TIDE: Lightweight Device Composition for Enhancing Tabletop Environments with Smartphone Applications

Léo Sicard[1], Aurélien Tabard[2,1], Juan David Hincapié-Ramos[3,1], Jakob E. Bardram[1]

[1] IT University of Copenhagen, Denmark, [2] University of Munich (LMU), Germany, [3] University of Manitoba, Canada.

{leo.sicard,aurelientabard,jhincapie}@gmail.com
bardram@itu.dk

Abstract. Interactive surfaces like tabletop computers provide large touch-enabled displays, support novel forms of interaction and collaboration, and extend computation to new environments. However, being a novel platform, the existing application pool is limited and applications existing for other platforms have to be re-developed. At the same time, smartphones are pervasive computers that users carry around and with a large pool of applications. This paper presents TIDE, a *lightweight device composition* middleware to bring existing smartphone applications onto the tabletop. Through TIDE, applications running on the smartphone are displayed on the tabletop computer, and users can interact with them through the tabletop's interactive surface. TIDE contributes to the areas of device composition and tabletops by providing an OS-level middleware that is transparent to the smartphone applications, maintaining privacy by limiting content transfer between devices, and enhancing the usefulness of tabletops with already existing smartphone applications and software developers. We present the design and implementation of TIDE, the study of different interaction techniques to manipulate TIDE's interactive content, and an analysis of different research directions. Initial user feedback shows that TIDE is easy to use, learnable, and convenient for collaborative activities and private environments.

Keywords: Distributed User Interfaces; Multiple Display Environments; Tabletops; Smartphones; Device Composition.

1 Introduction

Tabletop computers have left the laboratories to become commercial products with rich input and output capabilities. Often used as appliances for specific purposes (e.g. exhibits, marketing events or demos), tabletops enable rich walk-up-and-use experiences. However, when building real-world applications for complex tasks [e.g. 6, 8, 29], designers must not only develop the application to support an activity (e.g. biology experiments, design meetings, etc.) but also support the set of basic expectations users would have from interacting with a computing device: the ability to browse the

Web, access remote files, communicate, etc. The lack of applications to support such needs limits the impact and adoption of tabletops in professional environments.

At the same time, smartphones with rich computing capacities are now common, leading to a huge application pool and an important developer base. Particularly relevant to our work, is that many of the general-purpose applications already exist for the smartphone: Web browsing, communication and personal information management, access work related resources, etc. However, inherent to the mobile platforms are restrictions such as limited screen space, occlusion, and problems of touch-based interaction (e.g. the "fat finger" problem).

This paper presents *TIDE*[1] (*Tabletop Interactive Display Extension*): a novel approach to integrate smartphones to tabletops (see fig. 1). TIDE is a middleware running on both devices, allowing the smartphone to connect to the tabletop and replicate its user-interface onto the tabletop. The smartphone screen is displayed on the tabletop, and touch events on the tabletop surface are translated into touches on the phone display. We call this type of integration *lightweight device composition* because it is limited to display graphics and input mechanisms.

The main implication of this approach is that any smartphone application can be used on the tabletop. Moreover, smartphones and tabletops share a similar interaction model mostly based on touch and gestures, meaning that the applications do not have to be adapted to be used “through” the tabletop, or other bridging mechanisms be provided. At the same time, using the smartphone’s applications from the tabletop can mitigate some of the phone’s limitations, like hand obstruction or small form factor. Other implications are that the tabletop’s larger screen size and support for multiple users allow TIDE to better support tasks like reading, drawing or typing, or more social usages of smartphone applications.

Our contributions relate to technical, design, and usability aspects of TIDE. First, we present TIDE and its technical architecture. Smartphone applications run on the phone and are replicated on the tabletop; hence, there is no need for special application programming and any phone application can be replicated to the tabletop. Moreover, personal data never leaves the phone; hence increased security and privacy. Second, we present TIDE’s interaction design which is based on an elicitation study combined with our own design considerations, and present an evaluation of TIDE’s interaction techniques and their discoverability. Finally, we discuss the most convincing use cases based on an evaluation of our prototype, and lay out future research directions for TIDE.

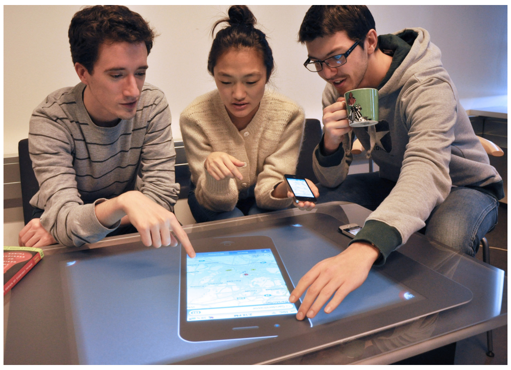

Figure 1: Tabletop users interacting with a map application running on a smartphone through TIDE.

[1] Demo video: http://www.youtube.com/watch?v=SAEARu-WRYk
[2] https://www.apple.com/appletv/airplay/

2 Background

Our motivation for designing TIDE comes from our experience building and deploying tabletops applications for professionals [28]. Such tabletop systems [6, 8] usually rely on a unique custom designed full-screen application. This means that most basic computing tasks are not supported or must be re-developed from scratch. To mitigate this problem the eLabBench [29] supports native Windows applications alongside its main application, while WeSpace [33] supports redirection of windows from laptops to shared displays. However, both the eLabBench and WeSpace only support native Windows applications and their WIMP style of interaction, which is particularly ill-suited to touch interaction.

This work is based on the observation that smartphones now offer a wide variety of touch-friendly applications for carrying everyday computing tasks. We thus developed TIDE to enable mobile applications to run on tabletops. In doing so, we push further existing concepts of smartphones' screen projection onto TVs or dedicated displays (e.g. [31]) through the notion of *lightweight device composition*. In this approach a host device (tabletop) allows a client device (smartphone) to use part of its screen to display content, and channel touch events to the client for processing. Based on our experience developing tabletop applications, lightweight device composition should support the following requirements:

R1. Enable applications to run without any modification.
R2. Support multiple client devices.
R3. Support walk-up-and-use scenarios with minimal set-up.
R4. Enable privacy control by letting users hide the client screen quickly.
R5. Support resizable applications (i.e. not only full-screen).
R6. Support physical separation between host and clients (e.g. putting the phone back in the pocket).

3 Related Work

As a middleware to integrate smartphones to tabletops, TIDE sits at the crossroad of research into the fields of device composition, tabletop augmentation and smartphone projection.

3.1 Device Composition

Inspired by the Ubicomp vision of seamless interaction between devices, device composition explores how heterogeneous devices can interoperate smoothly with each other. Initially under the name of "smart spaces", a number of projects (e.g. Augmented Surfaces [20], i-LAND [27] or Interactive Workspaces [9]) investigated this direction. From an architectural perspective, smart spaces were conceived as closed environments that rely on a centralized software infrastructure for coordination and control. Examples are BEACH, which supports the i-LAND project [30], and Gaia OS supporting Active Spaces [23]. The infrastructure facilitates the sharing of resources between the devices involved, including displays and peripherals, but also storage and computation. The advantage of working with a centralized infrastructure is that the interaction between devices can be optimized as they share a set of semantics (e.g. priorities, quality of service, etc.), and control mechanisms can take complex forms (authorization and authentication). However, the drawback is that new devices cannot be integrated easily, as they require specific software configurations.

More recent work on device composition aims at replacing centralized infrastructures by an ad-hoc peer-to-peer approach [11] or by creating a virtual or composite device. In the ad-hoc approach, devices communicate directly with each other to negotiate access to a certain resource (e.g. computer looking for a printer); each device makes decisions about resource allocation locally without the involvement of a central infrastructure. The virtual or composite device approach, seeks to aggregate resources from all the involved devices into a single virtual entity [2, 12, 13, 25]. Common to these two novel approaches is that they have new concerns aside from architecture of the underlying infrastructure, and include human-centered needs like user preferences [16] and manageability [17].

In TIDE, the smartphone and the tabletop come in direct communication with each other without depending on a central infrastructure, thus being closer to the second wave of device composition research. However, our approach differs from previous work in that it is lightweight: it is limited to the tabletop replicating the display of the smartphone and channeling touch events; neither the smartphone nor the tabletop can access any other resources, like computational power or storage, on the other device. Moreover, the composition is started only when the devices come into direct contact, i.e., the smartphone lies on the tabletop.

3.2 Tabletop Augmentation

The tabletop community has demonstrated repeatedly the value of going beyond simple touch interaction. For instance, augmenting tabletops with keyboards and mice can facilitate tasks for which touch input is not well suited like typing or selection of distant items [6]. Augmenting tabletops with tangibles enables users to control the state of tabletop applications with physical items and leverage the intrinsic properties of physical objects [10] or to display information attached to these objects [6]. Finally, tabletops have been integrated into larger ecologies of devices for tasks like meeting support [33], laboratory work [29] or collaborative search [14].

The most straightforward way to augment tabletop interaction with other devices is to leverage users' smartphones and transform them into tangible inputs. For example, Bluetable [34] connects wireless mobile devices to interactive surfaces using vision-based handshaking and Bluetooth. Phonetouch [24] combines image recognition with phones accelerometer data, by asking users to tap 3 times with their phone on the tabletop to initiate a Bluetooth connection between the two devices. Both projects allow for easy transfer of content (e.g. pictures) between the smartphone and the tabletop and from there to other smartphones.

TIDE differs from previous efforts to augment tabletops by focusing on giving access to the smartphone applications from the tabletop, rather than using the smartphone as a peripheral, as a control device or as a data source. Through the proposed lightweight device composition, TIDE not only provides the same functionalities of other smartphone-tabletop integration approaches (file sharing, picture viewing, etc), but also does so through the smartphone's interfaces the user is already familiar with.

3.3 Smartphone Projections

Another research line studied the opportunities of extending user-interfaces onto larger surfaces by projecting it either physically or virtually [19]. The recent availability of pico-projectors led to a number of projects projecting the screen of smartphones onto any surface. SixthSense [15] projects the interface on any object right in front of the user. Winkler et al. use the projected interface to provide extra features to the phone application [35], for example it extends phone calls with synchronous remote collaboration features on the projected interactive surface. Virolainen et al. enhance a docking station with a projector-based vertical FTRI touch-display [31] in order to increase the screen size of the phone. Nonetheless, pico-projectors have intrinsic properties which affect their usage: they need to be held in a specific and stable manner to provide a convenient interactive space, their luminosity and resolution is limited which inhibits rich UIs, and their energy consumption is significant.

Virtual projection is an alternative to circumvent some of the limits of pico-projectors. In its simplest form, a number of commercial smartphones can mirror their display on Television screens (e.g. iOS devices with AppleTV's AirPlay feature[2], Sony's Xperia devices[3], etc.). However this only projects the phones' output to the TVs and does not support input from the TV to the phone. Baur et al. proposed to virtually project the smartphone UI onto a larger one [3]. Here, optical projection serves as a metaphor for multi-device interaction: the user positions the smartphone toward the desired target surface, and the surface renders the device's interface taking into account its position and orientation in a way that resembles a physical projection. While this is an interesting direction, the users' engagement with the target surface is limited as they need to hold the smartphone, a requirement which in the long term could cause arm fatigue and thus reduced usage. Moreover, the user interacts on the phone and not on the projected surface, something which affects the system's usability in a collaborative situation.

TIDE is inspired by the simplest form of virtual projection, but adds the possibility to interact with the content from the target surface. Furthermore, the user does not need to hold the smartphone to control the projection, providing a stable replication, reducing the muscular strain and liberating both hands for interaction. Moreover, the luminosity is normally higher on a tabletop than it is with a pico-projector.

[2] https://www.apple.com/appletv/airplay/
[3] http://www.sonymobile.com/gb/xperia/

4 Tabletop Interactive Display Extension - TIDE

TIDE relies on principles of lightweight device composition: it merely pushes the display of the phone to the tabletop and pushes touch input from the tabletop to the smartphone. TIDE's approach to device composition does not involve other resources like storage, computation, or peripherals. This section presents a technical overview of TIDE focused on the requirements for lightweight device composition.

To initiate the screen replication phone and tabletop must be paired. We aimed at making this process transparent, thus users simply have to put their phone on the tabletop (fig. 2-A). Once TIDE detects a smartphone on the tabletop surface, it tries to connect to it and asks the user to confirm that s/he wants to establish the connection via a simple dialog on the tabletop (fig. 2-B). After the user validates the pairing, the display of the phone is replicated on the tabletop (fig. 3). At this point the phone can be moved to the side, and the user can manipulate and interact with the replicated smartphone on the tabletop (R6).

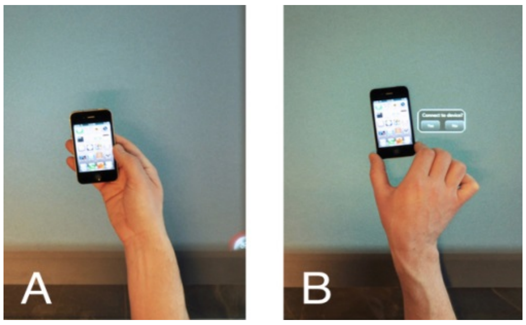

Figure 2: Pairing a TIDE-enable iPhone to a Microsoft Surface tabletop.

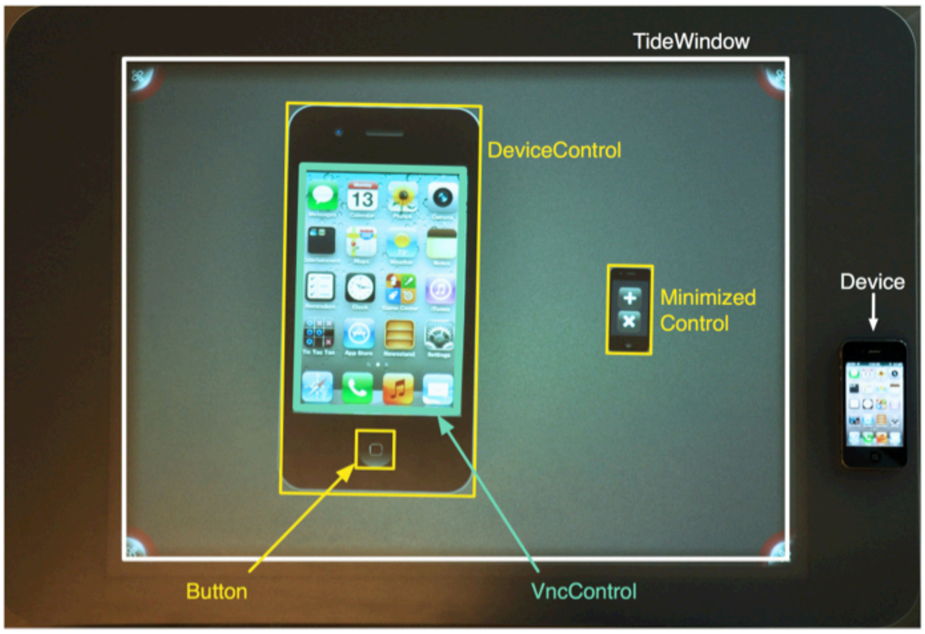

Figure 3: TIDE Surface UI

4.1 System Components

At its core, TIDE relies on Virtual Network Computer (VNC) [21] to replicate the smartphone's user interface on the tabletop. A VNC server runs as a background process on the smartphone, and streams its screen-capture to a VNC client running on the tabletop (see green elements in fig. 4). Discovery and pairing relies on a vision-based tracking algorithm running on the tabletop (see blue elements in fig. 4). Finally, all user interactions are handled by standard WPF components for Microsoft Surface applications (see yellow elements in fig. 4).

Figure 3 shows the interactive elements of TIDE on the tabletop. The `TideWindow` is the main application window covering the whole surface of the tabletop. It contains UI elements that are enabled for touch-based interaction such as dragging, rotating and resizing. Each paired smartphone is associated with a virtual device implemented by a `DeviceControl` object which contains the replicated UI. The `DeviceControl` is responsible for detecting touch inputs that are destined to manipulate the replicated UI (i.e. capturing the touch events that are to be channeled to the smartphone). To provide consistency in the user experience, the `DeviceControl` has the visual aspect of the body of the smartphone. We considered less literal alternatives and discuss this choice in more details in section 5, but this one provide a set of benefits: users appreciated it and could quickly control the virtual phone e.g. scaling, rotating, hiding or accessing the physical buttons of the phone, such as the 'home' button of the iPhone.

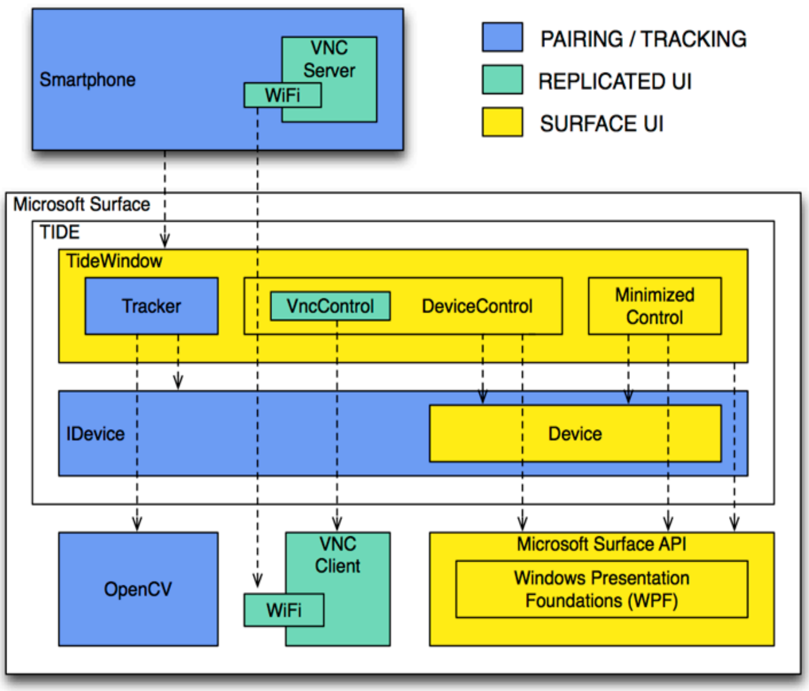

Figure 4: TIDE component diagram

4.2 Implementation

Discovery and Pairing: The tracking component of TIDE is responsible for discovering smartphones by leveraging the MS Surface v1.0 infrared cameras. It uses shape recognition to discover new smartphones but also of keeping track of them during the application session. TIDE leverages the OpenCV shape recognition libraries to identify specific features of the phones' casings in order to track the phones' position and orientation on the tabletop. Once tracked, the pairing process relies on a simple approach which consists in attempting to connect to a series of known IP addresses on the local wireless network (R3). This simplistic approach enables an efficient discovery and pairing in testing and demonstration scenario, but would not scale well.

Replicated UI: We implemented UI replication with VNC in order to make TIDE compatible with all phone applications (R1). A VNC server runs on the smartphone, and the tabletop's TIDE implementation connects an embedded VNC client to it. The UI is streamed over Wifi, requiring the two devices to be connected to the same network. Once connected, the server sends a pixel-based replication of the phone's screen to the VNC client. TIDE receives the updates from the client via the VncSharp library and displays them on the tabletop. TIDE can create multiple client instances and connect to different smartphones (R2).

Surface UI: The Surface UI is implemented with the Surface SDK, which is itself based on WPF (Windows Presentation Foundations). The virtual phone is implemented by a `DeviceControl` object, which extends the Surface `ScatterView` control. The `ScatterView` class provides manipulation capabilities that are extensively used by the TIDE interaction model. The `DeviceControl` contains a `VncControl` object (from the VncLibrary), which displays the actual phone UI replication. Touch inputs to the virtual phone's screen are processed by the `VncControl` which maps their precise location on the screen and relays them to the VNC server on the phone.

4.3 Technical limitations

Pairing and Discovery: TIDE uses shape recognition to discover new smartphones placed onto the tabletop. Once it detects a smartphone on the surface, TIDE tries to connect to a list of known IP-addresses for each device type (e.g. iPhone 4, HTC Desire, etc.) We took this approach, as the main focus of our work is the interaction with the replicated UI. However, a number of alternatives are available to provide a more robust discovery and pairing process in a real-life situation: Device discovery can be implemented using e.g. mDNS/Bonjour or UPnP and taking into consideration properties like phone type, location, and/or accelerometer data. Another approach would be to leverage the new Near Field Communication [32] of recent smartphones to exchange pairing information when being in close proximity of the tabletop.

UI Replication Protocol: TIDE builds upon VNC, which has the benefit of being available on most platforms and very stable. For instance, VNC is readily available on most mobile OSes, through third party applications. Moreover, because VNC is a pixel-based protocol is does not require any modification of existing applications.

However, VNC comes with a set of shortcomings, mostly due to the fact that it was designed to allow remote computing in a more “traditional” personal computer paradigm with keyboard, mice, WIMP applications, and stable/fast connectivity. Therefore, our implementation of TIDE only supports single-touch interaction. Moreover, VNC presents delays in the screen updates when a lot of visual changes take place on smartphones with high screen resolution (i.e. Retina displays).

Alternatives to mitigate these limitations include providing an improved VNC implementation and reducing features in the user-interface before registering small changes as updates.

Pixel Density: The resolution of modern smartphones is comparable to the ones of tabletops[4]. However the pixels per inch of the two classes of devices are significantly different: a tabletop pixel is considerably bigger than one from the smartphone. Moreover, when the replicated UI is bigger on the tabletop, the fidelity of the image is limited by the pixel density on the smartphone: a pixel on the phone is represented by several pixels on the tabletop; causing the replicated UI to look less sharp than it is.

Screen Size: Because of pixel-based replication and important differences in screen sizes, scaling up applications to take over the full screen of tabletops is not always valuable and depends on the type of applications. Applications based on default mobile widgets should be mostly used at a size equivalent to the one of the phone. Whereas applications with dense information, e.g. documents, maps or games can be run at larger size (e.g. full-screen) without suffering from the lower pixel density of tabletop screens.

5 Interaction Design Study

This section presents TIDE’s interaction design, a process focused on discovering the main elements of the user experience: *how users would interact with the phone’s representation on the tabletop*. Our goal was to maximize discoverability of the functionalities for novice users and foster spontaneous interaction. We took inspiration from Wobbrock et al.’s elicitation study of user-defined gestures for surface computing [29]. The authors conducted a study aiming at discovering the gestures (i.e. *actions*) participants tended to do to trigger specific behaviors (i.e. *commands*) on the tabletop, like scaling an image. Users were presented the *commands* and asked to perform the *action* they believe should be the cause.

Our approach is similar, but extends classical elicitation studies in that we wanted to incorporate the study results in a working prototype while maintaining a coherent set of commands. In our study, we thus took both a bottom-up approach (i.e. classical elicitation study, by asking participants what they would do to trigger a command) and a top-down approach in which we curated the actions before and after the study to maintain consistency among the different commands and actions.

[4] For instance, the resolution of Microsoft’s PixelSense screen is 1920×1080, whereas recent Android phones often have a resolution of 1280×700 and the latest iPad a resolution of 2048×1536.

5.1 Interaction Design

Commands are the manipulations that can be completed on a TIDE window on the tabletop. We defined the following list of commands, as the minimal set needed for using TIDE:

1. *Dragging* the replicated UI across the interactive surface.
2. *Rotating* the replicated UI across the interactive surface.
3. *Resizing* the replicated UI across the interactive surface.
4. *Minimizing* the replicated UI, and restoring it.
5. *Hiding* the content of the replicated UI.
6. *Closing* the replicated UI (i.e. disconnecting.)

Actions are the way users can trigger the commands. Unlike Wobbrock et. al's study which focuses on gestures (i.e. no visual cues) [36], TIDE relies on visual elements. We created ***action controllers***, which are user-interface elements that allow the user to issue actions. We used sketches, storyboards and prototypes to envision different *action controller*. For each *action controller* there was a total of six possible *actions*. We focused on *action controller* that would be consistent with both the interaction experience on a smartphone and a tabletop (see fig. 5):

1. *Action Tabs* are traditional buttons/tabs that implement functionalities.
2. *Window Toggle* uses a switch to toggle the window between inactive and active states. In its inactive state the window can be handled like a digital picture.
3. The *Action Bar* is a manipulation area which resembles to a virtual touch-pad.
4. The *Active Border* is a digital frame around the 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.

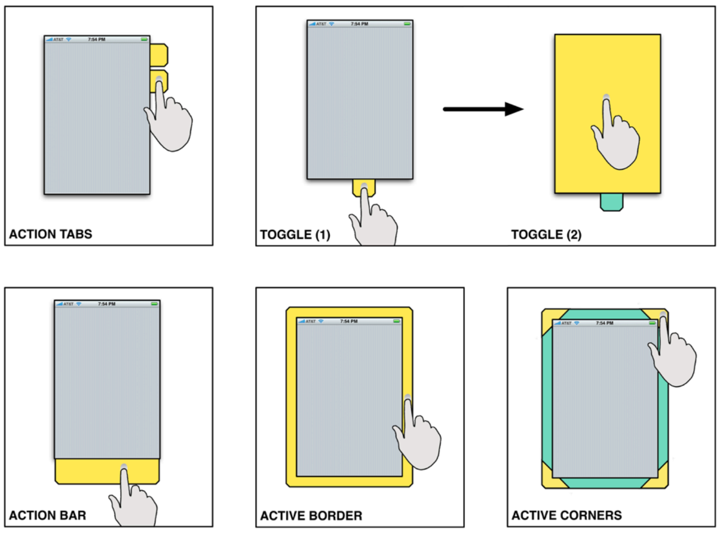

Figure 5: Action controllers

For the purpose of the study, we created a 6th action controller called *'Other'* grouping *actions* that did not correspond to any of the five action controller, but were well suited to a specific *command*:

1. Drag by holding a finger on a specific tab, and using another finger to tap a destination target to move the window,
2. Rotate by performing a one finger dragging gesture on a corner of the window,
3. Resize by pulling the window apart with both hands,
4. Minimize by dragging the window to the bottom of the surface,
5. Hide by placing and holding a hand on the window,
6. Close by dragging the window to a specific location on the surface, and
7. Close by double tapping the window.

5.2 Elicitation Study

We conducted an elicitation study to investigate how participants would link the *actions* to the *commands*. We used paper mockups representing the phone interface, i.e., the replicated UI, and the action controllers (see [26] for details and images).

Participants: We recruited 12 participants aged between 25 and 35 years old. All of them were smartphone users. Three had prior tabletop experience.

Setting: An experimenter sat in a laboratory room with a participant, facing each other across a Microsoft Surface v1.0. A camera was used to record the sessions for future reference.

Procedure: Based on a script, the experimenter first presented an early version[5] of TIDE (non-interactive) to participants and the paper mockups of the iOS interface that participants could manipulate. The experimenter asked participants to envision writing an email on the tabletop with TIDE. This email-writing scenario was split into six steps. Each step required the user to issue a command. The experimenter used the following process to elicit every *action* from the participants:

1. Explain the desired command to the participant.
2. Ask first the participant to express an open-ended suggestion, i.e. suggest an action that s/he would perform to obtain the desired effect, and to demonstrate the action using the paper mockups.
3. Present three possible *actions* to issue the desired *command*, ask the participant to try them out to make sure they were understood. Then ask participants to rank the actions by order of preference.

Data analysis: We ranked the {*command*, *action*} pairs based on the preference they received from participants. A weight of 3 was given to the first position, a weight of 1 to the second, and a weight of 0 to a third. We then aggregated the preferences of all the users and normalized them on an [0-1] interval, where a 1 meant that the entry was awarded a first position by all participants, and a 0 meant that all participants

[5] Video of the early prototype: https://www.youtube.com/watch?v=EVWndZgTnPQ

ranked the entry third. Table 1 summarizes the normalized scores, with colored cells containing values above 0.6, a score that can only be obtained if half the participants awarded it first position.

We also registered the participants' suggestions and counted how many users independently expressed a specific suggestion.

	Action Tabs	Window Toggle	Action Bar	Active Border	Active Corners	Other
Dragging	0.50	0.22	0.61	0.89	0.44	0.00
Rotating	0.06	0.17	0.56	0.61	0.78	0.50
Resizing	0.56	0.17	0.22	0.50	0.67	0.56
Minimizing	0.44	0.00	0.72	0.67	0.00	0.67
Hiding	0.33	0.39	0.17	0.78	0.44	0.56
Closing	0.28	0.50	0.06	0.50	0.33	1.00
Avg	0.36	0.24	0.39	0.66	0.44	0.55

Table 1: Normalized weighted average rank given to each pair (command, action controller).

5.3 Results

We can split the commands into two groups. Commands from the first group have a concrete visual signification, i.e., dragging, rotating and resizing. Commands from the second group are more abstract, i.e., minimizing, hiding, and closing.

For the first group, there is a strong coherence in the participants' choices. The favored *action controller* are the active border, the action bar and active corners. All three require the user to interact with an area directly around the window in order to manipulate it and modify its position, orientation or size. These interaction techniques are similar to the current standard for manipulating pictures on interactive touch screens. However in the present case, participants avoided touching the replicated UI because of its role as input relay between the tabletop and the smartphone.

For the second group, the action bar and active border also scored high, even though there is no apparent relation between the visual aspect of the strategy and the effect implied by the command. Looking closer at the results, participants preferred the double-tap for closing the window, possibly because it is a common technique in many other application contexts, as well as a quick and easy to execute.

The only '*Other*' strategies that scored above 0.6 are related to minimizing and closing the replicated UI. Both strategies, involve dragging the window to a specific location on the surface. This suggests that moving the window off screen is a natural way to remove focus from the application. Interestingly, this correlates with the analysis of the user suggestions, presented hereunder.

5.4 Participants' Suggestions

When asked to interact with TIDE for the first time, participants intuitively reached within the replicated UI to perform a dragging gesture with one or more fingers. However, any touch inside the replicated UI is forwarded to the smartphone. It was therefore necessary to stress again the distinction between replicated UI and surface

UI. Even though TIDE had been described to the participants their first reaction was still to actually interact with the phone's content. This finding suggests that for TIDE to be really intuitive, it should be able to interpret the intention of the user: to control the phone image or to control the phone remotely.

From the variety of ideas participants suggested, we identified a clear trend in two situations. For *resizing*, 8 out of 12 participants suggested grabbing the sides of the window with two fingers, and pulling the window apart to enlarge it. This shows that pinch and zoom is now completely part of users' vocabulary. For *minimizing*, 7 out of 12 participants suggested dragging the window off-screen (or to a specific location along the surface edge). The same suggestion reoccurred for the hiding and closing commands, although less strongly. This action is consistent with removing a real piece of paper from a table and appeared to be intuitive.

5.5 Final Design Choices

Based on the participants' rankings and suggestions we implemented the active borders and active corners (for manipulating and resizing – R5), and double tapping (for quick closing – R4). This selection represents the highest ranked *action controller* across all commands and the most popular suggested *actions*.

6 Usability Study

We conducted a final usability evaluation focusing on the learnability, ease of use and usefulness of the system. Our goal was to identify which *actions* and *commands* participants would discover in our working prototype

Participants: We recruited 10 participants aged between 25 and 40 years. All were regular smartphone users, and 5 had prior tabletop experience.

Apparatus: TIDE was installed on the Microsoft Surface v1.0, that has a 30 inch display (76cm) with a resolution of 1024 x 768 pixels. The tabletop is at the height of a coffee table, and the participants sat by it. It was used in combination with an iPhone 4 running iOS 5 and a HTC Legend running Android 2.1, both equipped with third-party VNC applications. An additional desktop computer was available for filling out a questionnaire.

Data Collected: In order to determine which *actions* were used to perform a given *command*, we used simple logging at the application level. Each session produced a set of CSV log files capturing the *command*, *action* and time. We also gathered participants' feedback through a questionnaire after each session, and the sessions were captured on video, for future reference.

6.1 Procedure

We first introduced the experiment to participants, explaining they would go through the following phases:

Exploration: The participants had three minutes to *learn by doing,* i.e. explore the system and discover its features on their own. We recorded which *actions* they discovered in order to evaluate system learnability for users with no prior knowledge of TIDE.

Guided test: We asked participants to perform specific tasks with the application, in order to evaluate ease of use. The given instructions only included *commands*, and it was up to the participants to decide which *actions* to use.

Questionnaire: Participants filled out a questionnaire, allowing us to gather data assessing the usability and usefulness of TIDE.

6.2 Results

During the exploratory phase, all participants discovered at least one *action* for performing each *command*; suggesting TIDE is highly discoverable. Participants also felt subjectively that TIDE was easy to learn; this statement received a median score of 4.46 (sd=1.08) on a Likert scale from 5 Strongly Agree to 1 Strongly Disagree. The basic *actions* to *drag*, *rotate* and *resize* the application window were discovered by all participants in the exploration phase. This shows that tapping, dragging and pinching have now become part of a shared vocabulary among users.

When asked to select their favorite interaction techniques (*action* or *action controller*), the users did not necessarily choose the easiest one to discover. For instance, 8 users discovered the minimize *action* by resizing the window down, but only 1 selected it as a favorite way of activating this *command*. On the other hand, 6 users chose the double tap, which was as easily discovered, but quicker to perform. This finding emphasizes the importance of implementing multiple interaction techniques (*actions*) for a *command* as a good way to provide both discoverability and efficiency.

We evaluated the usefulness of TIDE by asking study participants, as well as 3 participants of the elicitation study, to assess in which situations and contexts they would

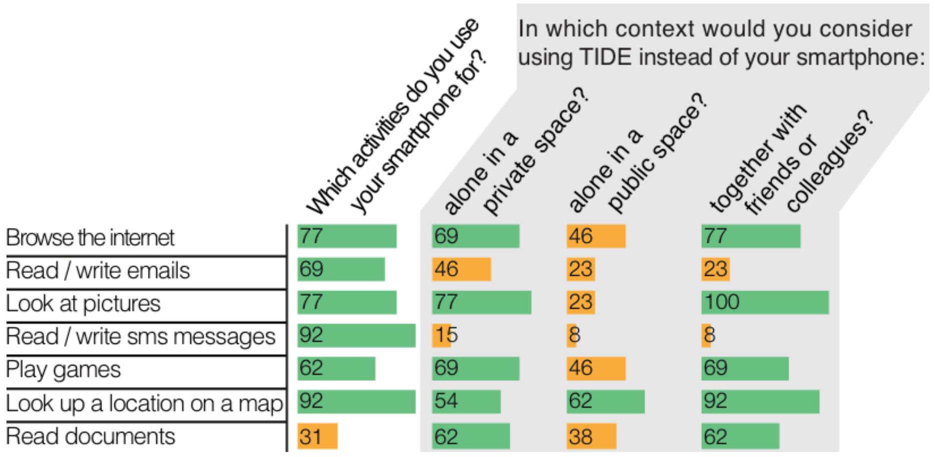

Figure 6: The situations and contexts in which TIDE is found useful (Values expressed as percentage of the participants).

consider using the application. Figure 6 provides an overview of the situations in which TIDE made sense (or did not) to the participants. Participants' feedback clearly showed that a system like TIDE is not suited for interacting alone in a public space, nor does it seem well suited for interacting on tabletop like one would on a smartphone. The activities for which TIDE shows relevance are browsing the Internet, looking at pictures, playing games, looking at maps and reading documents; especially in collaborative situations. Such situations are not currently supported well by smartphones, not because of a lack of features but because of the inherent limitations of their small form factor.

7 Discussion and Perspectives

Participants' feedback outlines situations where TIDE would be useful in everyday life, like in private spaces or collaborative settings. This observation supports our initial motivation for developing TIDE as a platform to complement tabletop deployments in professional settings which are either private workspaces or semi-public ones with only known colleagues accessing the information. In this section we discuss the implications of TIDE for tabletop deployments in the real world, its benefits, and possible venues of future research.

7.1 TIDE in the World

Based on participants' informal feedback, we identified three situations beyond the professional environment in which participants would consider using TIDE:

1. At home, TIDE would play the role of a leisure device, to browse, play in a more public and shared manner than one would with a phone or a tablet.
2. In shared spaces, such as meeting rooms for collocated browsing of documents.
3. In public spaces but in a semi-public way, e.g. around a table with friends in a café or bar for casual browsing, playing, photos; while the space is public, the situation offers some degree of protection to intruders.

These situations add to our original motivation of providing touch-friendly applications in professional settings, to augment one's interactive desk.

7.2 TIDE Benefits

Compared to running tabletop based applications, by having applications run on users' smartphones, TIDE provides a set of interesting benefits:

1. **Users' personal data does not leave the phone**. This is of particular value given the shared nature of tabletops. By having applications run on the phone, users can be sure that their data is not shared without their consent, that it will not stay cached in the tabletop, and that it's not subject to eavesdropping during transport.
2. **No code coming from the tabletop runs on the users' smartphones**, by only transmitting input information from the tabletop to the phone and using a "*dumb*"

protocol unaware of the content being transmitted, risks of accessing personal data are quite low.

3. TIDE provides users and developers with a **single-point of updates** to manage. Getting the latest version of an application happens only at the smartphone. This solves one of the usual problems of Ubicomp applications: the complexity of maintaining them and keeping them up-to-date.

7.3 Tabletop-Aware Phone Applications

Having applications installed on smartphones running through TIDE does not necessarily imply that such applications should be designed only for smartphones. We envision that TIDE could be extended so that applications running on the smartphone could be targeted for tabletop use. For example, UI elements could be adjusted to the number of pixels per inch of the tabletop, which is significantly lower than on smartphones. Other extensions could focus on the collaborative aspect of tabletops, for instance by allowing users to drag and drop data from on smartphone to another through TIDE.

This would mean extending VNC or developing a new framework dedicated to UI distribution like Substance [5]. Or like XICE [1], which is a programming framework supporting the development of applications enabling the annexation of displays by nomadic users.

7.4 Tabletop as a Hub for Peripherals

Finally we envision TIDE as a first step towards using tabletops as device composition hubs, offering extra computational power, faster connectivity, larger memory but also extending smartphones' input and output capabilities, by enabling users to plug-in external devices like keyboards and mice.

8 Conclusion

In this paper, we have introduced the notion of lightweight device composition and identified some of the requirements for its implementation. We presented TIDE, a middleware to provide such lightweight device composition between a smartphone and a tabletop computer. We presented the interaction design process of TIDE based on an elicitation study. We further carried out an evaluation to assess the discoverability of TIDE features. Finally we presented compelling cases in which TIDE could be used.

We observed that TIDE often generates a lot of questions about privacy related to the personal nature of smartphones. Nonetheless, after interacting with TIDE, users felt in control and did not voice any concern. This may be related to the fact that they had become familiar with the commands to hide or disconnect TIDE, and felt comfortable triggering them. In this sense, the privacy questions were rather at the application level, here users clearly stated that they would not use TIDE to view or write personal messages in a public setting, but would rather use TIDE for collaborative activities in which smartphones are not well suited. Moreover, we believe that social conventions play a role in controlling the privacy element, like it is the case today when a smartphone or tablet is shared among friends to show photos or play games.

References

1. Arthur, R. and Olsen, D.R. Jr.: Xice windowing toolkit: Seamless display annexation. ACM Trans. Comput.-Hum. Interact., 18:14:1–14:46, (2011)
2. Bardram, J. E., Fuglsang, C., & Pedersen, S. C.: Compute: a runtime infrastructure for device composition. In Proc. AVI 2010.
3. Baur, D., Boring, S., and Feiner, S. Virtual projection: Exploring optical projection as a metaphor for multi-device interaction. In Proc. CHI 2012, ACM (2012).
4. Chehimi, F., and Rukzio, E. Throw your photos: an intuitive approach for sharing between mobile phones and interactive tables. In Proc. Ubicomp 2010, pp 443–444. ACM (2010)
5. Gjerlufsen, T., Nylandsted Klokmose, C., Eagan, J., Pillias, C., and Beaudouin-Lafon, M.: Shared substance: developing flexible multi-surface ap- plications. In Proc. of CHI'11.
6. Hartmann, B., Morris, M. R., Benko, H., and Wilson, A.: Augmenting Interactive Tables with Mice and Keyboards. In Proc. of UIST 2009, ACM (2009)
7. Hartmann, B., Morris, M. R., Benko, H., and Wilson, A. D.: Pictionaire: supporting collaborative design work by integrating physical and digital artifacts. In Proc. of CSCW '10.
8. Hunter, S., Maes, P., Scott, S., and Kaufman, H.: MemTable: an integrated system for capture and recall of shared histories in group workspaces. Proc. of the SIGCHI conference on Human factors in computing systems, CHI'01, pp. 3305--3314, ACM (2001)
9. Johanson, B., Fox, A., and Winograd, T.: The interactive workspaces project: experiences with ubiquitous computing rooms. In Pervasive Computing, 1(2): pp. 67--74, IEEE (2002)
10. Jordà, S., Geiger, G., Alonso, M., & Kaltenbrunner, M.: The reacTable: exploring the synergy between live music performance and tabletop tangible interfaces. In Proc. of TEI'07.
11. Karmouch, E., and Nayak, A.: A distributed protocol for virtual device composition in mobile ad hoc networks. In Proc. of ICC'09, pp. 1-6. IEEE (2009)
12. Lee, J., Kim, S., Lim, H., Schuster, M., and Domene, A.: A software architecture for virtual device composition and its applications. Ubiquitous Computing Systems (2007).
13. Lyons, K., Pering, T., Rosario, B., Sud, S., & Want, R.: Multi-display composition: Supporting display sharing for collocated mobile devices. In Proc. of Human-Computer Interaction–INTERACT 2009, pp. 758-771 (2009)
14. McGrath, W., Bowman, B., McCallum, D., Hincapié-Ramos, J.D., Elmqvist, N., and Irani P.: Branch-explore-merge: facilitating real-time revision control in collaborative visual exploration. In Proc. of ITS '12, ACM (2012)
15. Mistry, P., and Maes, P.: SixthSense: a wearable gestural interface. In ACM SIGGRAPH ASIA 2009 Sketches p. 11. ACM (2009)
16. Mukhtar, H., Belaïd, D. and Bernard, G.: User Preferences-Based Automatic Device Selection for Multimedia User Tasks in Pervasive Environments. In Proc. of the 2009 Fifth International Conference on Networking and Services, ICNS '09.
17. Newman, M., Elliott, A., & Smith, T.: Providing an integrated user experience of networked media, devices, and services through end-user composition. In Proc. of Pervasive Computing, pp. 213--227 (2008)
18. Olwal, A. and Wilson, A. D.: Surfacefusion: unobtrusive tracking of everyday objects in tangible user interfaces. In Proc. of GI'08, pp. 235--242, Toronto, (2008)

19. Pinhanez, C.: The everywhere displays projector: A device to create ubiquitous graphical interfaces. In Ubicomp 2001: Ubiquitous Computing. Springer, (2001).
20. Rekimoto, J. and Saitoh, M.: Augmented surfaces: a spatially continuous work space for hybrid computing environments. In Proc. of CHI '99, pp. 378--385, ACM (1999)
21. Richardson, T., Stafford-Fraser, Q., Wood, K. R., and Hopper, A.: Virtual network computing. In Internet Computing, pp. 33--38 IEEE 2 (1998),
22. Rogers, Y., Hazlewood, W., Blevis, E., and Lim, Y.-K. Finger talk: collaborative decision making using talk and fingertip interaction around a tabletop display. In Proc. of CHI E.A.'04, pp. 1271--1274, ACM (2004)
23. Roman, M., Hess, C., Cerqueira, R., Ranganathan, A., Campbell, R.H., and Nahrstedt, K.: A middleware infrastructure for active spaces. Pervasive Computing, IEEE, 1(4): 74 - 83, (2002).
24. Schmidt, D., Chehimi, F., Rukzio, E., and Gellersen, H. Phonetouch: a technique for direct phone interaction on surfaces. In Proc. UIST 2010, pp. 13--16, ACM (2010)
25. Schuster, M., Domene, A., Vaidya, R., Arbanowski, S., Kim, S. M., Lee, J. W., & Lim, H.: Virtual device composition. In Proc. of the Eighth International Symposium on Autonomous Decentralized Systems, ISADS'07. pp. 270--278. IEEE (2007)
26. Sicard, L.: TIDE - Using Device Composition on Tabletop Computers to Extend the Smartphone Experience, Master Thesis, IT University of Copenhagen, 2012.
27. Streitz, N. A., Geissler, J., Holmer, T., Konomi, S., Müller-Tomfelde, C., Reischl, W., Rexroth, P., Seitz, P., and Steinmetz, R.: I-land: an interactive landscape for creativity and innovation. In Proc. of CHI '99, pp. 120--127, ACM (1999)
28. Tabard, A., Hincapié-Ramos, J. D., and Bardram, J. E.: The eLabBench in the wild: supporting exploration in a molecular biology lab. In Proc. of CHI '12, pp. 3051–3060, ACM (2012)
29. Tabard, A., Hincapié-Ramos, J. D., Esbensen, M., and Bardram, J. E.: The elabbench: an interactive tabletop system for the biology laboratory. In Proc. of ITS'11, pp. 202--211, ACM (2011)
30. Tandler, P.: Software infrastructure for ubiquitous computing environments: Supporting synchronous collaboration with heterogeneous devices. In Ubicomp 2001 pp. 96--115, Springer (2001)
31. Virolainen, A., Paldanius, M., Lehtiö, A., Häkkilä, J.: Projector-based Multi-touch Screen for Situated Interaction with a Mobile Phone. In Mobile and Personal Projection workshop at CHI'11 (2011).
32. Want, R. Near Field Communication (**NFC**). In IEEE Pervasive Computing, Smart Phone Dept, Vol. 10, No. 3, pp. 4--7, IEEE (Jul-Sep 2011)
33. Wigdor, D., Jiang, H., Forlines, C., Borkin, M., and Shen, C.: WeSpace: the design development and deployment of a walk-up and share multi-surface visual collaboration system. In Proc. of the SIGCHI Conference on Human Factors in Computing Systems, CHI '09, pp. 1237--1246, ACM (2009).
34. Wilson, A. D., and Sarin, R. Bluetable: connecting wireless mobile devices on interactive surfaces using vision-based handshaking. In Proc. GI 2007, pp. 119--125ACM (2007)
35. Winkler, C., Reinartz, C., Nowacka, D., and Rukzio, E.: Interactive phone call: synchronous remote collaboration and projected interactive surfaces. In Proc. of ITS '11.s
36. Wobbrock, J. O., Morris, M. R., and Wilson, A. D.: User-defined gestures for surface computing. In Proc. of CHI '09, pp. 1083--1092, ACM (2009).