THAW: Tangible Interaction with See-Through Augmentation for Smartphones on Computer Screens

Sang-won Leigh Philipp Schoessler Felix Heibeck Pattie Maes Hiroshi Ishii
{sangwon, phil_s, heibeck, pattie, ishii }@media.mit.edu
MIT Media Lab, 75 Amherst Street, Cambridge, MA, United States

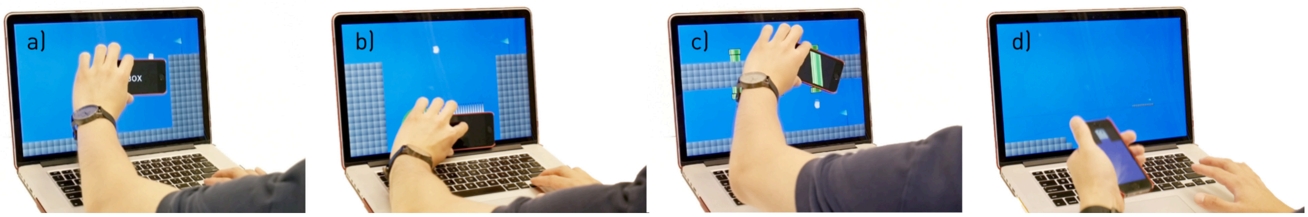

Figure 1: A smartphone screen can be used as a user interface intervening into the display space of a computer screen.

ABSTRACT

The huge influx of mobile display devices is transforming computing into multi-device interaction, demanding a fluid mechanism for using multiple devices in synergy. In this paper, we present a novel interaction system that allows a collocated large display and a small handheld device to work together. The smartphone acts as a physical interface for near-surface interactions on a computer screen. Our system enables accurate position tracking of a smartphone placed on or over any screen by displaying a 2D color pattern that is captured using the smartphone's back-facing camera. As a result, the smartphone can directly interact with data displayed on the host computer, with precisely aligned visual feedback from both devices. The possible interactions are described and classified in a framework, which we exemplify on the basis of several implemented applications. Finally, we present a technical evaluation and describe how our system is unique compared to other existing near-surface interaction systems. The proposed technique can be implemented on existing devices without the need for additional hardware, promising immediate integration into existing systems.

Author Keywords

Multi-device Interaction; Tangible Magic Lens;

ACM Classification Keywords

H.5.2 User Interfaces: Input devices and strategies (e.g., mouse, touchscreen)

TEI '15, January 16 - 19 2015, Stanford, CA, USA

ACM 978-1-4503-3305-4/15/01…$15.00
http://dx.doi.org/10.1145/2677199.2680584

General Terms

Human Factors; Design;

INTRODUCTION

A growing number of people own a smartphone in addition to their computer. The collocated interaction with those devices poses the question of how to seamlessly connect the different display spaces and their afforded interactions. Some existing systems mediate users' actions across multiple devices, however, their use scenarios are mostly focused on using a secondary device as mere a remote controller or a viewport [2, 3, 6].

To challenge this limitation, our research focuses on the spatial fusion of the two display devices through near-surface interaction. This allows to best leverage both device's affordances to create a fluid experience: The physical body of the phone affords tangible manipulation, while the screens on both devices can display virtual graphics that augment or interact with each other. If the interaction between the devices happens in close proximity, the phone's physicality and the graphics on each device in combination with our strong visual-motor skills bridges the gap between spatial reality and the digital as shown in prior research in the fields of Augmented Reality (AR) [10] and Tangible User Interfaces (TUI) [14]. The two domains are not mutually exclusive, having slightly different focuses on visual augmentation and tangible interaction respectively.

In this paper we present THAW (Tangible, Handheld, and Augmented Window), a system that enables near-surface interaction with ordinary computer displays and smartphones without any necessary hardware modifications. We present the underlying technology and provide a classification of possible interactions. The implemented applications explore scenarios to transfer digital content between two devices, novel game mechanics and map navigation with multiple users.

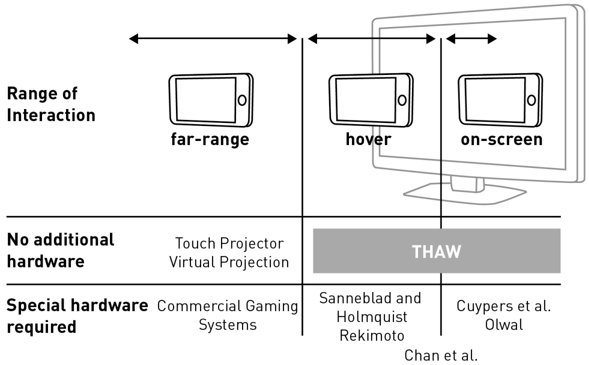

Figure 2: Systems based on tracking of handheld devices. THAW fills the void of systems that offer near-surface interaction and are easy to deploy

RELATED WORK

Prior see-through augmentation styled interaction systems explored interactions between a phone and a computer screen over a certain distance, however, none thoroughly explored near-surface (on-screen) interactions. This is mainly because the proposed tracking techniques require a certain distance between the devices to work properly or special tracking hardware / setups. (Figure 2)

Handheld Displays with Large Display

Commercially available gaming systems like the Wii U provide a game controller consisting of a small touchscreen [2]. The controller acts as a position-tracked window that can display an additional layer of information. Touch Projector [6], by Boring et al., is a system that allows manipulating digital content on a large screen through touching on live video captured by smartphones. Baur et al. presented Virtual Projection [4], simulating projections of graphics onto computer screens inferring a phone's 3D position in space. Due to the nature of the tracking techniques used in these systems, the handheld screen can be operated only from a considerable distance to the screen. Particularly in [4, 6], the graphics on the computer screen are used as a tracking pattern, which has to be updated every frame creating significant latency.

There are systems that explored closer range interactions using mobile phones on top of a larger display. Rohs et al [23], Sanneblad and Holmquist [24], and Reilly et al. [20] explored the application of digital maps. Hansen et al. [12] used a fiducial marker on a display screen for tracking the position of a smartphone hovering above it. For the purpose of their studies, they used mock up systems [20], fiducial markers printed on a map [12, 23], or a pseudo tracking setup using a commercial touch pen [24], which were at a yet preliminary stage for practical uses. Chan et al. [8] and Cuypers et al. [9] presented hardware systems that project a tracking pattern from beneath a tabletop display. BlueTable [27] tracked phones on a tabletop using external cameras. The LightSense system [16] tracked a phone's flashlight with a camera installed behind a surface or with embedded light sensors. These systems allowed continuous tracking of the phone, but are quite complex in their setup and large in size, as additional hardware is needed.

Tangible and See-through Interaction

We consider our work situated at the intersection of AR and TUI, as the use of a handheld screen on a larger display enables both tangible interactions and see-through augmentation. Here we introduce previous and current research from each domain that shares similarities with the THAW system.

Physical Tokens

Physical tokens are widely used for tabletop computing systems. Underkoffler et al. presented Urp [28], a tabletop system that utilizes physical wireframe buildings to simulate environmental behaviors for urban planning. Sensetable [18] uses magnetic gadgets that can be actuated in 2D space by a computer-controlled electromagnet array. Liang et al. [15] presented Gaussbits, magnetic tokens that can be tracked using an array of bipolar-magnetic sensors. Capstone [7], are widgets that can be stacked on top of capacitive touchscreens. While those systems all showcase the utility of physical tokens to interact with digital data, they require special hardware.

Magic Lens and Touching Through Video

Bier et al. introduced Toolglass and Magic Lens [5], a software widget controlled by the non-dominant hand, and users can touch "through" the widget with their other hand. This concept of see-through interaction has then been further explored by using handheld screen devices that would act as the magic lens. Fitzmaurice and Buxton [11], as well as Rekimoto [22] used palm top displays to reveal information based on the devices' 3D positions in space. Spindler et al. [25] did a general study on the interaction using handheld magic lens displays in a tabletop system.

Tani et al. [26] presented Object-oriented Video, a concept of touching and controlling physical interfaces through a video feed. This concept has been iterated through many recent projects: the most notable is Smarter Objects [13] by Heun et al., which allows controlling and reprogramming the behavior of physical objects by overlaying the graphical UI when looking through the display of a handheld device.

Most of the listed projects explore different techniques for location aware interaction with remote objects through the video feed of a handheld device. However, none have taken an exhaustive investigation into the combination of see-through and tangible interaction, mainly due to the lack of a reliable system for such combined interaction.

INTERACTION

In our system, the surface of a screen represents the main space for interactions. A handheld screen device is put on top of a larger screen, allowing a user to manipulate the handheld device in a very close proximity to the larger

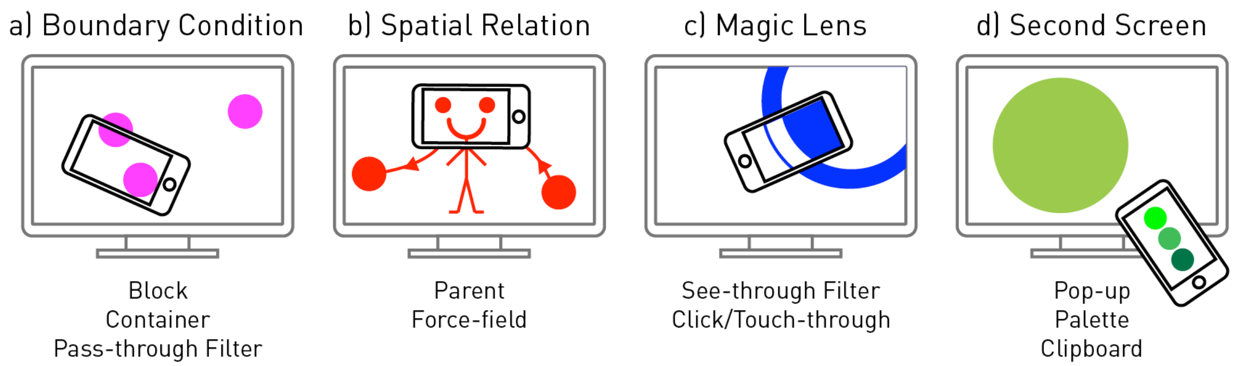

Figure 3: Classification of interaction on our system. Each represents different modes of direct, near-surface interactions.

screen. As the user moves the handheld screen, the graphics on both screens respond accordingly, right at the location of the user's action. Our strong visual-motor skills help us to make the perceptual connection between our hand's motion and the visuals on the screen, producing a believable experience of connecting both display spaces.

In this section we present our classification of possible interactions with the THAW system (Figure 3). The phone can be used as a physical token to directly interact with digital entities based on their relative positions (Figure 3(a), (b)). It can act as a lens for controlling or augmenting objects on a computer screen and also offers an additional space to be used for extended control or as a physical clipboard (Figure 3(c), (d)). The presented interactions can be building blocks for more sophisticated interactions as will be presented in next section.

Boundary Condition

Using a handheld screen device as a physical token lets us intervene "in" a larger screen and interact with its displayed graphics content. Similar to the computer game Roy Block [1], in which a virtual character needs to jump on physical blocks to solve a level. Interactions are classified based on the boundary condition that determines the entry direction into the handheld screen device. (Figure 3(a))

Block: The handheld screen device acts a physical boundary and has the ability to block and constrain elements on the screen. It can be used to manipulate objects in virtual space.

Container: The handheld screen device provides a complimentary space for containing data. The user can put data or virtual objects "into" their device, for example through drag and drop.

Pass-through Filter: Digital elements can travel "through" the handheld device in turn modifying their properties. For instance, this can be used to filter, translate, encode/decode certain data (sounds, images, etc.) as well as for compression and decompression.

Spatial Relation

This section defines the set of interactions based on the relative position of the handheld device to the graphics on the larger screen. (Figure 3(b))

Parent: The handheld device can act as a parent object providing a physical anchor for graphical elements on the larger screen.

Force-field Generator: The handheld device is able to generate a force field that affects the virtual objects on the larger screen by attracting or repelling them.

Magic Lens

Using the handheld screen device as a virtual window we can reveal certain information or content. Hidden annotations can be displayed while the users are browsing the Internet or editing a document. Additionally, the device's touch screen affords touch interactions with the displayed content. (Figure 3(c)) This enables Toolglass and Magic Lens type interaction [5].

See-through Filter: The handheld screen device becomes a lens for revealing or filtering displayed information.

Click/Touch-through: By putting the handheld screen device on items on the larger screen, users can touch or click the items through the handheld screen. Olwal and Feiner [17] provided a comprehensive study on this type of interaction using stylus and phone.

Second Screen

Here we describe the use of off-screen interactions on the secondary display space. (Figure3(d)) The handheld screen device can be used to link or store digital elements that can then be controlled or modified off-screen or from a remote location. This category aligns with Pick-and-Drop styled interactions introduced by Rekimoto [21].

Clipboard: The handheld device can be used as a mobile clipboard to collect digital elements (images, text, etc.). The handheld device can then be brought to a larger screen where the stored elements can just be swiped out of the handheld device onto the larger screen.

Palette: We imagine the handheld device to be an adaptive tool with which we can alter digital data as a painter arranges and mixes different colors on a real palette. Through making use of the handheld device's various sensors (e.g. touchscreen, accelerometer) the data can be modified. This subcategory describes a direct alteration of the data on the handheld device, in contrast to a palette example introduced by Piazza et al [19] where a phone is used as a separate controller for modifying brush strokes.

Pop-up: The phone can be linked to an element on the screen and will then constantly give updates about that element.

MIXED INTERACTION

Used in combination or sequentially, the interactions introduced in the previous section can construct more unique convolutions of tangible interaction and see-through augmentation (Figure 4). The near-surface configuration of our system allows the formerly disparate interaction styles of TUI and AR to be mixed.

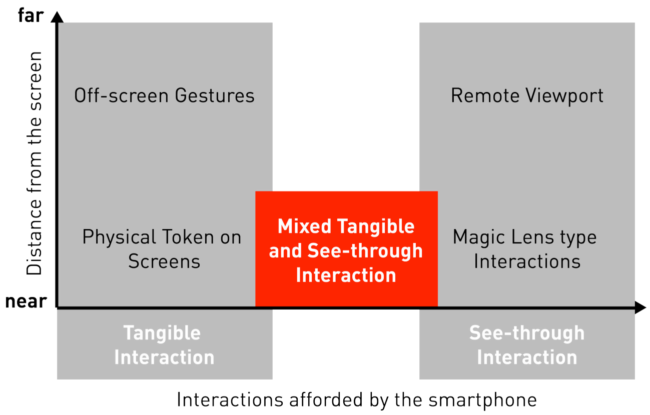

Figure 4. Near-surface interaction space allows unique combination of tangible and see-through interaction.

Magic Lens + Container: Figure 5(a) shows an example of using '*see-through*' and '*container*' interactions in combination. This combination allows selective manipulation of active foreground information (the game character) while displaying the static background information (the gray wall) through the smartphone screen. This spatially and contextually separates controllable entities from unnecessary ones. In addition to this, touch events can be used to move data across the phone (in the front) and the computer screen (in the back) (Figure 10).

Figure 5. Mixed interaction techniques used in a gaming scenario. (a) Simultaneously displaying foreground (character) and background (wall) (b) Using gestures to change the size of the character jumped into the phone (c) Changing behavior of an object by adding virtual friction

Second Screen + Container: Interactions on the phone screen allows manipulation of the data contained in it and feeding it back to the larger screen. As a result, the phone turns into an always-accessible magic box that lets users transform digital objects in a seamless manner. Figure 5(b) illustrates how using gestures on a smartphone changes the size and behavior of a game character that jumped "into" it. A more practical use of this is to apply the same adjustments to multiple files selectively by simply dragging them into the phone.

Magic Lens + Pass-through Filter: The secondary screen can be used as a moveable tangible filter that alters properties of digital elements behind it. However, it is difficult to know the filtering process without being able to see through the filter's body. In this case, see-through augmentation will greatly improve the visual cues that provide direct visual feedback to the users. Figure 5(c) shows an example of changing the speed of a moving object on the computer screen, where motion friction being applied by the phone is displayed via see-through augmentation.

IMPLEMENTATION

The very close distance from the smartphone to the screen (< 2cm) makes conventional feature-based tracking impossible due to the camera's lack of near-focusing capability and limited field of view (FOV). In our system, a computer screen displays a distinct color pattern. The phone's back-facing camera detects the pixels' color shown on the screen behind the phone. Sampled points are used to infer the phone's position from the RGB values through linear transformation. Constantly displaying the color pattern to enable tracking limits the aesthetics and readability of many applications. To minimize this interference we use a masked pattern that only shows in the camera's field of view. Its position and size are updated based on the previous frame's tracking result. (Figure 6)

For the implementation we used the iPhone 5S and 4S on a 15-inch Retina Macbook Pro as well as a 50-inch Hisense 1080p LED TV. The software is built upon openFrameworks. On the host computer, a server listens for incoming connections from the smartphone. Once a connection is established the devices exchange the calibration and tracking information. For data communication between the devices, we used Open Sound Control (OSC), an UDP-based communication library.

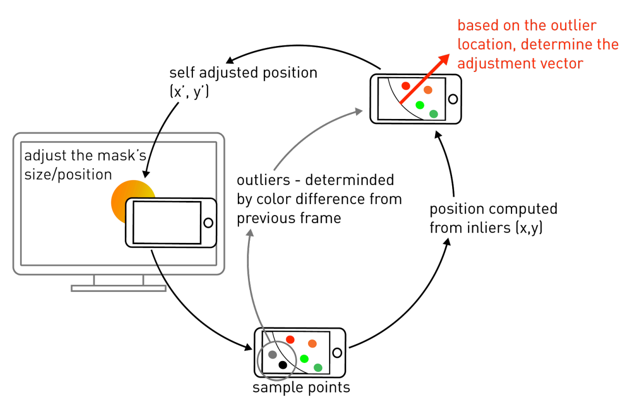

Figure 6. Sequence diagram of the tracking algorithm

Position

On capacitive touchscreens, the position of a phone can easily be tracked by attaching copper tape to the bottom of the phone to create capacitance between the two devices. Here, we focused on computer screens without capacitive touch capabilities to create a system that could potentially be applicable to any available display. We propose a method for tracking the 2D position of a smartphone relative to a computer screen using a distinct color pattern displayed on the host screen (top left is black (0, 0, 0), top right is red (255, 0, 0), bottom left is green (0, 255, 0) and bottom right is yellow (255, 255, 0)). Using the back-facing camera of the phone, we can detect the color of the pixels that appear on the screen behind the phone. Four corner points are sampled from the captured image. If there is an abrupt change in color value, the sample is regarded as an erroneous measurement and thus filtered out.

From the captured RGB values the corresponding 2D coordinate can be calculated. The mathematical relationship between RGB color and 2D coordinate can be described as:

$$\boldsymbol{x} = A\boldsymbol{c}$$

with $\boldsymbol{x} = (x, y)$ and $\boldsymbol{c} = (r, g, b)$. Due to white balance and the dynamic range of camera and screen the mapping might deviate from a linear function. However, experimentally, we found that linear mapping gives a reliable approximation. The coefficients are determined through Linear Least Squares using known measurement points $\boldsymbol{x}$ and detected color $\boldsymbol{c}$,

$$\hat{A} = \underset{A}{\operatorname{argmin}} \|\boldsymbol{x} - A\boldsymbol{c}\|$$

Since each computer display has fixed color characteristics, the calibration needs to be performed only once per device.

Orientation

The phone's orientation can be determined using the four sample points in the captured image. However, on a vertical host screen we get more precise tracking results using the phone's accelerometer and gyroscope. If the host screen is in a horizontal position, as most tabletop interfaces, we can determine the phone's orientation using its compass in combination with the color pattern.

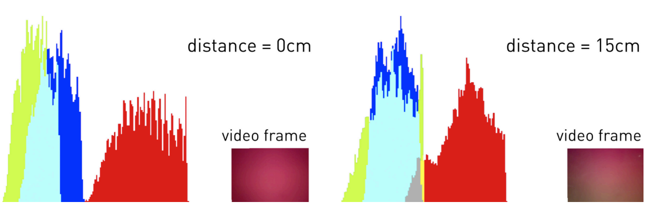

Figure 7. Proximity can be obtained via color analysis

Proximity

The distance between the computer screen and the phone can be inferred from the color delta of the video frame captured by the camera. As the computer's background is a linear color gradient, viewing it from further distance will show a larger color delta. The corresponding histogram can be seen in Figure 7. Through this we are able to get considerably reliable tracking results for up to 15 cm.

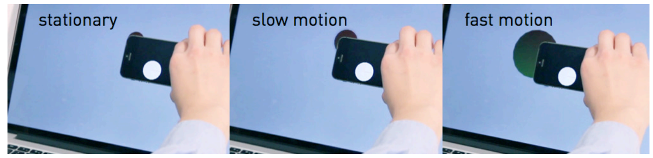

Figure 8. The masked pattern changes its size proportionally to the phone's velocity.

Updating Mask Position

To achieve continuous tracking it is sufficient for the color pattern to only be visible in the camera's field of view. Therefore we can hide the pattern in the area occluded by the phone. This allows tracking on the screen without sacrificing valuable information space.

To compensate for fast movements and the camera's relatively slow frame rate (30 fps), we constantly adjust the size of the pattern to provide a larger tracking area when the phone moves faster. (Figure 8) The initial position of the mask is found by increasing its size until the camera picks up the tracking pattern.

This masking technique not only hides the pattern behind the phone, but it also affords auto-calibration. A mismatch in color characteristics causes a spatial mismatch of the camera's location and the masked pattern. Once the camera picks up the pattern's boundary, the pattern's position is adjusted to again fit the cameras field of view. This offset could not be detected without the masked pattern, making this technique more robust.

Figure 9. Limiting the color spectrum minimizes the invasiveness of the tracking pattern.

Limited Color Spectrum

An additional method to make the tracking less invasive is to limit the color spectrum used. (Figure 9) If the display's dynamic range is high enough for fine color representation, we can choose a small range for R and G values to create the pattern (e.g.220 - 255). Limiting the color spectrum makes the pattern nearly invisible to human eyes. By adjusting the camera's exposure and white balance it is possible to get a good dynamic range within this color spectrum. However, this technique is not as reliable and requires a higher calibration effort. As cameras improve this might become a feasible technique in the future.

EXAMPLE APPLICATIONS

To showcase our system, we developed a series of example applications. Each application incorporates different interaction modes that were described earlier in this paper.

See-through Mouse

We use the phone as an advanced see-through mouse tool. This enables us to touch and drag/drop multiple files at once. At the same time the phone serves as a tangible clipboard for easy copy and paste of digital content. By placing a file inside the phone and removing it from the screen the data gets transferred to the phone. This makes it extremely simple to transfer files between two devices. This see-through mouse extends the modality of a conventional mouse as it can contain and visualize files that are being dragged. Users can select, drag/drop or delete files by direct touch gestures (Figure 10). The user can also place the phone on top of any link on the computer's screen and open it on the phone through a simple tap (Figure 11).

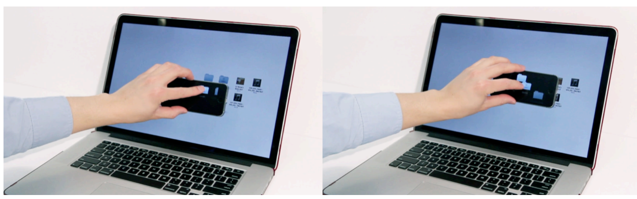

Figure 10. Seeing and touching through the smartphone. This enables easy file transfer between devices.

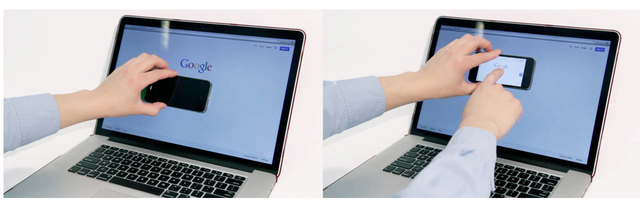

Figure 11. Opening a hyperlink in the smartphone

Adaptive UI Tool

Here we present an image editing application (Figure 12). The graphical UI displayed on the phone is aware of its location and context on the screen and can adjust accordingly. We use it to display functionalities only relevant to the task. Additionally, we incorporated motion gestures to manipulate image properties. Rotating the phone mimics a real lens focus ring to blur an image. Shaking the phone adds grain to the image just as if you were using a salt & pepper shaker. (Figure 12 right)

Figure 12. Images can be adjusted by using the phone's position, rotation and gesture recognition.

Navigating Maps and Buildings

To explore multiuser interaction as well as tabletop scenarios we developed a geo navigation application. On a shared tabletop map, different users can place their phone on the area of their interest to obtain different information, e.g. satellite view, traffic information, etc. In addition to revealing information through this magic lens, users can access different layers of information by changing the phone's distance to the computer screen. In this example lifting the phone can access different floors of a building. In its current state all phones are registered on the tabletop screen before starting the app. (Figure 13)

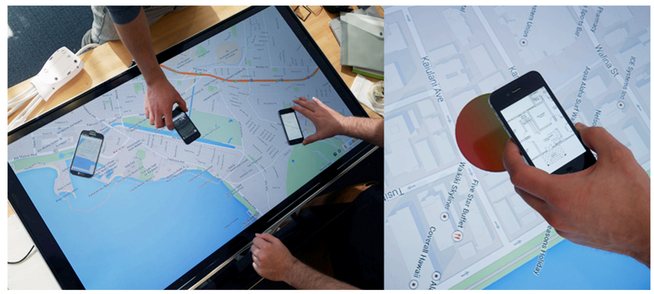

Figure 13. Multiple users can reveal additional information using their phone in this map application.

Game

We developed a simple game in which the goal is to help the character reach the flag. The smartphone acts as an active controller that can be used to physically intervene in the gameplay (Figure 1, 5). Players have to choose different strategies to clear the current stage. Each stage is designed to showcase a specific interaction or a mix of interactions we described in our classification. By combining the display space of a handheld device and a larger screen through near-surface interaction we can explore countless novel gaming scenarios.

SYSTEM EVALUATION

Tracking Accuracy

We tested the tracking accuracy by comparing the sample 2D coordinates on the computer screen to the measured coordinates from the phone. In the test, the distance between the phone and the computer screen was set to zero. We examined and compared the cases of fullscreen pattern and masked pattern. We found that the masked pattern provides better tracking accuracy because of its dynamically self-adjusting calibration. Figure 14(a) shows the result of the fullscreen pattern tracking, and (b) shows the masked pattern. Average deviations (distance of the measure point from ground truth) were 33.67 pixels (7.73mm) and 27.59 (6.33mm) pixels respectively. The maximum deviations were 98.25 pixels (22.55mm) and 56.44 pixels (12.95mm) for each. The screen's resolution was 1440x900 at 110 ppi.

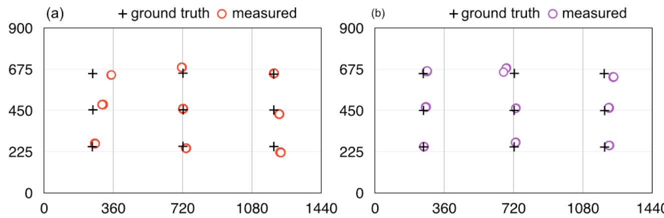

Figure 14. Tracking Accuracy. Left: without masked pattern, Right: with masked pattern.

Display's Dynamic Range vs. Tracking Resolution

As our tracking method relies on the color variation, the dynamic range of the computer display is critical for the quality of tracking. Here we examined the relationship between the display's dynamic range and obtainable tracking resolution on our 15-inch Retina display. Figure 15 shows the available color range (R-value) and the color captured by the camera from different x locations when changing the display's brightness from 100% to 25%. We found that 75% brightness provided the best range for tracking. Brightness of 100% and 87.5% will give equally good ranges of color. However, the captured color gets saturated, leading to clipping at the edges, thus limiting the effective tracking range.

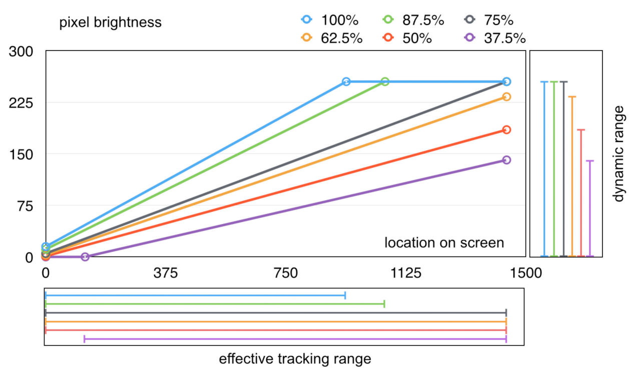

Figure 15. Screen brightness determines the effective tracking range.

Lost Tracking Behavior

To evaluate the reliability of our tracking system, we had to analyze the main factors for failure and the behavior of the system in such situation. We found that fast acceleration of the phone is the main reason for failed tracking. This is being addressed by rapidly increasing the masked pattern size, leading to a fast recovery of the phone's position. The general behavior of the masked pattern size is illustrated in Figure 16. After ~150 repetitions the average relocation time was 235ms, with a maximum of 737ms.

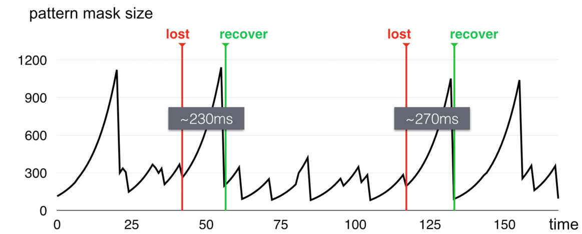

Figure 16. Rapid increase of the masked pattern size leads to fast tracking recovery.

LIMITATION

The proposed system has some limitations remaining to be solved. The use of color gradients for tracking promises easy deployment, while it still interferes with displayed content. Partial solutions of masked pattern/limited spectrum pattern are presented in this paper, however, the system needs very high dynamic range screens and cameras for reliable tracking. Another limitation is the latency caused by the relatively low framerate of the smartphone's camera (30 fps in general). To get tracking with almost no latency the system need to match the refresh rate of other human interface devices like optical mice. These are potentially resolvable by utilizing light spectrum invisible to human eyes. For example, if a distinct 2D infrared pattern is displayed on a monitor with dual infrared/visible light channel, a low resolution, high framerate infrared camera can reliably track its position much faster and more reliably.

CONCLUSION & FUTURE WORK

In this paper, we proposed an easy to deploy technology as well as interaction scenarios to better utilize the near surface space of computer screens with handheld smart devices. We show that the combination of AR and TUI enables versatile user interfaces for context aware seamless interactions. A growing number of people own computers and mobile phones, and with no need for additional hardware, the THAW system could work with those existing devices immediately. We believe this that this will open up a new space and tools for interaction designers to create fluid experiences using multiple personal devices.

A necessary step towards ubiquitous deployment of our system is to enable an effortless connection of personal handheld computing devices to any screen. One incremental step would be to develop a data protocol that allows easy and fast communication of relevant specifications between different devices. Another step forward is to expand this concept to devices with different display sizes and types (large, small, wearable, transparent, or flexible). Furthermore, a ground and more coherent taxonomy of the interactions that encompasses all different configurations would potentially benefit future multi-device applications.

ACKNOWLEDGMENTS

The authors would like to thank Professor Ramesh Raskar and everyone from 2013 Computational Camera class for valuable feedbacks during the conception of the project. We also appreciate Professor Ken Perlin and Alex Olwal for advises and inspiring inputs on the project.

REFERENCES

1. http://sebastianschmieg.com/royblock
2. http://www.nintendo.com/wiiu
3. Ballagas, R., Rohs, M., Sheridan, J., Borchers, J. The Smart Phone: A Ubiquitous Input Device. In IEEE Pervasive Computing, 5 (1), 2006.
4. Baur, D., Boring, S. and Feiner, S. Virtual Projection: exploring optical projection as a metaphor for multi-device interaction. In Proc. CHI ’12.
5. Bier, E. A., Stone, M. C., Pier, K., Buxton, W., and DeRose, T. D.. Toolglass and Magic Lenses: The See-Through Interface. In Proc. SIGGRAPH ‘93.
6. Boring, S., Baur, D., Butz, A., Gustafson, S., and Baudisch, P. Touch projector: mobile interaction through video. In Proc. CHI ’10, 2287–2296.
7. Chan, L., Muller, S., Roudaut, A., and Baudisch, P. CapStones and ZebraWidgets: sensing stacks of building blocks, dials and sliders on capacitive touch screens. In Proc. CHI ’12, 2189–2192.
8. Chan, L.-W., Wu, H.-T., Kao, H.-S., Ko, J.-C., Lin, H.-R., Chen, M. Y., Hsu, J., and Hung, Y.-P. Enabling beyond-surface interactions for interactive surface with an invisible projection. In Proc. ACM UIST ’10, 263–272.
9. Cuypers, T., Francken, Y., Vanaken, C., Reeth, F. V., and Bekaert, P.. Smartphone localization on interactive surfaces using the built-in camera. In Proc. Procams, pages 61–68, 2009.
10. Feiner, S., MacIntyre, B., Haupt, M., Solomon, M Windows on the world: 2D windows for 3D augmented reality. ACM UIST. p. 145-155.
11. Fitzmaurice, G., Buxton, W. The Chameleon: spatially aware palmtop computers. In Proc. ACM CHI ’94.
12. Hansen, T., Eriksson, E., Lykke-Olesen, A. Mixed Interaction Space - Designing for Camera Based Interaction with Mobile Devices. In Extended Proc. CHI ’05.
13. Heun, V., Kasahara, S., and Maes, P. Smarter Objects: Using AR Technology to Program Physical Objects and Their Interactions. In Extended Proc. CHI ‘13, 961–966.
14. Ishii, H. and Ullmer, B. Tangible Bits. Towards seamless interfaces between people, bits and atoms. In Proc. CHI ’97, 234–241
15. Liang, R.-H., Cheng, K.-Y., Chan, L., Peng, C.-X., Chen, M. Y., Liang, R.-H., Yang, D.-N., and Chen, B.-Y. GaussBits: magnetic tangible bits for portable and occlusion-free near-surface tangible interactions. In Proc. CHI ’13, 1391–1400.
16. Olwal, A. LightSense: Enabling spatially aware handheld interaction devices. Proc. ISMAR ’06, 119–122.
17. Olwal, A, and Feiner, S. Spatially aware handhelds for high-precision tangible interaction with large displays. In Proc. TEI ’09, 181–188.
18. Patten, J., Ishii, H., Hines, J., and Pangaro, G. Sensetable: a wireless object tracking platform for tangible user interfaces. In Proc. CHI '01, 253-260.
19. Piazza, T., Fjeld, M., Ramos, G., Evren Yantac, A., and Zhao, S. Holy Smartphones and Tablets, Batman! Mobile Interaction’s Dynamic Duo. In Proc. APCHI ’13, 10.
20. Reilly, D., Welsman-Dinelle, M., Bate, C., and Inkpen, K. Just point and click?: using handhelds to interact with paper maps. In Proc. MobileHCI '05, 239-242.
21. Rekimoto, J. A multiple device approach for supporting whiteboard-based interactions. In Proc. CHI ’98, 344–351.
22. Rekimoto, J. The magnifying glass approach to augmented reality systems. In International Conference on Artificial Reality and Tele-Existence / Conference on Virtual Reality Software and Technology (ICAT/VRST), 123–132, 1995.
23. Rohs, M., Schöning, J., Raubal, M., Essl, Georg and Krüger, A. Map Navigation with Mobile Devices: Virtual versus Physical Movement with and without Visual Context. In Proc. ICMI ‘07, 146-153.
24. Sanneblad, J. and Holmquist, L.E. Ubiquitous Graphics: Combining Hand-held and Wall-size Displays to Interact with Large Images. ACM AVI Conference on Advanced Visual Interfaces 2006.
25. Spindler, M., Martsch, M., and Dachselt, R. Going Beyond the Surface: Studying Multi-Layer Interaction Above the Tabletop. In Proc. CHI ’12, 1277–1286.
26. Tani, M., Yamaashi, K., Tanikoshi, K., Futakawa,M., and Tanifuji, S. Object oriented video: interaction with real-world objects through live video. In Proc. CHI ‘92, 593-598.
27. Wilson, A.D., Sarin, R. BlueTable: connecting wireless mobile devices on interactive surfaces using vision-based handshaking. In GI ‘07, 119-125.
28. Underkoffler, J. and Ishii, H. Urp: a luminous-tangible workbench for urban planning and design. In Proc. CHI ‘99, 386-393.