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 Wellner [1993]. Researchers have investigated the use of interactive tables in a number of different ways: support for meetings, canvas for architectural design Clifton et al. [2011], 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 Dietz and Leigh [2001] allow tabletops to support multiple and simultaneous users. Example of applications include sharing data between smartphones, collaborating on a design Hunter et al. [2011], or simply taking notes during a meeting. In the case of multiple individualized users, solutions are needed to identify each user, as seen in Schmidt et al. [2010], 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 Dietz and Eidelson [2009] allows the Microsoft Surface to sense the fluid level in a slightly enhanced drinking glass. Another example is the software developed by Amnesia Razorfish amn [2011], 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 Hincapié-Ramos et al. [2011], 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 Bardram et al. [2010]. 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 Schmidt [2000].

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 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?

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

Theory

Chapter 4

Design

WHY we design this

HOW we design this

WHAT this chapter reports

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.

4.1 Enhancing mobile computing with tabletops (application context analysis)

It is critical to understand fully the context in which the system will be used in order to achieve a good design. Most users own a computing device with personal data and applications that are tailored to their needs. Those *personal devices* are becoming smaller and more mobile, with devices such as tablet and handheld computers. In many cases, the display size of the personal device is a limitation in terms of graphical input and output, and has a negative influence on the user experience. One of the main characteristics of tabletops, however, is that they have superior graphical I/O capabilities. This project focuses on situations where a tabletop can be used as a *display extension* to the personal device, thus enhancing the user experience.

Figure 4.1 describes the primary use case. The system should basically be a graphical peripheral unit for other computing devices. Its main functions are to forward user input to the device, and to display graphical output from the device.

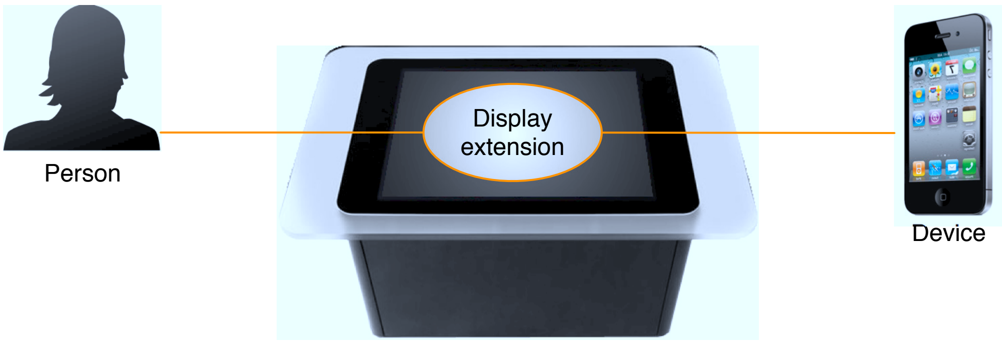


Figure 4.1: Main use case.

4.1.1 The devices

Tabletops

A tabletop presents a range of characteristics that have concrete implications on the system design. A tabletop is ...

... a computer. This project investigates a strategy where a tabletop is used as a smart IO peripheral. The system should allow a tabletop to function as a relay between a personal device and users, forwarding both input and output as required.

... a table. Its working surface is horizontal by nature, with the implication that it cannot support any prolonged interaction, because of the bad ergonomics of the “hunched over” working position. Furthermore, a table gives naturally rise to various activities, such as eating, and it supports all sorts of objects, often in a collocated collaborative atmosphere. The system should therefore support simultaneous users, and handle limited space availability. Finally, a table offers no specific orientation, implying that the orientation of the system UI will have to adapt to the user’s position around the tabletop.

. . . a situated device. A tabletop is not mobile. It usually sits at a specific location, and users come to it in order to use it. The main implication is that tabletops seem to naturally fit in public spaces, where they are shared among multiple users. This tendency is accentuated by the price factor, that makes a private person not likely to buy such a device for private use. The system should handle public use, characterized by short anonymous sessions and an often interrupted interaction flow. Other scenarios should be considered as well, such as a tabletop in an individual office, or in a family home. In any case, it can be expected to have dedicated power supply and network connection, removing such concerns from the developer's mind.

. . . a shared device. In many cases the tabletop will be shared by multiple users, raising concerns of user identification and data protection.

. . . an interactive surface. As such, it typically offers large graphical output, as well as a range of input techniques to allow for user interaction. Most tabletops support multitouch-based input. This introduces a new kind of interaction model, more intuitive, that the system should be based upon. In some cases, such as with camera-based devices, it is possible to add tangible objects to the tabletop experience. This project will report on the possibilities to use a personal device as a tangible control integrated to the tabletop.

Mobile computing devices

A mobile computing device is . . .

. . . a computer. It offers enough computing power, storage space and connectivity to support most users' daily tasks. The system should allow the user to access his/her applications and personal data.

. . . a small device. Hence the mobility. Such a device is carried around by users. However, the small form factor implies a suboptimal graphical user experience. The system should try to improve this, by offering superior IO resources.

. . . a mobile device. With the implications that its power supply is limited by battery life, and its connectivity is unstable. The system should be developed with those concerns in mind.

4.1.2 The situations

There exists different types of everyday situations in which the system can be put to use. They vary depending on whether they involve one or more users, and whether the tabletop is a public or private device. A tabletop is considered a public device as soon as there are more than one user that have access to it. As a consequence, the scenario of multiple users on a private tabletop is not considered. Each situation implies a slightly different set of application features. The original scenarios are included in appendix A.

Single user on a public tabletop

It should be possible for the user to wirelessly pair his/her personal device with a public tabletop computer. This implies that the devices are both connected to the local wireless networks, that they are able to detect each other and discover each other's identity on the network. It would not be safe to establish this connection automatically in a public space. Therefore, dialogs should be used both on the mobile device and on the tabletop to gather user input. The UI of the mobile device should be transferred to the interactive surface as graphical output, and this transferred display should be able to accept touch input to be forwarded back to the device. The transferred display should be contained in an application window, and this window should be manipulable (drag, resize, rotate, minimize, hide, ..).

The application window should react to the state of the interactive surface. An example of this is that the application window should turn inactive if it is obstructed by an object on the table.

Finally, the mobile device could be a participant in the interaction model, i.e. listing it off the table should interrupt the connection and exit the application.

Multi users on a public tabletop

In a collocated collaborative context, it should be possible for more than one personal device to simultaneously have their display extended to the interactive surface. This implies that the implementation should support parallel connections and simultaneous use.

Mobile computing devices come in many forms, and ideally the system should support all of them. Devices vary in terms of software and hardware specifications. Some parameters that are especially important here are the programming platform, as well as the display resolution.

When a tabletop is used simultaneously by multiple users, there is a very concrete risk of lack of space on the surface. This fact introduces a new need for the system, to allow a user to remove his/her personal device from the table, while keeping the display extension active.

Single user on a private tabletop

If the tabletop is private, such as a home computer or a working desk, the system should offer extended functionalities to the user. He/she should have the option to configure the tabletop in order to allow/initiate those extended functionalities. Some suggested functionalities are:

- automatic launch of the display extension application
- push application widgets from the extended display to the tabletop
- share data between the personal device and the tabletop

4.2 Solution requirements

4.2.1 Detection and pairing

system setup on the personal device
discovery and pairing protocol

4.2.2 UI transfer

UI output transfer to the tabletop
UI input transfer to the personal device

4.2.3 Surface UI

manipulation of the UI extension on the tabletop (drag, resize, rotate)
minimize and restore of the UI extension
support the personal device's physical buttons' functions
switch display orientation on the UI extension
handle obstruction of the UI extension
push application widgets to the tabletop

4.2.4 Tangible UI

UI extension attach/detach to the personal device on tabletop
control the UI extension with the personal device (move UI, exit application)

4.3 Interaction design

4.3.1 Storyboards

After identifying system features, storyboards are the next step in the design process. They are a more concrete design approach, and help defining the features in further details. During this phase, the designer starts considering the graphical aspects of the system.

As an example, let us consider the action of resizing the UI extension. This action is in effect one of the main points of interest of the system, as it allows a person to use her device on a potentially much larger screen. Therefore, the resizing feature strikes as a necessary one. Logically, the feature is included in a storyboard. Thus, a UI element must be defined that implements it. At this point, different solutions come up, such as using 2 fingers to pull the window apart, or dragging a corner down to make the window larger.

INCLUDE STORYBOARD EXCERPT

Storyboards help generating design options.

4.3.2 Generating ideas with low fidelity prototypes

4.3.3 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, [Wobbrock et al. 2009].

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.

4.3.4 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.

4.4 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.

4.4.1 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

4.4.2 Results

4.5 Design Choices

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.

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

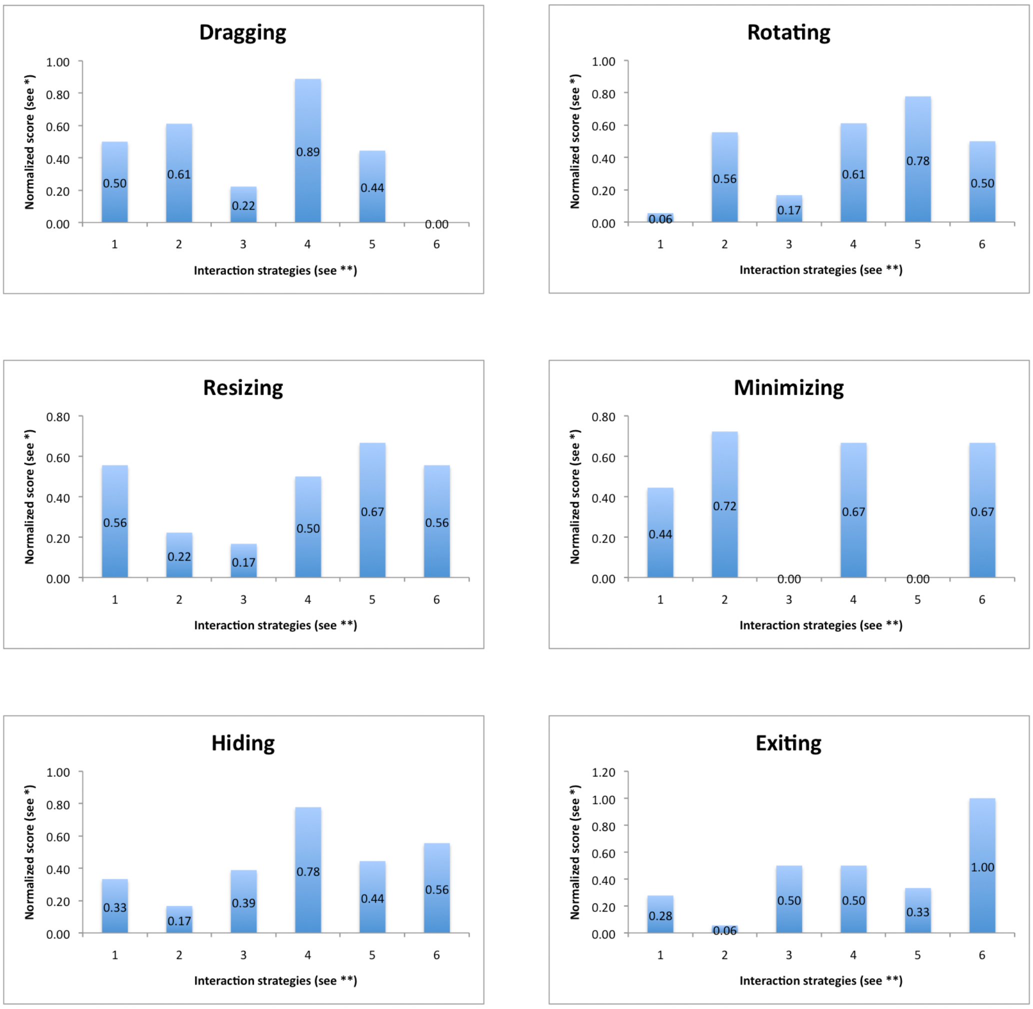
Interest of appealing design

4.5.1 users best

4.5.2 designers best

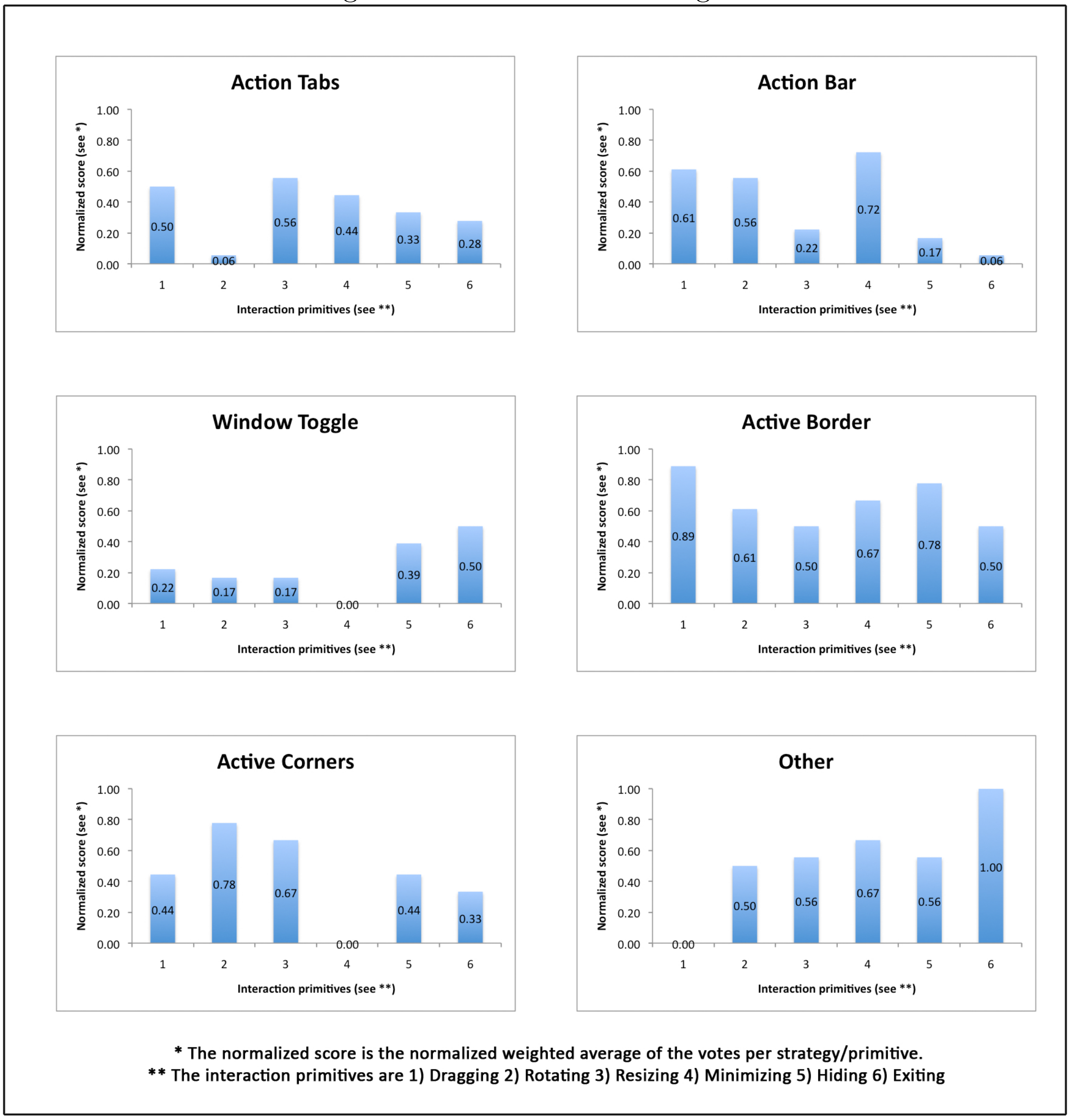
4.5.3 hybrid

Figure 4.2: 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 4.3: Interaction strategies.



Chapter 5

The TIDE application: Tabletop Interactive Display Extension for mobile devices

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?

5.1 Device Composition

5.1.1 Device setup

5.1.2 Detection and 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.

5.1.3 Vision-based device tracking

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

5.1.4 UI transfer

(I/O approach) - technology issues (slow Veency)

5.2 Surface application UI

5.2.1

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

Chapter 7

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 8

Conclusion

Bibliography

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

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. ISBN 978-1-4503-0076-6. doi: http://doi.acm.org/10.1145/1842993.1843014. URL `http://doi.acm.org/10.1145/1842993.1843014`.

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. ISBN 978-1-4503-0478-8. doi: http://doi.acm.org/10.1145/1935701.1935778. URL `http://doi.acm.org/10.1145/1935701.1935778`.

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. ISBN 1-58113-438-X. doi: http://doi.acm.org/10.1145/502348.502389. URL `http://doi.acm.org/10.1145/502348.502389`.

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. ISBN 978-1-60558-493-5. doi: http://doi.acm.org/10.1145/1517664.1517717. URL `http://doi.acm.org/10.1145/1517664.1517717`.

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.

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. ISBN 978-1-4503-0228-9. doi: http://doi.acm.org/10.1145/1978942.1979432. URL http://doi.acm.org/10.1145/1978942.1979432.

Albrecht Schmidt. Implicit human computer interaction through context. *Personal and Ubiquitous Computing*, 4:191–199, 2000. doi: 10.1007/BF01324126.

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. ISBN 978-1-60558-934-3. doi: http://doi.acm.org/10.1145/1868914.1868964. URL http://doi.acm.org/10.1145/1868914.1868964.

Pierre Wellner. Interacting with paper on the digitaldesk. *Commun. ACM*, 36:87–96, July 1993. ISSN 0001-0782. doi: http://doi.acm.org/10.1145/159544.159630. URL http://doi.acm.org/10.1145/159544.159630.

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. ISBN 978-1-60558-246-7. doi: http://doi.acm.org/10.1145/1518701.1518866. URL http://doi.acm.org/10.1145/1518701.1518866.

Appendix A

Scenarios

The coffee shop: single user on a public tabletop

Alice is sitting in a Coffee Shop, waiting for her friend Bob. The table is an interactive surface, which allows her to order her drink via the digital interface.

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 id she wants to establish the connection between the two devices. Alice confirms this by a simple tap. A menu appears on the table beside her phone. Alice taps the 'Connect' button, and her phone's display appears on the table beside her phone.

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 phone's applications, and starts typing an email.

When Bob arrives, Alice minimizes the surface application, keeping her phone in place. Bob orders a drink and they start catching up. After a while, Bob leaves. Alice restores the application, finishes up her email, and exits the application by lifting the phone off the table.

The meeting: multiple users on a public tabletop

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

Jill places her phone on the top right corner of the table and establishes a UI transfer. She uses the 'Grab' button 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 UI transfer application. With a single tap on the 'Grab' button, Jack pins the UI to the table, allowing him to remove the physical tablet while keeping the transferred display active. By using the 'Resize' corner, he enlarges and flips the orientation of the window to a landscape view. 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 minimizes the window by tapping a button. 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 exit the application.

The office: single user on a public tabletop

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 a UI transfer is automatically launched. Bill uses a widget on his phone to push application 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 uses the 'Grab' button beside his phone to attach the 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' button to effectively pin all applications and UI display to the tabletop, allowing him to pick up the phone without interrupting the UI transfer.