Device Composition on Tabletop Computers

Master Thesis by Leo Sicard
lnsi@itu.dk

March 2012

Abstract

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum. Typi non habent claritatem insitam; est usus legentis in iis qui facit eorum claritatem. Investigationes demonstraverunt lectores legere me lius quod ii legunt saepius. Claritas est etiam processus dynamicus, qui sequitur mutationem consuetudium lectorum. Mirum est notare quam littera gothica, quam nunc putamus parum claram, anteposuerit litterarum formas humanitatis per seacula quarta decima et quinta decima. Eodem modo typi, qui nunc nobis videntur parum clari, fiant sollemnes in futurum.

Acknowledgements

Juan David Hincapie Ramos, Aurlien Tabard, Jakob Bardram, Sebastian Bttrich, PitLab, experiment volunteers. Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum. Typi non habent claritatem insitam; est usus legentis in iis qui facit eorum claritatem. Investigationes demonstraverunt lectores legere me lius quod ii legunt saepius. Claritas est etiam processus dynamicus, qui sequitur mutationem consuetudium lectorum. Mirum est notare quam littera gothica, quam nunc putamus parum claram, anteposuerit litterarum formas humanitatis per seacula quarta decima et quinta decima. Eodem modo typi, qui nunc nobis videntur parum clari, fiant sollemnes in futurum.

Contents

List of Figures

Chapter 1

Introduction

Tabletops should be used as peripherals

Improve user experience

both mobile devices and tabletops are made useful.

Tabletop displays offer a more direct, intuitive experience than traditional computers: - the interactive area is more natural and regroups both input and output - by using touch / hand gestures instead of input peripherals (keyboard, mouse) - collocated and collaborative atmosphere They have potential for being adopted by a broader public, not necessarily computer literate. By focusing on an intuitive interaction, tabletops can be made accessible to a broader range of users.

1.1 Problem formulation

Tabletop computers are cutting-edge devices that merge input and output spaces into one single interactive surface [10]. Researchers have investigated the use of interactive tables in a number of different ways: support for meetings, canvas for architectural design [3], media for document navigation, mediator for sharing files, etc. Due to their size and embedded nature, tabletops seem to naturally fit in public spaces such as shops, bars and work places. Common scenarios include catalog browsing, drink ordering and product configuration. Technologies such as DiamondTouch [4] allow tabletops to support multiple and simultaneous users. Example of applications include sharing data between smartphones, collaborating on a design [7], or simply taking notes during a meeting. In the case of multiple individualized users, solutions are needed to identify each user, as seen in [9], where the simple action of placing one's hand on the surface enables a person to identify and start interacting with the device.

Another interesting property of interactive surfaces is their ability to integrate with physical objects, both passive and dynamic, for the purpose of augmenting them with digital information, or controlling the application state. For example, SurfaceWare [5] allows the Microsoft Surface to sense the fluid level in a slightly enhanced drinking glass. Another example is the software developed by Amnesia Razorfish [1], that allows the sharing of data between multiple handheld devices using the actual devices, as well as gestures, on the Microsoft Surface. Finally, researchers at ITU have developed the Rabbit [6], a device that integrates small RFID-tagged objects and tabletops.

The specificity of tabletops raises the question of how to interact with them on an everyday basis. Recent development initiatives tend to answer this question by regarding tabletops as yet another computational platform, requiring its own software. With this project, we explore a different approach to integrating tabletops in our environment, namely by using them only as UI peripheral, providing touch-based input and graphical output to the devices that we already have. Exploring this path is supported by three important factors. First, most users already own computing devices, such as laptops or smart phones, with tailor-made applications and local storage, and might be less prone to use an additional device if it requires management (updates, backups, synchronizations, etc) and the purchase of applications. Second, tabletops are embedded in the environment and as such can be expected to be shared devices. Using them as simple graphic peripheral would allow to avoid the traditional desktop/laptop issues related to user profiles, privacy and data integrity. Finally, as embedded devices, it is reasonable to expect tabletops to have good networking capabilities.

Device composition focuses on getting the most out of various computing entities, by making them work together and function as one, as seen in [2]. This project explores device composition for UI integration between tabletops and mobile devices, focusing on seamless user experience and implicit human computer interaction as defined by Schmidt in [8].

UI integration can happen in several different ways:

- *UI transfer* (mirror): the tabletop 'takes over' and displays the UI of the connected device.
- *Dual view*: the tabletop display becomes secondary screen space for the connected device.
- *UI nesting*: the connected device is physically located on the tabletop, and its UI is extended to the additional screen space around it.

Following is an open list of problems that we will address in order to achieve device composition by means of implicit interaction.

1. *Setup*: How is a device enabled for integrating with a tabletop? The setup should be simple, to be performed only once by non-technical users. An ini-

tial survey of possible solutions points towards the use of tagging mechanisms and/or camera-based object recognition.

2. *Discovery*: How do the tabletop and the device discover and communicate with each other? How do we solve the issues of discovery, handshake, network connectivity, and encryption mechanisms to ensure privacy?

3. *UI transfer*: Given the computational constraints of mobile devices, how can the UI transfer be efficiently implemented so as to support native applications and guarantee a seamless user experience?

4. *Input*: How can the users interact with their applications on the tabletop (touch and other peripherals)?

5. *Interaction Design*: What means of interaction are best-fitted for the tabletop-based systems that we propose to develop? How can we best adapt to public/private uses and single/multiple users? How can we take advantage of the larger interaction surface?

Chapter 2

Related Work

Smart Room systems
- Systems that integrate tangibles
- Systems that focus on device composition
- Systems that focus on user identification/authentication
- Interaction Design papers

Chapter 3

Analysis

The thesis investigated here is that using tabletops as basic Input/Output peripherals for other computing devices will help their adoption by the masses. A user should be able to transfer the display of his/her device to a tabletop, and interact with it without any further complications. To demonstrate the feasibility of this interaction model, an analysis of the requirements is conducted, leading to the design and implementation of a prototype application.

The application UI consists of two elements.

- The *virtual screen* provides the user with a clone of his/her device's display, and forwards touch input to the same device. As such, it is device specific and requires no interface design.
- The *application interface* provides the user with ways to manipulate the virtual screen. It needs to be designed and implemented on the tabletop.

3.1 Context of use

3.1.1 Scenarios

The coffee shop

> Alice is in a Coffee Shop, waiting for her friend Bob. She is sitting at an interactive table, which allows her to order her drink via a digital interface, as well as supports TableShare technology.
>
> She takes her phone and notebook out of her purse and places them on the table. A dialog pops up on her smartphone, asking her to confirm the UI transfer, which Alice does by a simple touch. A simple menu appears on the

table beside her phone. Alice taps the Expand tab, and her phones screen appears on the table beside her phone, as a same size mirrored display.

She resizes the window to her convenience, and moves it closer to her by sliding her phone on the surface. The screen goes gray to notify Alice that an object (the notebook) is in the way. Alice removes the notebook, and the window becomes active again. She accesses her phones applications, and starts typing an email.

When Bob arrives, Alice minimizes her display, keeping her phone in place. Bob orders a drink and they start catching up. After a while, Bob leaves. Alice restores her phones screen, finishes up her email, and disconnects TableShare by simply lifting the phone off the table.

The meeting

Jim, Jack and Jill are having a meeting about the development of a product. They are sitting around an interactive table, with different artefacts, including paper, pens, computing devices and coffee cups. Jill is responsible for the meetings agenda, which is stored on her smartphone.

Jill places her phone on the top right corner of the table and establishes a UI transfer via TableShare. She uses the Grab tap to drag the display window to the left of the phone where there is space. She opens the document containing the agenda, and uses the Resize corner of the window to enlarge the display, so as to allow convenient visual reference for all present.

It is now time for Jack to present a diagram of the development process. He switches on his tablet computer, opens said diagram, and places the tablet on the table for the others to see. The screen is however too small, so Jack decides instead to use the TableShare application. With a single tap on the Grab tab, Jack pins the UI to the table, allowing him to remove the physical tablet while keeping the mirrored display active. By using the Resize corner, he enlarges and flips the orientation of the window to a landscape display. By the use of a double touch on the Grab tab and the active window, Jack rotates the display to a convenient position, and presents the diagram to his colleagues. When done, Jack minimize the window by tapping the Retract tab. The window takes the shape of an active icon, ready to be restored or closed as convenient. When the meeting is over, Jack taps the Close tab on the icon to interrupt TableShare.

The office

It is monday morning and Bill arrives at his office. His working desk is made up of an interactive table, extended with a vertical screen, mouse and keyboard. On it are various physical objects, including a stack of papers, some books, pens, an empty cup and a lamp.

Bill wakes the tabletop up from its standby state by simply placing his smartphone on it. The devices know each other, so TableShare is automatically launched. Bill uses the TablePush widget on his phone to push other widgets to the table space. Bill places his calendar up in one corner, together with his Skype widget.

After reading through his mail on the vertical screen, Bill starts typing an answer using the keyboard. He needs to refer to a document that is stored on his phone. Bill and uses the Grab tab beside his phone to attach its display beside the device. He enlarges the window and moves the display to a convenient location by sliding the phone. Bill types on..

Suddenly the phone rings. Bill taps the Grab tab to effectively pin all applications and UI display to the tabletop, allowing him to pick up the phone without interrupting the UI transfer.

3.1.2 Storyboards

Interest of storyboards

3.2 Application requirements

3.2.1 Hardware constraints

Design concerns for tabletops

A tabletop computer is ..

.. a situated device:
not mobile
public
interruptions in interaction flow
.. a table:
horizontal working surface
concurrent activities on table: other apps, other devices (papers, cup..)
limited (screen) space availability
reach issue
.. an interactive surface:
use of Tangible UIs possible
input techniques: fingers (gestures), pen, mouse, other devices
use of screen space: drag/resize/rotate, touch areas, full screen
.. a shared device:
user identification/authentication (virtual lenses, NAI privacy/security
multi-user scenarios: collaborative/parallel

.. a computer:
specs, platform, hardware ??
network capabilities

Designing for mobile devices

3.2.2 Software components

Detection and discovery

Display transfer

Surface user interface

3.2.3 Choice of features

3.3 Interaction design

3.3.1 Generating ideas with low fidelity prototypes

3.3.2 Commands and actions

Human computer interaction can be modeled as a simple cause-effect relationship.

The user wishes the computer to execute a command. To achieve that, he/she performs an action, making use of the available input devices (keyboard, mouse, etc..) and interfaces. The action is the cause, the command is the effect, and together they form a single interaction between user and machine.

On an interactive surface, the traditional input devices are gone, and the interaction is based on hand gestures.

Those concepts are inspired by the paper "User-Defined Gestures for Surface Computing" by Wobbrock, [11].

The following six basic commands (interaction primitives) were identified and progressively chosen as being essential to the TableIO application.

1. *Dragging* the application window across the interactive surface.

2. *Rotating* the application window across the interactive surface.

3. *Resizing* the application window across the interactive surface.

4. *Minimizing* the application window, making it possible to restore it easily.

5. *Hiding* the content of the application window.

6. *Exiting* the application, thus closing the application window.

EXTRA command: device buttons (HOME on the iPhone) should be supported.

3.3.3 Interaction strategies

Various interaction techniques can be used to invoke application level commands.

Early idea generation process lead to the definition of specific interaction strategies.

Each strategy can be consistently implemented for each previously defined command.

1. *Action Tabs* are traditional buttons/tabs that implement functionalities.
2. The *Action Bar* can be compared to a virtual touchpad, it includes a manipulation area and buttons.
3. *Window Toggle* refers to using a switch to toggle the window between inactive and active states. In its inactive state, the window is made manipulable as a common digital picture.
4. The *Active Border* is a digital frame around the application window used for manipulation.
5. *Active Corners* is a strategy similar to Active Border, with the difference that the border's corners implement specific functionalities.
6. *Other* regroups suggestions that do not fit with any specific strategy.

3.4 Preliminary usability study

A well-designed product is a successful one. Usability and appeal are key elements towards the success of an application. The goal of this experiment is to gather knowledge directly from users to inform important design decisions.

Example of designers designs that fails from user standpoint.

The goal is to have potential users of the system describe their ideal user interface.

The focus of the experiment is not the interaction with the mirrored smartphone's screen. The focus is the manipulation of the application window that contains the mirrored display.

3.4.1 Method

Parameters

The parameters of the experiment are the above mentioned commands and actions. The 6 commands form a set of features that are considered necessary for the application to function.

For each of the chosen commands, we have 6 implementation suggestions (1 for each interaction strategy).

Participants

Experiment

The experiment is based on low fidelity prototypes. Instead of a digital user interface, the participant interacts with paper representations.

The user is asked to perform a task using the application. The task is divided into the 6 actions that

3.4.2 Results

Comments: users suggest dragging/rotating the window directly (forget that window forwards input to device)

LIMITS: UNFORTUNATELY, results are biased due to splitting the suggestions in 2, for each participant group. Ranking result for 1 strategy is only valid compared to 2 other suggestions from same participant group, not valid across all strategies.

Chapter 4

Design

focus on coherence of interaction

move from traditional explicit buttons towards a more implicit, physical interaction.

Interest of playful, exploratory learning process for the user

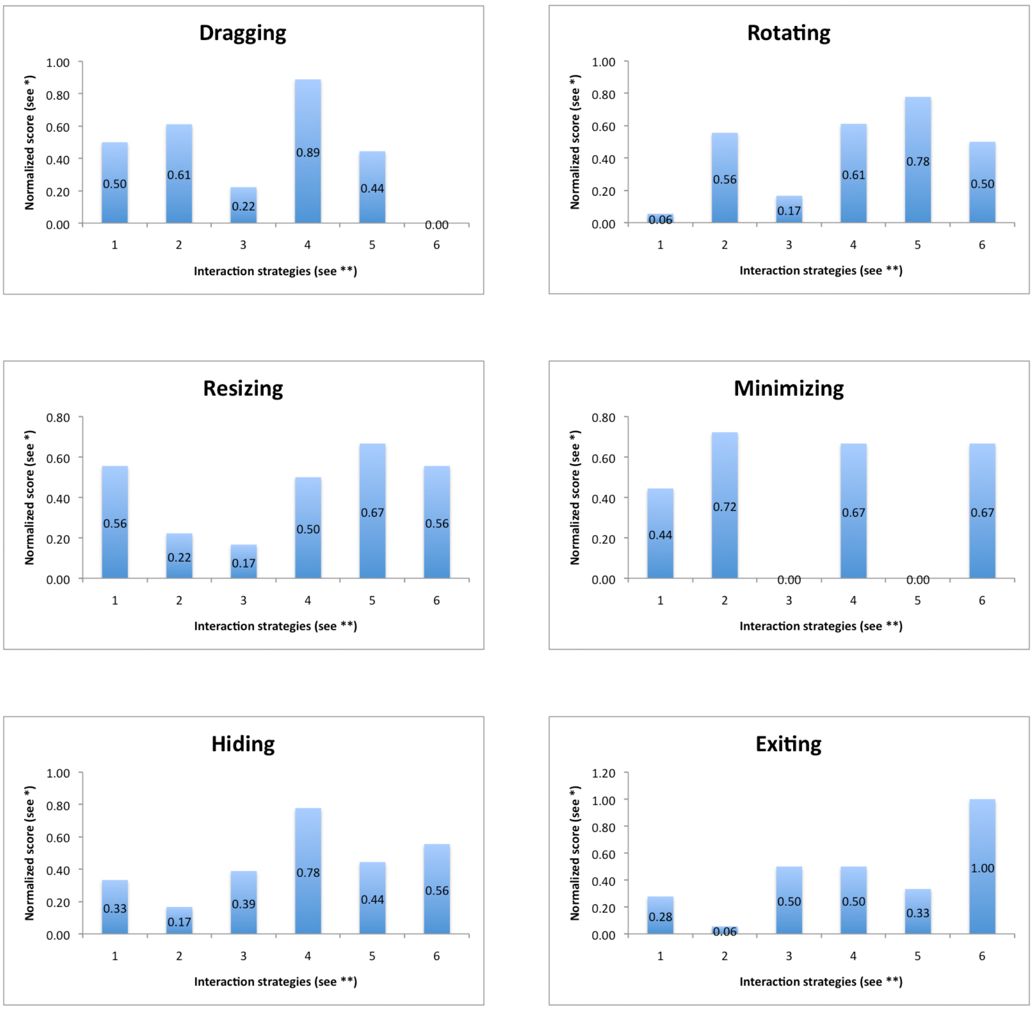
Interest of appealing design

4.1 users best

4.2 designers best

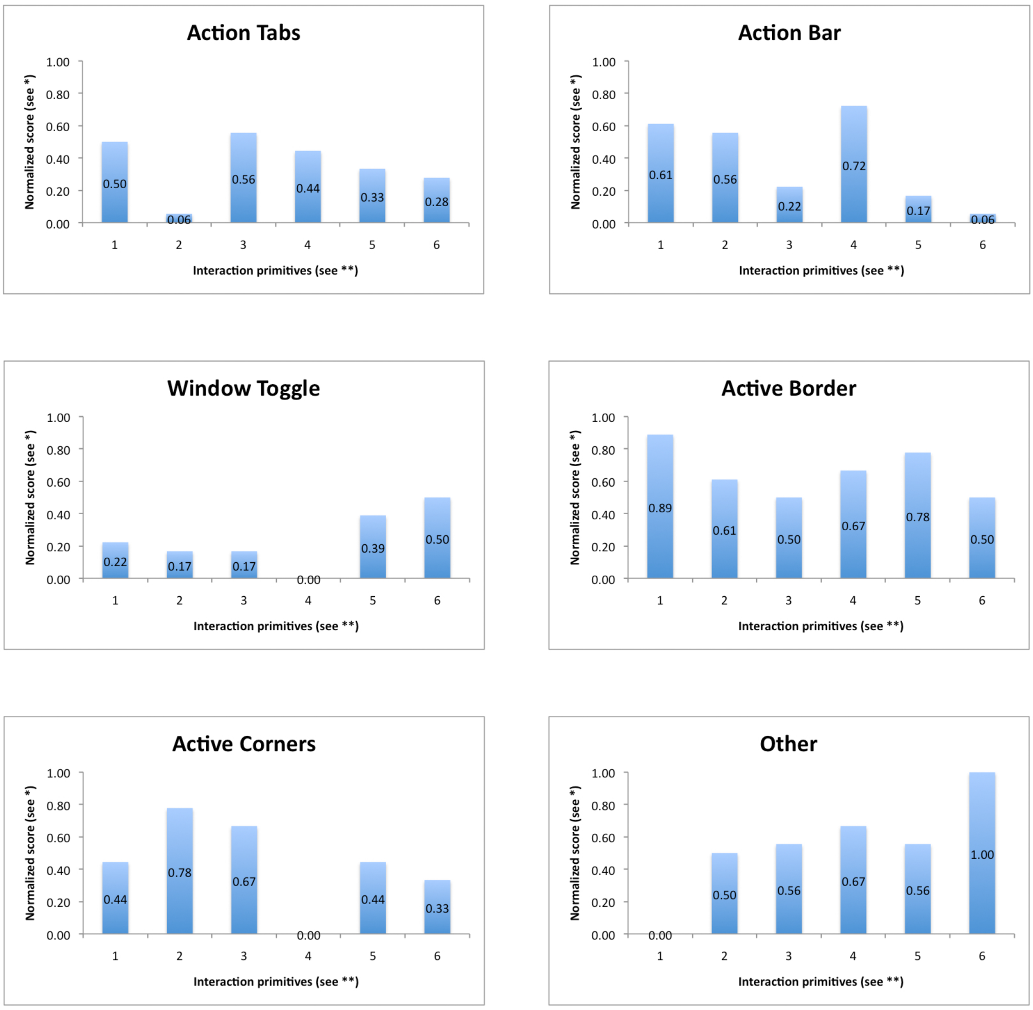
4.3 hybrid

Figure 4.1: Interaction primitives.



*** The normalized score is the normalized weighted average of the votes per strategy/primitive.**
**** The interaction strategies are 1) Action Tabs 2) Action Bar 3) Window Toggle 4) Active Border 5) Active Corners 6) Other**

Figure 4.2: Interaction strategies.



* The normalized score is the normalized weighted average of the votes per strategy/primitive.
** The interaction primitives are 1) Dragging 2) Rotating 3) Resizing 4) Minimizing 5) Hiding 6) Exiting

Chapter 5

Implementation

refer to problem formulation:

Following is an open list of problems that we will address in order to achieve device composition by means of implicit interaction.

1. *Setup*: How is a device enabled for integrating with a tabletop? The setup should be simple, to be performed only once by non-technical users. An initial survey of possible solutions points towards the use of tagging mechanisms and/or camera-based object recognition.

2. *Discovery*: How do the tabletop and the device discover and communicate with each other? How do we solve the issues of discovery, handshake, network connectivity, and encryption mechanisms to ensure privacy?

3. *UI transfer*: Given the computational constraints of mobile devices, how can the UI transfer be efficiently implemented so as to support native applications and guarantee a seamless user experience?

4. *Input*: How can the users interact with their applications on the tabletop (touch and other peripherals)?

5. *Interaction Design*: What means of interaction are best-fitted for the tabletop-based systems that we propose to develop? How can we best adapt to public/private uses and single/multiple users? How can we take advantage of the larger interaction surface?

1) setup
2) discovery

–¿ how it is not new, what are the existing options, what would I recommend in this context. Discussion. How did I solve it and why.

3) vision-based device tracking detection options: iPhone App, Tag, camera based

4) UI transfer (I/O approach) - technology issues (slow Veency)

5) Interaction design

6) generic implementation attempt include android phone

Chapter 6

Evaluation

in some cases, compare different interaction strategies for same command.

Compare same implementation with no visual aids, discrete visual aids, and overkill visual aids.

6.1 Experiment

Compare and discuss

6.2 Results

6.3 Discussion

Implementing same functionality with several parallel interaction techniques is a good choice because it will augment the number of situations in which a user will obtain the desired effect intuitively on his first try, without referring to any manual.

Chapter 7

Conclusion

Bibliography

[1] http://www.amnesiarazorfish.com.au/2011/01/amnesia-razorfish-announces-gesture-sharing-for-smartphones-and-tablets-using-microsoft-surface-press-release.

[2] Jakob E. Bardram, Christina Fuglsang, and Simon C. Pedersen. Compute: a runtime infrastructure for device composition. In *Proceedings of the International Conference on Advanced Visual Interfaces*, AVI '10, pages 111–118, New York, NY, USA, 2010. ACM.

[3] Paul Clifton, Ali Mazalek, Jon Sanford, Claudia Rébola, Seunghyun Lee, and Natasha Powell. Sketchtop: design collaboration on a multi-touch tabletop. In *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction*, TEI '11, pages 333–336, New York, NY, USA, 2011. ACM.

[4] Paul Dietz and Darren Leigh. Diamondtouch: a multi-user touch technology. In *Proceedings of the 14th annual ACM symposium on User interface software and technology*, UIST '01, pages 219–226, New York, NY, USA, 2001. ACM.

[5] Paul H. Dietz and Benjamin D. Eidelson. Surfaceware: dynamic tagging for microsoft surface. In *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction*, TEI '09, pages 249–254, New York, NY, USA, 2009. ACM.

[6] Juan David Hincapié-Ramos, Aurélien Tabard, and Jakob E. Bardram. Mediated tabletop interaction in the biology lab - exploring the design space of the rabbit. *Ubicomp*, 2011.

[7] Seth Hunter, Pattie Maes, Stacey Scott, and Henry Kaufman. Memtable: an integrated system for capture and recall of shared histories in group workspaces. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, CHI '11, pages 3305–3314, New York, NY, USA, 2011. ACM.

[8] Albrecht Schmidt. Implicit human computer interaction through context. *Personal and Ubiquitous Computing*, 4:191–199, 2000.

[9] Dominik Schmidt, Ming Ki Chong, and Hans Gellersen. Handsdown: hand-contour-based user identification for interactive surfaces. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*, NordiCHI '10, pages 432–441, New York, NY, USA, 2010. ACM.

[10] Pierre Wellner. Interacting with paper on the digitaldesk. *Commun. ACM*, 36:87–96, July 1993.

[11] Jacob O. Wobbrock, Meredith Ringel Morris, and Andrew D. Wilson. User-defined gestures for surface computing. In *Proceedings of the 27th international conference on Human factors in computing systems*, CHI '09, pages 1083–1092, New York, NY, USA, 2009. ACM.