Personal and Ubiquitous Computing manuscript - DOI 10.1007/s00779-009-0246-3
(The original publication is available at `http://www.springerlink.com/content/0873v10144j12801/`.)

Supporting Casual Interactions between Board Games on Public Tabletop Displays and Mobile Devices

Florian Echtler, Simon Nestler, Andreas Dippon, Gudrun Klinker

Technische Universität München
Institut für Informatik I16
Fachgebiet Augmented Reality
Boltzmannstr. 3
D-85747 Garching, Germany
{echtler|nestler|dippona|klinker}@in.tum.de

Received: 10 October 2008 / Accepted: 15 March 2009

Abstract As more interactive surfaces enter public life, casual interactions from passersby are bound to increase. Most of these users can be expected to carry a mobile phone or PDA, which nowadays offers significant computing capabilities of its own. This offers new possibilities for interaction between these users' private displays and large public ones.

In this paper, we present a system which supports such casual interactions. We first explore a method to track mobile phones that are placed on a horizontal interactive surface by examining the shadows which are cast on the surface. This approach detects the presence of a mobile device, as opposed to any other opaque object, through the signal strength emitted by the built-in Bluetooth transceiver without requiring any modifications to the devices' software or hardware.

We then go on to investigate interaction between a Sudoku game running in parallel on the public display and on mobile devices carried by passing users. Mobile users can join a running game by placing their devices on a designated area. The only requirement is that the device is in discoverable Bluetooth mode. After a specific device has been recognized, a client software is sent to the device which then enables the user to interact with the running game. Finally, we explore the results of a study which we conducted to determine the effectiveness and intrusiveness of interactions between users on the tabletop and users with mobile devices.

1 Introduction

With interactive surfaces, a novel and interesting field of research has gained increased attention in recent years. These devices are particularly suited for the exploration of new user interface paradigms, especially those that require little to no prior knowledge of the system. Such interfaces are therefore well adapted to being deployed in public scenarios, where many casual users will be using the device for relatively short periods of time, sometimes even concurrently.

Nowadays, most people who would use an interactive surface can be expected to also carry a mobile phone with them. Current models have many advanced features, such as Bluetooth, a camera or 3D graphics and are able to run custom programs. It is therefore reasonable to assume that, given the opportunity, users who interact with a tabletop device would also like to use their mobile phone in conjunction with the interactive surface.

Games are among the most common kinds of casual interaction between people. Examples include the venerable chess pieces in public parks, playgrounds, cafes which specialise in board games and many more. Especially those games which benefit from collaboration between several users can be highly attractive in a public setting and are therefore prime candidates for being deployed on a public interactive surface. When considering the fact that games are also the most popular application for mobile devices, it becomes apparent that a combination of the two interaction modalities could offer a large entertainment potential.

One popular game which can be played collaboratively is Sudoku, in which a grid has to be filled with symbols according to a set of rules. Although it is originally not a board game, it can easily be played as one when the free symbols are placed on tiles which players can move into free grid slots. We chose this game as it represents a search task that can be used to model other application-specific problems. While Sudoku is not a highly complex game, it nevertheless presents some challenges to building a suitable user interface - or indeed two of them, as the completely different environments of interactive surface and mobile device require individual solutions.

In this paper, we present an approach for easy and natural interaction between users which collaboratively play a Sudoku game. Per default, the game is running on a public interactive table and can be played by several users simultaneously. When users prefer to play not directly at the surface, but would rather join the game with their mobile device, they can do so by shortly placing the device on a designated "join" area on the table. One important aspect is that the mobile device does not have to be modified in any way; the only requirements are an active Bluetooth transceiver and the ability to execute a custom application.

2 Related Work

Interaction with physical devices on display surfaces is a concept that has been extensively examined, e.g. by Ishii [14] and Greenberg [10]. Most of these systems rely on an optical tracking modality, often a rear-mounted camera.

Tracking of devices via Bluetooth has already been investigated, e.g. by Hallberg et al. [11] and Castano et al. [3]. These systems rely on measuring the signal strength of an established Bluetooth link. One severe drawback of this approach is that the time to open a Bluetooth connection can take on the order of tens of seconds, which is longer than most people will be prepared to wait.

Along with the increasing interest in tabletop interfaces, the integration of mobile phones into such systems has also been investigated by several researchers. One widely published example is the Surface system from Microsoft [18], which recently has been deployed in a mobile operator's store to provide information about mobile phones placed on top. Our work has been inspired by this system. However, the Surface relies on special tags (either optical or RFID) that have to be attached to the phones to be recognized.

One interesting system which also combines an interactive surface with tracking of mobile devices is BlueTable by Wilson et al. [23]. This setup is based on the PlayAnywhere system [22] and uses a short-throw projector and wide-angle infrared camera to convert any flat surface, such as a regular table, into an interactive surface. While the devices tracked in this setup do not have to be modified physically, they still require prior installation of a software which allows remote control of the screen or the infrared port.

As the interaction on the surface itself is supposed to support multiple concurrent users, a multi-touch interface is needed. Multi-touch technologies for public displays have first been developed by Lee et al. in 1985 [16]. The multi-touch table top which we employ for our application is based on the concept of frustrated total internal reflection (FTIR) which was proposed by Han [12].

Previous research on coupling mobile hand-held devices with public displays has been performed by Greenberg et al. [9,8]. Their approach combines hand-held devices with personal information and large displays with public information. During a real-time meeting the participants can share personal information and modify all public information. Carter et al. [2] propose a combination of public displays with hand-held devices for public annotation of multimedia content. They use hand-held devices to augment, comment and annotate public content which is displayed on public displays. In the healthcare domain, public and private displays are used by Favela et al. [7] to support the decision making of doctors and nurses with mobile computing technologies. Semi-public displays for collaboration within smaller groups have been developed by Huang et al. [13]. Their concept focuses on sharing information on activities within certain user groups. Information shared by group members is not fully public, it can only be viewed and modified by group members. In the context of pervasive gaming, the STARS system by Magerkurth et al. [17] is of interest. It aims to offer a central game board on a tabletop display along with the possibility to add PDAs as additional input devices.

An approach where mobile phones are employed as authentication tokens has been examined by Schöning et al. [20]. By triggering the flash of the mobile phone upon request from the server controlling the interactive surface, a bright spot can be observed through the rear-mounted camera, thereby momentarily locating a specific phone on the surface. Again, a Bluetooth link has to be established and special software installed beforehand.

While a significant amount of work on combining mobile devices with interactive surfaces exists, all these setups share the common trait that prior to interaction a preparation step of some sort is needed, hindering truly casual interaction. In contrast to these systems, our setup aims to support unmodified phones as they are carried by everyday users.

3 Tracking Mobile Phones on Interactive Surfaces

As the ability to identify and track the mobile device on the surface is a prerequisite for our desired mode of interaction, we shall first consider the hard- and software setup which is needed for this task.

3.1 Hardware Setup

The central element of our system is an FTIR-based multi-touch table. The table is also equipped with an infrared shadow tracker. In this scenario, it will be used to determine the location of devices on the surface. While the additional hardware increases total complexity, it is nevertheless necessary as a means to track objects on the surface. As mobile phones are likely to have a hard plastic shell which does not show up on the FTIR screen, a second tracking mode is needed. While a soft silicone layer might also be used to this end, the shadow tracking has the added benefit of differentiation between a phone and, e.g., the user's palm, which also generates an FTIR response in addition to a shadow. Moreover, the shadow-based approach can also be adapted to an "inverted" setup with an overhead infrared camera. An overview of the system is given in Figure 1.

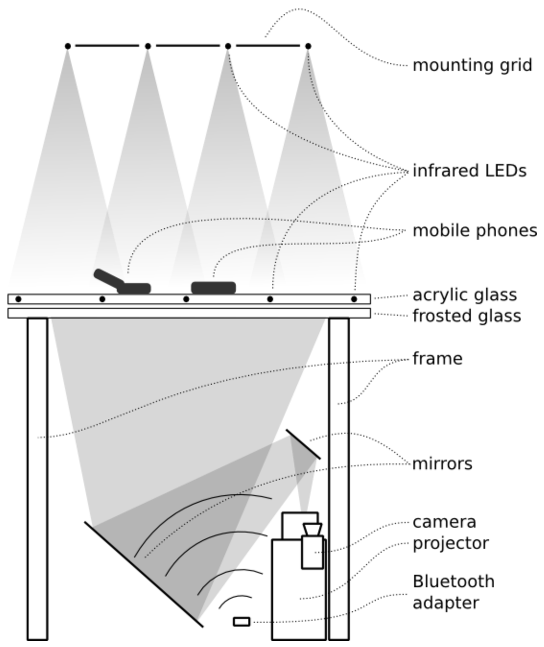


Fig. 1: Hardware setup: interactive table with overhead shadow tracker.

Note that although the primary mode of interaction in this scenario happens through the shadow tracker, the touch screen is still functional. As the rear-mounted camera generates 60 frames per second, it is possible to alternate between the shadow tracker and the FTIR surface every other frame while still providing a smooth user experience. For a more detailed description of the hardware design and the various issues which had to be solved, see our previous paper [4].

To perform proximity detection on the mobile phones and exchange data with them, the integrated computer is equipped with a Bluetooth adapter. As proximity sensing is performed via the Received Signal Strength Indicator (RSSI), this information should be available with low latency. Therefore, we selected a Broadcom USB adapter that supports the "inquiry with RSSI" feature which was introduced in version 1.2 of the Bluetooth Core Specification [1]. This allows us to have the adapter continuously run inquiry scans while at the same time getting RSSI data on all discoverable devices within radio range. One scan cycle takes about 1 second as opposed to older Bluetooth devices that do not have this feature. These adapters have to issue a time-consuming connection request (up to 11 seconds in the worst case) for every single RSSI measurement. Moreover, this connection requires pairing the mobile device and the adapter for which the user has to enter a PIN code. In contrast, our system is able to function without explicit user interaction.

Of course, this very fact raises some questions regarding security and privacy. One might argue that persons who allow a personal Bluetooth-enabled device to be discoverable know what they are doing. However, most people are unaware of the implications. For example, certain phones are vulnerable to a wide range of Bluetooth-based attacks when they are in discoverable mode [21]. It might therefore be advisable to remind the user through a message on the interactive surface that the Bluetooth transceiver should be turned off after use. Care should also be taken not to compromise the users' privacy by generating a log of detected devices. Although the software does store a list of device addresses, this list is not timestamped and never saved to disk and should therefore be unproblematic regarding privacy.

3.2 Software Design

Our setup has two data sources: shadow location data from the optical tracking system as well as RSSI values from the Bluetooth adapter. The optical tracking layer processes the raw image data from the infrared camera and provides a high-level abstraction. After background subtraction, thresholding and segmentation, every blob is analyzed with respect to its size, location of centroid and major/minor principal axis. To determine this data, the blobs' central moments of first and second order are calculated. The blob is also assigned an numerical tracking identifier. This identifier stays with the blob as it moves over the surface by calculating the motion vector of the blob from the previous frame and matching it with the blob which is closest to the predicted location in the next frame. This data is then transmitted via a network interface to applications or higher-level trackers (see also Figure 2a).

Our tracking software is composed of two threads. The first one continuously collects RSSI data from Bluetooth devices within reception range, while the second one receives and processes the optical tracking data. The first step in this thread is to differentiate between shadows that are really cast by mobile phones and those cast by other objects, such as the users' hands.

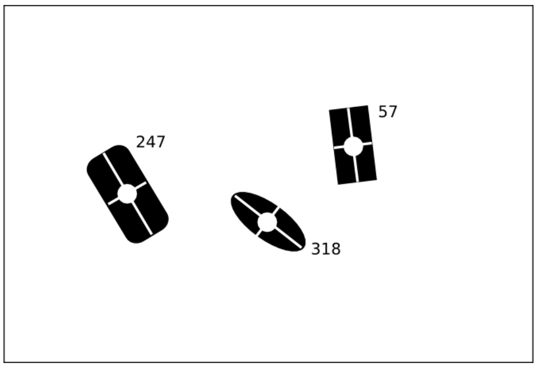
One obvious and easily applied criterion is blob size. There are upper and lower bounds on the surface area which a mobile phone covers, as it is usually roughly pocket-sized. For our setup, we have experimentally determined these bounds to be at 2000 and 10000 square pixels, respectively. These values depend on camera resolution and size of the surface. In our case, a camera with a resolution of 720 x 576 pixels is viewing a surface area of 1.15 x 0.75 m, resulting in a covered area of approximately 2 mm^2 per pixel. While the area range may seem large, it was chosen to account for, e.g., the difference between open and closed clamshell phones. In practice, these values have proven to be sufficient to include every phone we have placed on the surface while filtering out other objects such as a user's arm.

The second criterion which we examine is blob motion. Before a phone can be reliably recognized, it should remain motionless on the table for one second, as this is approximately the duration of one inquiry scan cycle. Our software examines the blob position for the last 30 frames and calculates its standard deviation. If it falls below a threshold of 2 pixels, the blob is considered a candidate for a mobile phone.

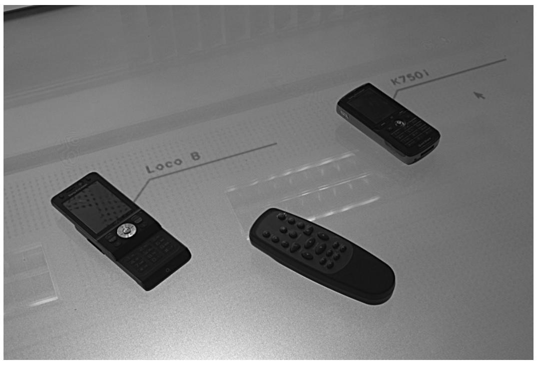
The next step is to correlate these candidates with the proximity data from the Bluetooth thread. The RSSI measurements are usually returned in dBm.[1] The values typically range between -40 dBm for close proximity and -90 dBm at the limits of reception range. Obviously, these values are dependent on the mobile phone as well as the Bluetooth adapter in use. As our adapter is mounted at a distance of approximately 80 cm below the tabletop, a phone lying on the surface generates RSSI values of about -60 dBm. Therefore, we use a proximity threshold of -65 dBm to determine whether a phone is on or near the table surface. Although the distance to a mobile phone which is carried in the pocket of a person standing beside the table is about the same, the RSSI values for such phones are significantly lower. This is due to the non-uniform reception pattern of the dipole antenna which is used in almost all Bluetooth dongles. Such an antenna usually exhibits several distinct lobes with high reception sensitivity. In our case, the antenna is oriented so that the main lobe points straight upwards, thereby favoring phones located on the surface and not those beside the table.

Finally, the list of phone candidate blobs and Bluetooth devices can be compared. In an ideal case, there is one unassigned blob and one newly detected device in range, which makes the assignment trivial. In this case, the optical characteristics (size and length of major and minor axis) of the blob are also stored along with the Bluetooth data. This can be used for later identification of devices if ambiguities arise. For example, a candidate blob can appear without suitable Bluetooth devices in range. This can occur when the discoverable mode of a phone has a fixed timeout. In this case, the blob features are compared with the list of previously recognized devices. The best-fitting match according to a squared-error measure is then used to match the blob with Bluetooth data. Also, if several blobs and Bluetooth devices appear simultaneously, the blob features are compared with previous matches to resolve this ambiguity. Therefore, this method is currently unable to differentiate between several previously unknown objects which are placed on the surface in a short timeframe (< 1 sec.). For additional details on the discovery and matching process, see our previous paper [6].

One important aspect of this concept is that it is not limited to FTIR-based interactive tables. Any system



(a) Data provided by the shadow tracker: tracking ID, location of centroid, size of shadow area and major/minor principal axis. Roundness is implied by size/axis ratio.



(b) Two mobile phones are automatically annotated with their Bluetooth names. One non-Bluetooth object on the table surface is ignored.

Fig. 2: Shadow tracker data and resulting name/location assignment

[1] Even though this value might not reflect the true received signal power, but rather some internal measure, we can accept this measurement as-is, as we are currently relying on an experimentally determined threshold to decide between the inside- and outside-range cases.

Table 1: Sudoku solution (start values are written in bold).

7	**9**	**4**	5	8	2	**1**	**3**	6
2	6	8	9	3	1	7	4	5
3	1	5	4	**7**	**6**	9	8	**2**
6	**8**	9	7	**1**	5	3	2	4
4	**3**	**2**	8	6	9	5	7	1
1	5	7	**2**	4	3	8	**6**	9
8	2	1	6	**5**	7	**4**	9	3
9	4	3	1	2	**8**	6	5	**7**
5	7	**6**	**3**	9	**4**	2	1	**8**

which is able to capture the outline of a mobile device on the surface should be able to implement this method. The contact sensor does not have to be vision-based, the same process would also work for sensing surfaces which detect pressure or electrical field strength (e.g. SmartSkin [19]) as long as they do not interfere with Bluetooth reception.

4 Interacting with a Sudoku Board Game

Assuming that the location of any mobile device on the table surface is known, we shall now focus on the interaction with the Sudoku game through each of the two interfaces, mobile and tabletop.

4.1 Sudoku

As mentioned before, Sudoku is a highly popular puzzle game which can be played alone or as a team. The reason why we chose this particular game was that from an abstract point of view, Sudoku closely models a search problem with multiple constraints. Therefore, insights gained through analysis of such a game may offer the potential to be applied to other, real-world problems.

There are many variants of Sudoku. For this interface, we settled on the best-known one, which consists of a 9*9 grid made from nine 3*3 sub-grids. The puzzle has to be filled with numbers from 1 to 9 so that each row, column and sub-grid contains every number exactly once, as shown in table 1. On the basis of given start values the Sudoku puzzle typically is uniquely solvable.

Playing Sudoku requires the player to perform two basic tasks: keeping track of unoccupied fields, and keeping track of a subset of the numbers which are already on the board. Usually, players focus on one number at a time and therefore need to know all other locations which are occupied by that number. When playing in a team, the players need to communicate to take advantage of their combined efforts. Numbers placed by one person may help another person by generating additional constraints which make it easier to decide on specific locations.

4.2 System Architecture

For playing the Sudoku puzzle collaboratively on mobile hand-helds and the stationary table top device we designed a simple system architecture. The state of the Sudoku game can be described by a string of 81 characters (assuming that a standard Sudoku puzzle with a 9*9 grid is played). Starting in the upper left corner of the grid, all fields of the grid are listed row-by-row. In summary each field can take on one of 19 different states, besides the *empty* state (represented by *0*) it can contain a *user state* from 1 to 9 (represented by *1-9*) or a *start state* from 1 to 9 (represented by *A-I*). The current system architecture bases on a client-server model. The tabletop system provides the server to which the mobile hand-helds are connected via the wireless network. The current communication protocol is restricted to the commands which are necessary for the collaborative solving of a Sudoku puzzle:

- **State?**
  Client request for sending the current state of the Sudoku puzzle
- **State! <valueString>**
  Client request for setting the current state of the Sudoku puzzle to the state which is described by the *valueString*
- **State <valueString>**
  Server response on both state requests with the state contained in the *valueString*
- **Action?** $<x>$ $<y>$ **<value>**
  Client request for setting the field in column x and row y to *value*
- **Action** $<x>$ $<y>$ **<value>**
  Server response on an action request containing the *value* for the field in column x and row y

The requests for changing the server state typically succeed, provided that the *valueString* is syntactically and semantically correct. The string has to contain 81 characters from 0-9 or A-I to be syntactically correct. In order to succeed the test on semantical correctness, the *start states* in the Sudoku puzzle must be arranged in such a way that the Sudoku is solvable. For instance, each of the characters A-I may occur only once in each row, column and sub-grid. The fields filled with *user states*, however, are not tested during the semantical test because the Sudoku remains solvable even if the user states are semantically inconsistent (assuming that the user interface contains the functionality to revert changes). On the one hand clients can join a running game by sending the *State?* request and on the other hand the clients can share their game with other clients by sending the *State!* request.

The requests for performing actions are slightly more complicated. An action which a client wants to perform can fail for two reasons: The client tries to overwrite a field filled with a *start state* with a *user state* or another client tries to change the field at the same time. When one of these conflicts occurs, the server sends the current field state (which differs from the state requested by the client) in his *action* response to inform the client that his action failed. This concept is generally completely resistant against state inconsistencies due to the fact that a central server decides whose action succeeds and whose fails.

4.3 User Interface

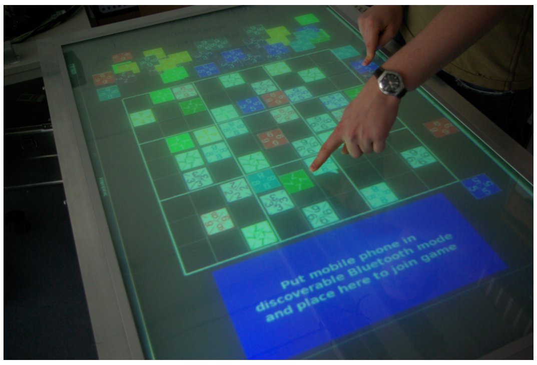
As stated above, two different user interfaces are necessary for the two interaction modalities. The table top system has to support multiple concurrent users, while the mobile UI should be easily usable with a stylus.

4.3.1 Tabletop Device The table top user interface is presented in Figure 3. It was built with libTISCH [5] (`http://tisch.sf.net/`), a library for tangible and multitouch interaction which was developed by the authors.

On one side of the display, a "join area" is located in which users can place their mobile device in order to join the game. Instructions for activating the Bluetooth transceiver and placing the device are displayed inside the join area. Once a device has been located as described above, download of the client software to the device is initiated. Users can now pick up their device in order to authorize the file transfer. After the download has finished, most mobile devices will automatically prompt the user whether the received software should be executed. After the client has been started, it will immediately connect to the game server and join the running game.

The remaining part of the table top display is used for the Sudoku game interface itself. It was inspired by the JigSawDoku browser game [15]. To the side of the central grid, users are presented with a selection of colored number tiles. Fixed numbers which are already present in the grid are shown with a white background. Users can drag and drop the colored tiles into the free fields of the Sudoku grid by simply touching and moving them with their fingers. As the table top system provides multitouch input, several users can concurrently move and place tiles. As the users can view the table from any side, the tiles show each number in four different orientations. To ease correct placement, the tiles snap into the free fields below a certain distance. During the game, users can quickly determine the approximate number of fields left for a certain number by looking at the tile colors. When the grid has been filled correctly, a message is displayed that the game has finished. The time which users took to complete the puzzle is displayed on top of the screen as well as logged to a file for later evaluation.

All tiles which are placed in the table top interface are wirelessly transmitted to the mobile devices and also displayed there. Vice versa, when a number is set on the hand-held, one of the unused tiles on the table top is moved to the correct cell with a short animation.


Put mobile phone in discoverable Bluetooth mode and place here to join game
(a) Playing Sudoku on the tabletop interface.

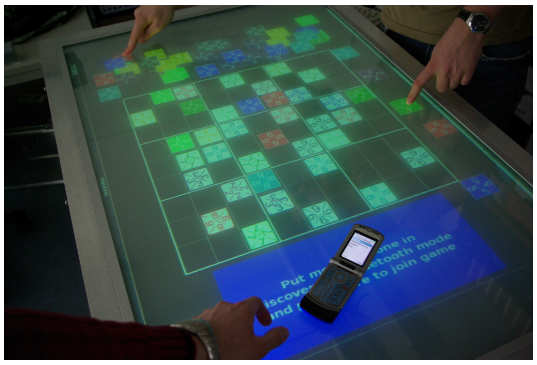

(b) Joining the game with a mobile phone.

Fig. 3: Sudoku user interface.

4.3.2 Mobile Devices The user interface for the mobile hand-held devices is shown in Figure 4. As screen space is highly limited when developing for mobile hand-helds, the visualization differs from the one for the table top device. To avoid cluttering the interface, we have decided against displaying all unset tiles separately. Otherwise, space would be too limited to show the complete Sudoku grid at once. Thus the user interface would then have to contain intuitive metaphors to scroll, pan and zoom. Therefore, we display all unset tiles on 9 different stacks and indicate the height of these stacks numerically.

The metaphor for moving tiles differs slightly from the one for the table top. After some short experiments, we concluded that the movement of tiles by the "drag-and-drop" metaphor is very inaccurate for hand-held devices. However, separating the tile movement into the two steps *tile selection* and *tile placement* works quite

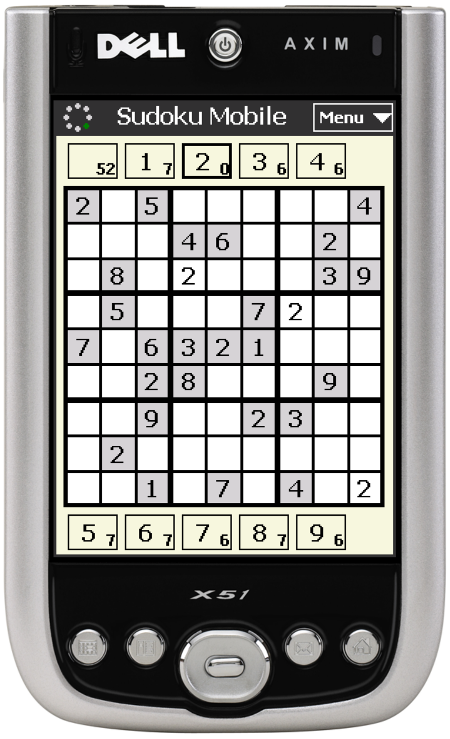


Fig. 4: Sudoku puzzle on mobile devices

well when performing both of these sub-actions with a separate click. First, the user clicks on the tile which he wants to place and second, he clicks on the field which he wants to fill with that tile. Furthermore, when the user wants to place several tiles from the same stack, the first click is only required once because the tile stacks remain selected. Additionally a tenth stack was included, the "empty stack" which can be used to clear *user state* fields. The metaphor for clearing fields works in analogy to the one for filling fields: first the empty stack and then the field which has to be cleared is selected. The height of the "empty stack" indicates the number of tiles which have still to be set in the current game.

4.4 Evaluation

To test our casual connection method in a real-world scenario, we first evaluated the pairing process between mobile devices and the tabletop system through an expert review with four participants. The reviewers were given the task to join the running Sudoku game with their mobile phone by following the instructions displayed on the tabletop system.

All four persons agreed that the main drawback was the still noticeable delay between putting down the device and starting the download. While the optical tracker is able to recognize the device almost immediately, the list of visible Bluetooth devices is often cached internally by the transceiver. This caching process may add a significant delay until the download is initiated. The duration after which cached entries expire is usually not adjustable externally and depends on the transceiver being used. Testing several transceivers with respect to their scan cycle duration and cache expiry time is therefore advisable.

A valuable suggestion by one of the reviewers was to display visual feedback as soon as the optical tracking detects a potential mobile device. For example, a spinning circle could appear below the device, thereby telling the user that the joining process has been started. We will implement this feature in a future revision of our software.

In addition to the expert review, we performed a small-scale evaluation to determine the advantages of coupling mobile hand-helds with table top devices. The better the two user-interfaces support collaborative problem solving, the less face-to-face discussions are essential for successful problem solving. We therefore compared the effectiveness of face-to-face collaboration at the table top device and a mixed mode of collaboration where some participants used hand-held devices. The subjective impressions of the participants were identified by a questionnaire.

In summary 16 people (ages 16 to 49, 6 female, 10 male) participated in our small-scale user study. Their objective was to solve three different Sudoku puzzles collaboratively in teams of four. We evaluated two different alternatives of collaboration:

– **Tabletop.** All four people are collaborating at the table top.
– **Mixed.** Two people are collaborating at the table top, two people are equipped with hand-helds. All participants are in the same room. Roles are swapped after the first game.

As we wanted every participant to evaluate the *mixed* collaboration on the hand-held as well as on the table top, we needed two trials for this alternative. In total, three alternatives were evaluated by each of our four teams. We randomized the order of the alternatives to avoid training effects and to compensate potential differences in the difficulty of the three Sudoku puzzles. However, the puzzles were automatically generated with identical difficulty levels beforehand.

The quantitative results of the user-study are shown in Figure 5. When using only the table top device, the users solved the Sudoku puzzle within 473 seconds in average (std. dev. 194 s), whereas the mixed-mode collaboration needed 585 seconds (std. dev. 506 s). While a first glance suggests that the collaboration with handhelds is slowing down the overall progress, this difference is not statistically significant according to a t-test ($df = 10, p = 0.686$). Therefore, the results do not support the hypothesis that collaborative solving of a Sudoku game on a single device is more efficient than on

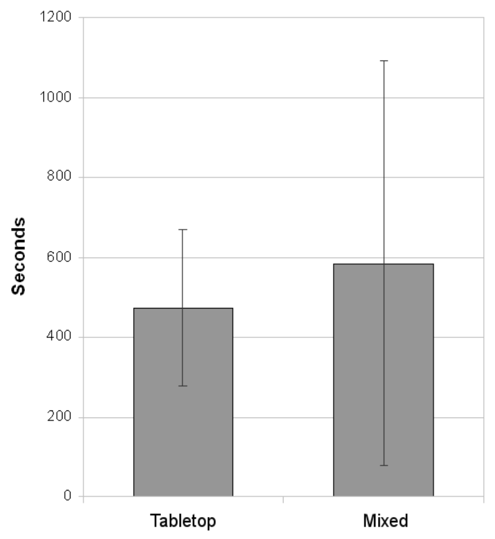


Fig. 5: Quantitative evaluation results with standard deviation.

multiple devices. It remains to be seen whether this is an effect of the Sudoku game itself. One problem in this case is that to gather enough samples, a very large number of participants is needed, as four persons need to work together to generate one single sample.

In addition to the quantitative evaluation, the subjective impression of the 16 participants was documented by a simple questionnaire which consisted of six questions:

– Which interface you did enjoy more? (1..table top – 5..hand-held): **2,4** (std. dev. 1,4)
– Which interface was more efficient? (1..table top – 5..hand-held): **2,6** (std. dev. 1,3)
– Have you been disturbed by the actions of other players when you played on the hand-held? (1..very often – 5..never): **3,1** (std. dev. 1,2)
– Have you been disturbed by the actions of other players when you played at the table top? (1..very often – 5..never): **1,9** (std. dev. 0,7)
– How present were the other players when you played on the hand-held? (1..very present – 5..not present): **2,5** (std. dev. 1,0)
– How present were the other players when you played at the table top? (1..very present – 5..not present): **2,1** (std. dev. 0,9)

Regarding the interface, the participants could not clearly decide between the table top and the hand-held device, neither in terms of enjoyment nor with respect to efficiency. Both results do not significantly differ from a normal distribution; one might conclude that both choices are entirely due to personal preferences. The same applies to the question how players felt the other participants' presence when working at different devices. No significant difference can be seen here, either (paired t-test: $df = 15, p = 0.652$).

However, one interesting and statistically significant result according to a paired t-test ($df = 15, p = 0.006$) can be gained from the question whether users felt disturbed by the other players' actions. This was more often the case for tabletop users. An explanation for this result might be the animation which is displayed in the tabletop interface when a hand-held user sets a number. As one of the free tiles moves itself into the target field, this unexpected movement may cause distraction.

5 Conclusion and Future Work

In this paper, we have presented an approach for casual collaboration between players of a Sudoku game. Users have a choice of playing either on an interactive tabletop system or on their own mobile device, which they can join to the game by placing it on a designated area on the tabletop interface.

To support this type of interaction, we presented a method to track and identify unmodified mobile phones on an interactive surface. We are currently investigating possible improvements to our tracking approach, such as using several Bluetooth adapters in parallel. Although RSSI data is very noisy, a least-squares optimizer could be employed to provide at least a rough position estimate as opposed to the current binary within/outside range decision. This would allow the tracker to correctly distinguish between multiple phones and non-Bluetooth objects which are simultaneously placed on the surface. Another important aspect which should not be ignored is that active Bluetooth devices always pose a security risk. It might be advisable to remind the user to turn off the transceiver after use.

The evaluation showed that mobile hand-helds enable users to remotely collaborate with users playing on the tabletop. While no significant insights regarding the efficiency of the user interfaces could be gained yet, the questionnaire yielded better results. One significant set of answers underlined the importance of keeping the user interface free from visual clutter such as unnecessary animations.

Whereas a table top offers possibilities for direct collaboration, the physical presence of all participants can not be guaranteed in all cases. Therefore, the extension of existing table top applications with mobile user interfaces offers additional opportunities for interaction. One aspect of future work will be to find out how the different modalities of collaboration work in detail. For instance, it should be investigated whether the contribution of every single player depends on the used input device.

References

1. Bluetooth SIG. Core specification 2.1 + EDR. `http://www.bluetooth.com/Bluetooth/Technology/Building/Specifications/`, 2007 (accessed 2009-07-06).
2. S. Carter, E. Churchill, L. Denoue, J. Helfman, and L. Nelson. Digital graffiti: public annotation of multimedia content. In *CHI '04 extended abstracts on Human factors in computing systems*, pages 1207–1210, 2004.
3. J.G. Castano, M. Svensson, and M. Ekstrom. Local positioning for wireless sensors based on bluetooth. *Radio and Wireless Conference, 2004 IEEE*, pages 195–198, 19-22 Sept. 2004.
4. F. Echtler, M. Huber, and G. Klinker. Shadow tracking on multi-touch tables. In *AVI '08: Proceedings of the working conference on Advanced Visual Interfaces*, pages 388–391, 2008.
5. F. Echtler and G. Klinker. A multitouch software architecture. In *Proceedings of NordiCHI 2008*, pages 463–466, October 2008.
6. Florian Echtler and Gudrun Klinker. Tracking mobile phones on interactive tabletops. In *MEIS '08: Proceedings of the Workshop on Mobile and Embedded Interactive Systems*, 2008.
7. J. Favela, M. Rodriguez, A. Preciado, and V.M. Gonzalez. Integrating context-aware public displays into a mobile hospital information system. *Information Technology in Biomedicine, IEEE Transactions on*, 8:279–286, 2004.
8. S. Greenberg and M. Boyle. Moving between personal devices and public displays. In *Workshop on Handheld CSCW*, November 1998.
9. S. Greenberg, M. Boyle, and J. Laberge. PDAs and shared public displays: Making personal information public, and public information personal. *Personal and Ubiquitous Computing*, 3:54–64, March 1999.
10. S. Greenberg and C. Fitchett. Phidgets: easy development of physical interfaces through physical widgets. In *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 209–218, 2001.
11. J. Hallberg, M. Nilsson, and K. Synnes. Positioning with bluetooth. *Telecommunications, 2003. ICT 2003. 10th International Conference on*, 2:954–958, 23 Feb.-1 March 2003.
12. Jefferson Y. Han. Low-cost multi-touch sensing through frustrated total internal reflection. In *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 115–118, 2005.
13. E. M. Huang and E. D. Mynatt. Semi-public displays for small, co-located groups. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 49–56, 2003.
14. H. Ishii and B. Ullmer. Tangible bits: Towards seamless interfaces between people, bits and atoms. In *CHI '97: Proceedings of the Conference on Human Factors in Computing Systems*, pages 234–241, 1997.
15. R. Lee and G. Greenspan. Jigsawdoku. `http://www.jigsawdoku.com`, 2008.
16. S.K. Lee, W. Buxton, and K.C. Smith. A multi-touch three dimensional touch-sensitive tablet. In *CHI '85: Proceedings of the ACM Human Factors in Computing Systems Conference*, pages 21–25, San Francisco, California, USA, 1985. ACM Press.
17. C. Magerkurth, R. Stenzel, and T. Prante. Stars - a ubiquitous computing platform for computer augmented tabletop games. In *Adjunct Proceedings of the Fifth International Conference on Ubiquitous Computing (UBICOMP'03)*, pages 267–268. Springer, 2003.
18. Microsoft Corporation. Surface. `http://www.microsoft.com/surface/`, 2008 (accessed 2010-01-04).
19. J. Rekimoto. SmartSkin: an infrastructure for freehand manipulation on interactive surfaces. In *CHI '02: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 113–120, 2002.
20. J. Schöning, M. Rohs, and A. Krüger. Using mobile phones to spontaneously authenticate and interact with multi-touch surfaces. In *PPD'08. Workshop on designing multi-touch interaction techniques for coupled public and private displays*, May 2008.
21. A.J. Solon, M.J. Callaghan, J. Harkin, and T.M. McGinnity. Case Study on the Bluetooth Vulnerabilities in Mobile Devices. *IJCSNS*, 6(4):125, 2006.
22. A.D. Wilson. PlayAnywhere: a compact interactive tabletop projection-vision system. In *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 83–92, 2005.
23. A.D. Wilson and R. Sarin. Bluetable: connecting wireless mobile devices on interactive surfaces using vision-based handshaking. In *GI '07: Proceedings of Graphics Interface 2007*, pages 119–125, New York, NY, USA, 2007. ACM.