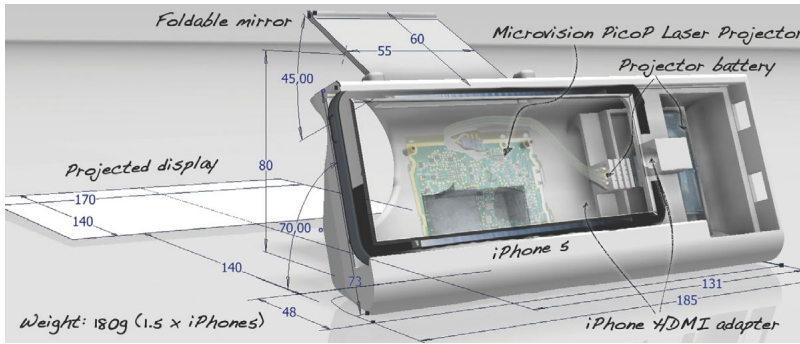
(a) The technical prototype design (dimensions in mm).

(b) The implemented prototype.

Figure 5: Design and instantiation of the SurfacePhone prototype that tracks finger touches using in-built camera and accelerometer. The red border (for illustration) in (b) is the relic of perspective counter distortion. The phone shows raw camera image (top-left), background-image (bottom-left) and finger tracking with green dot at recognized fingertip (top-right).

For the tracking to work robustly at arbitrary locations we must make sure that different lighting conditions are handled. We can let the iPhone automatically adjust exposure and focus of the camera to the center of the image to adapt to different conditions. However, we need the user's finger for a correct estimation. Therefore, in step 3, we ask the user to present their finger for 2 seconds to the center of the camera while we lock correct exposure and focus for future interaction.

Finger Position
In step 4, we capture a still frame for subsequent background subtraction. As the background of interaction can be arbitrary we use background subtraction to separate moving fingers from the background. This step is automatically performed whenever the device comes to rest on a plain surface. Since we constantly measure the accelerometer at 100Hz we can quickly recognize whenever the user starts and stops moving the device.

Our following tracking process runs at 22 FPS. To eliminate shadows as best as possible we first convert the camera image to the HSV space and then work on the saturation channel. The literature recommends working on the hue channel to eliminate shadows, but we found that desk colors are often very similar to skin colors which is why we use the saturation channel that works more reliably in our scenario. After background subtraction we find blobs using openCV's contour finding algorithm. Because of the steep camera angle in our setup, blob area sizes can range from a few pixels to sizes that fill half of the image. This makes the classification of correct blobs more difficult. Further, we cannot rely on standard CV techniques like convexity defects for finding fingertips as often there is only one finger plus parts of the thumb in our image which do not provide the defects information. Instead, the algorithm we developed first computes the convex hull of the contour and its normalized approximation. Then, for each point on this new contour that has a tangential slope of less than $15°$ with its surrounding points, it calculates a probability that this point is the fingertip by minimizing equation:

$$P(Fingertip) = \frac{(SCD * W_{SCD} - DCP * W_{DCP} + SA)}{AREA} \quad (1)$$

- where SCD is the second closest distance of the point to the corners of the bounding box. A finger should create a very rectangular bounding box where the fingertip lies almost at the center of the smaller side of the rectangle yielding small distances to the two closest corner points;
- where DCP is the summed distances between the point and corner points of the bounding box that lie on or outside of the edge of the camera frame. A correct fingertip of a pointing finger will always have maximum distance to the hand center. As we do not see the hand the corner point is only an average guess;
- where SA is the estimated area of the fingertip above the current point. This is calculated as the sum of distances between up to 15 surrounding points on the contour to both sides. Correct fingertips should yield smaller results than arbitrary shapes with peak endpoints;
- where $AREA$ is the size of the blob;
- and where W_{SCD} and W_{DCP} are weights found by experiment set to 10 and 5.

Finally we have to decide which blob represents the primary finger, which one is a possible second finger and which ones are not of interest. Our blob sorting and filtering algorithm favors blobs with fingers, high finger probabilities, less circular shape (to filter out hand areas) and lower Y position (to filter out shadows appearing below fingers).

Finger Touch and Touch Modalities
To support different *touch modalities* we cannot rely on the 2D camera image as small changes in depth are indistinguishable from small changes in height for touch recognition. Kane et al. in their Bonfire system used a combination of posi-
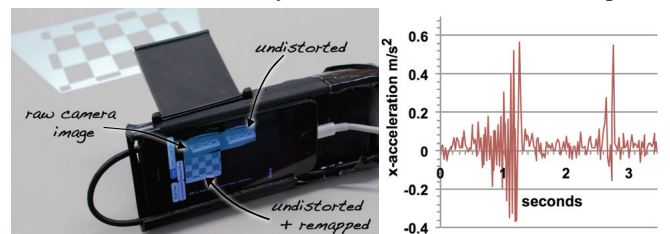
(a) Calibration proof mode: The projector projects a chessboard, the screen shows different proof views.

(b) First a strong, then a light touch, logged on the Surface-Phone.

Figure 6: Implementation details

tion tracking using the camera and touch recognition using an accelerometer that measures the touch vibration on the table [10]. This approach seems the most promising as the hardware is readily available in most (projector) phones – in contrast to e.g. [6, 19]. However, their camera angle was almost orthogonal which simplified finger tracking. Further, they used the accelerometer of a laptop that shares a much larger space with the surface than our prototype and therefore allowed to work with simple thresholds. Correctly classifying touches with the SurfacePhone is thus bigger challenge.

Furthermore, when relying on surface vibrations for touch detection we cannot distinguish between touch down and up events to, for example, classify a touch as "long touch". Similarly, double taps cannot be recognized as successive events since the second vibration could overlay the first one. We can, however, recognize the intensity of the touch quite well since light and strong touches create distinct vibration patterns (see Figure 6b). These two modalities, light and strong, can, for instance, be used to start dragging of an item using a strong touch.

For the touch detection we measure the current acceleration in X direction every 10ms. Then we compute the touch vibration as the difference between the averaged sum of the absolute amplitudes of the recent 150ms (15 values) and the previously calibrated sensor noise. Based on thresholds we then decide whether the measured vibration corresponds to a strong, a light, or no touch at all. The default strong threshold is twice the default light threshold. As not all surfaces transport vibrations equally, we also implemented a detection procedure that vibrates the SurfacePhone with a constant pulse and measures the resulting phone vibration. Based on our tests with different tables, lightweight tables will be good mediums resulting in low phone vibrations (down to $0.1m/s^2$) and good touch recognition whereas strong tables will not pass on the vibration very well, resulting in phone vibrations up to $0.5m/s^2$. Through this procedure we can adjust the default thresholds to increase touch accuracy on different tables.

Gesture Recognition
Since in our setup we only see small parts of the user's hand, gesture support of hand postures does not make much sense. However, we can well recognize gestures that are based on a trajectory of movement such as directional gestures (left, right, top, down swipes) or more complex gestures like a circle. Finger trajectories that do not end in touches are simply analyzed for long directional movements or otherwise handed to the 1$ gesture recognizer by Wobbrock et al. [22].

Detection of other SurfacePhones
Although not fully integrated into our prototype yet, we evaluated the use of Qualcomm's Vuforia on the SurfacePhone using *projected* frame- and image markers. Our interest was to see how the steep camera angle and the much lower resolution of the projected image compared to printed markers affects the recognition algorithm. Fortunately, the recognition worked better than expected. Frame markers of a size of $1/8$ of the projection size are perfectly recognized as soon as they are completely visible in the scene. Image targets are recognized even up to a $1/50$ of the projection (see video figure).
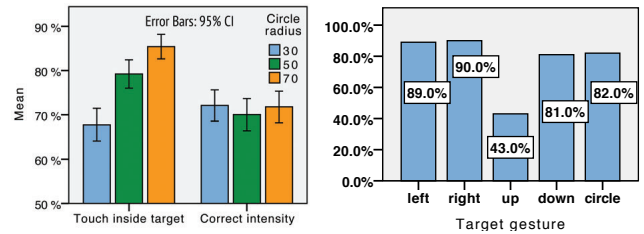
(a) Touch performance.   **(b) Gesture performance.**

Figure 7: Results of the second study.

Technical evaluation
For the technical evaluation we recruited 15 untrained right-handed participants (5 female) of an average age of 27 years (ranging from 22 to 65 years).

Evaluation procedure
In the first part of the study, we asses multi-modal finger touch. The projection shows circles the user has to touch. There are three different sets of circle radius (30, 50, 70px respectively 1.75, 2.9, 4.0cm physical diameter) that each are arranged in a grid of 3×4 circles (cf. Figure 8) that span the projection area. Additionally, each target exists once for light and once for strong touch, marked with a big L or S (see Figure 1e). Thus, in total there are $3 \times 12 \times 2 = 72$ different targets split across three circle groups. After a test round, each participant performs 2 successive study rounds (resulting in 2160 touches overall). The order of circle size sets is counterbalanced and the display order of targets randomized. Participants can take as much time as they need to perform the touch.

In the second part of the study participants performed the gesture (circle, swipe left, right, top, down) that was written on the projection. Again, users performed one test and two study rounds of 4×5 gestures in random order (resulting in 500 recorded gestures overall).

Results and Discussion
Overall, 93% of user touches were recognized (7% have been performed too light to be recognized), 71% of these were hit with the right intensity and 77% of targets at the right position (see Figure 7a). Furthermore, we measured that clearly misclassified fingertips ($\delta > 300px$ off the target center) have been responsible for about 12% of false position recognition.

Factorial repeated-measures ANOVA on the touch data reported significant main effects of *circle radius*, *target position in Y direction*, but not *target intensity* at the $p < .05$ level (Greenhouse-Geisser corrected) on positional accuracy. Post-hoc analysis using Bonferroni corrected pairwise comparison of means revealed significant differences ($p < .05$) between small and middle and small and large sized circles as well as target heights 140px/340px and 240px/340px. Thus, the larger targets have been and the farer they have been away from the device, the better they have been hit in terms of position. Touch intensity recognition is statistically independent of both target position and circle radius.

Left and right gestures yielded recognition rates around 90%, down and circle gestures around 80%. Only the up gesture performed significantly worse than all others with only 43% (Figure 7b). The reason for this is that the tracker confused
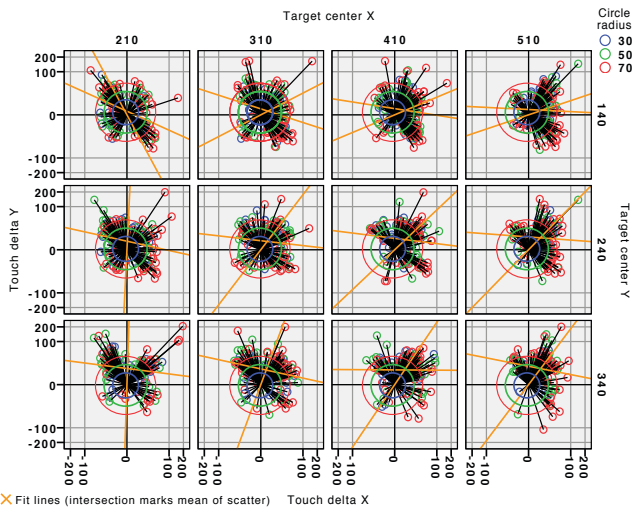
Figure 8: Participants touches visualized by target position and target radius. The orange fit line indicates general under-/overshooting.

44% of up gestures with the circle gesture. This may be due to the fact that after performing the correct up gesture participants moved their hand down and to the right to their default position, which the recognizer that evaluates the gesture after the hand has left the frame may have misinterpreted.

We also wanted to know if users would show a similar over- and undershooting behavior as they do on touchscreens (cf. [7]). Figure 8 depicts all performed touches except for the far outliers ($\delta > 300px$). From the touch distribution and their mean marked by the crossing of the fit lines we can conclude that the target position indeed has an effect on over-/undershooting in both X and Y direction. Targets to the lower-right are more overshot than targets to the upper left. However, overshooting is only compensated for farer targets without transforming into obvious undershooting as on touchscreens [7]. We assume the reason for this is that due to the steep viewing angle of the user on the projection the fat-finger problem only exists close to the device and decreases with increasing distance from the device. Also, perspective misjudgment may counter-balance overshooting. The issue of overshooting may thus also explain the significant effect of Y direction on touch accuracy mentioned before.

Regarding personal experiences, all participants thought that the device is already usable in many scenarios but maybe not for tasks like text entry (3 participants). Similarly, 10 participants stated that the difference between light and strong touches was difficult to learn, maintain (especially after performing the same intensity multiple times before switching), or to perform. For three female users the threshold for strong touch was set too high, for two male users rather too low. Overall, light touches have been slightly but significantly better ($p < .05$) recognized than strong touches (75% vs. 68%).

Overall, the results of the exploration and study of the second prototype reveal that a working SurfacePhone with the features (touch, drag&drop through multi-modal touch, gestures, and merging of projections) used in the first study can be built with today's mobile phone hardware. These findings underpin the overall concept but also show that the restriction to standard mobile phone hardware leaves room for the

improvement of the system's accuracy. In the following we present suggestions towards that matter.

LESSONS LEARNED AND GUIDELINES
Through our explorations and studies we learned the following lessons and concluded guidelines for the SurfacePhone and its interface design:

1. In our study all participants cherished having the private display. Therefore, applications designed for the Surface-Phone should always consider its proper usage in innately not very private mobile scenarios.
2. Where possible, interaction techniques should be used that do not require simultaneous interaction on both displays. Besides the visual separation, though small, the user has no hand available to keep the phone in place.
3. The *proxy* was seen as an advantage in the SDMU case where the screens are divided between the users and nobody wanted to intervene on the display of the other. In multi-device scenarios with merged and thus larger displays, transfer techniques that support precise placement (like *touch swipe* and *human link*) should be favored.
4. Interface elements placed to the top left side of the projected display will not be affected by the fat-finger problem. Specific to our implementation, interactive elements should also have a radius of at least 50pixel to ensure a good accuracy.
5. Users should receive feedback about their touch intensity (e.g., a color meter around their touch) to support their mental touch model. Further, touch thresholds should be personally adaptable to account for anatomic differences.
6. Multi-modal touch decreases the accuracy of touch recognition as sometimes users touch too light while they try to keep the intensity below the threshold for strong touches. Thus multi-modal touch should be disabled whenever the interface gets by with single touch plus gestures to increase the accuracy to at least 93%.
7. The proposed automatic calibration of touch thresholds only makes sense up to the physical limits of surface vibrations. For thicker surfaces this especially means that light and strong touch thresholds move closer together, possibly resulting in more falsely recognized strong touches. Thus, strong surface materials (e.g. stone) should be avoided.
8. The system should account for overshooting with an offset function to increase accuracy of finger position tracking.

CONCLUSION AND FUTURE WORK
In this paper we presented the SurfacePhone, a novel configuration of a physical display and a projector that are aligned to allow ad-hoc Tabletop interaction in a mobile setup. We explored its design space and identified new single- and multi-user application scenarios with tailored interaction techniques. Our evaluation using our concept prototype indicates that the device is suited for a variety of everyday scenarios. We also learned which setups and transfer techniques users preferred in different scenarios. Later we demonstrated how the SurfacePhone can be built with only today's commodity phone hardware and the help of a specialized case and state-of-the art techniques for finger tracking and multi-modal touch recognition. Results of a quantitative user study on touch and gesture tracking accuracy revealed that the present

prototype would already be applicable to many single- and multi-user scenarios and how it could be further improved, for instance, by counterfeiting typical overshooting behavior and personal adaptation of touch intensity. Overall, we can conclude that the SurfacePhone makes for a new and interesting device for ad-hoc collaboration.

In this work we concentrated on re-creating Tabletop scenarios with personal mobile data. However, there is a large potential of mobile scenarios that include movement of devices on or off the table as interactions. For instance, could the phone projection be used as peephole on a large desktop in joint learning scenarios, or games could involve tasks of which some can better be performed while holding the phone in hand and others better while placing the phone on the table. We leave this for future work, likewise further improvements to our technical prototype.

The SurfacePhone software, STL print files of the hardware, and assembly instructions are available for download http://uulm.de?SurfacePhone.

REFERENCES
1. Bachl, S., Tomitsch, M., Kappel, K., and Grechenig, T. The effects of personal displays and transfer techniques on collaboration strategies in multi-touch based multi-display environments. In *Proc. of*, INTERACT'11, Springer (2011), 373–390.

2. Cauchard, J. R., Löchtefeld, M., Irani, P., Schoening, J., Krüger, A., Fraser, M., and Subramanian, S. Visual separation in mobile multi-display environments. In *Proc. of*, UIST '11, ACM (New York, NY, USA, 2011), 451–460.

3. Chen, N., Guimbretiere, F., Dixon, M., Lewis, C., and Agrawala, M. Navigation techniques for dual-display e-book readers. In *Proc. of*, CHI '08, ACM (New York, NY, USA, 2008), 1779–1788.

4. Greenberg, S., Boyle, M., and Laberge, J. PDAs and shared public displays: Making personal information public, and public information personal. *Personal Technologies 3*, 1-2 (Mar. 1999), 54–64.

5. Hang, A., Rukzio, E., and Greaves, A. Projector phone: a study of using mobile phones with integrated projector for interaction with maps. In *Proc. of*, MobileHCI '08, ACM (New York, NY, USA, 2008), 207–216.

6. Harrison, C., Benko, H., and Wilson, A. D. OmniTouch: wearable multitouch interaction everywhere. In *Proc. of*, UIST '11, ACM (New York, NY, USA, 2011), 441–450.

7. Henze, N., Rukzio, E., and Boll, S. 100,000,000 taps: analysis and improvement of touch performance in the large. In *Proc. of*, MobileHCI '11, ACM (New York, NY, USA, 2011), 133–142.

8. Hinckley, K., Dixon, M., Sarin, R., Guimbretiere, F., and Balakrishnan, R. Codex: a dual screen tablet computer. In *Proc. of*, CHI '09, ACM (New York, NY, USA, 2009), 1933–1942.

9. Hinckley, K., Ramos, G., Guimbretiere, F., Baudisch, P., and Smith, M. Stitching: pen gestures that span multiple displays. In *Proc. of*, AVI '04, ACM (New York, NY, USA, 2004), 23–31.

10. Kane, S. K., Avrahami, D., Wobbrock, J. O., Harrison, B., Rea, A. D., Philipose, M., and LaMarca, A. Bonfire: a nomadic system for hybrid laptop-tabletop interaction. In *Proc. of*, UIST '09, ACM (New York, NY, USA, 2009), 129–138.

11. Kauko, J., and Häkkilä, J. Shared-screen social gaming with portable devices. In *Proc. of*, MobileHCI '10, ACM (New York, NY, USA, 2010), 317–326.

12. Lewis, James R. IBM computer usability satisfaction questionnaires: Psychometric evaluation and instructions for use. Technical 54.786, Human Factors Group.

13. Lyons, K., Pering, T., Rosario, B., Sud, S., and Want, R. Multi-display composition: Supporting display sharing for collocated mobile devices. In *Proc. of*, vol. 5726 of *INTERACT '09*. Springer, Berlin, Germany, 2009, 758–771.

14. Matsumoto, T., Horiguchi, D., Nakashima, S., and Okude, N. Z-agon: mobile multi-display browser cube. In *Extended Abstracts of*, CHI '06, ACM (New York, NY, USA, 2006), 351–356.

15. Molyneaux, D., Izadi, S., Kim, D., Hilliges, O., Hodges, S., Cao, X., Butler, A., and Gellersen, H. Interactive environment-aware handheld projectors for pervasive computing spaces. In *Pervasive Computing*, vol. 7319. Springer, Berlin, Heidelberg, 2012, 197–215.

16. Schmidt, D., Seifert, J., Rukzio, E., and Gellersen, H. A cross-device interaction style for mobiles and surfaces. In *Proc. of*, DIS '12, ACM (New York, NY, USA, 2012), 318–327.

17. Seifert, J., Simeone, A., Schmidt, D., Holleis, P., Reinartz, C., Wagner, M., Gellersen, H., and Rukzio, E. MobiSurf: improving co-located collaboration through integrating mobile devices and interactive surfaces. In *Proc. of*, ITS '12, ACM (New York, NY, USA, 2012), 51–60.

18. Shirazi, A. S., Döring, T., Parvahan, P., Ahrens, B., and Schmidt, A. Poker surface: combining a multi-touch table and mobile phones in interactive card games. In *Proc. of*, MobileHCI '09, ACM (New York, NY, USA, 2009), 73:1–73:2.

19. Wilson, A. D. PlayAnywhere: a compact interactive tabletop projection-vision system. In *Proc. of*, UIST'05, ACM (New York, NY, USA, 2005), 83–92.

20. Wilson, A. D., and Benko, H. Combining multiple depth cameras and projectors for interactions on, above and between surfaces. In *Proc. of*, UIST '10, ACM (New York, NY, USA, 2010), 273–282.

21. Winkler, C., Seifert, J., Reinartz, C., Krahmer, P., and Rukzio, E. Penbook: Bringing pen+paper interaction to a tablet device to facilitate paper-based workflows in the hospital domain. In *Proc. of*, ITS '13, ACM (New York, NY, USA, 2013), 283–286.

22. Wobbrock, J. O., Wilson, A. D., and Li, Y. Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes. In *Proc. of*, UIST '07, ACM (New York, NY, USA, 2007), 159–168.