DisplayPointers: Seamless Cross-Device Interactions

Paul Strohmeier
paul.strohmeier@gmail.com
Human Media Lab
Queen's University, Kingston, ON, Canada
Department of Computer Science,
University of Copenhagen, Denmark

ABSTRACT
We present a system for cross-device interactions and interaction scenarios based on touch events between devices. DisplayPointers were designed to explore the affordances of manipulating physical display-objects in multi-device environments. The interactions presented in this paper are an exploration of what a touch between two devices means; a search for a language of physical cross-device interactions. DisplayPointers are implemented by augmenting off-the-shelf devices: our system is fully mobile and can be easily implemented by other researchers and designers.

Author Keywords
Multi-device Interaction; Tangible Magic Lens; Cross-device Interaction

ACM Classification Keywords
H.5.2 [User Interfaces]: Input Devices and Strategies, Prototyping

INTRODUCTION
As consumer display devices become more ubiquitous, interaction methods spanning multiple devices become increasingly more appealing. Current systems that enable interactions across multiple devices–such as Synergy, AirParrot, Dropbox and other VNC-based approaches–require the user to navigate complex menus to connect devices, spread content across their displays, or to send information between devices. Even within the HCI research community we find approaches that, in our opinion, require excessive multi-menu navigation [11] or series of gestures unrelated to the task of moving content among devices [23].

Using multiple devices however introduces new interaction modalities. Mobile devices can be manipulated in 3D space. This enables users to configure devices in ways that may offer a visual benefit for the task at hand. When physically manipulating devices, they provide users with haptic and kinesthetic feedback. When devices touch each other or are moved across a surface, they also provide subtle audio cues.

ACE '15, November 16 - 19, 2015, Iskandar, Malaysia

ACM 978-1-4503-3852-3/15/11$15.00
DOI: http://dx.doi.org/10.1145/2832932.2832958

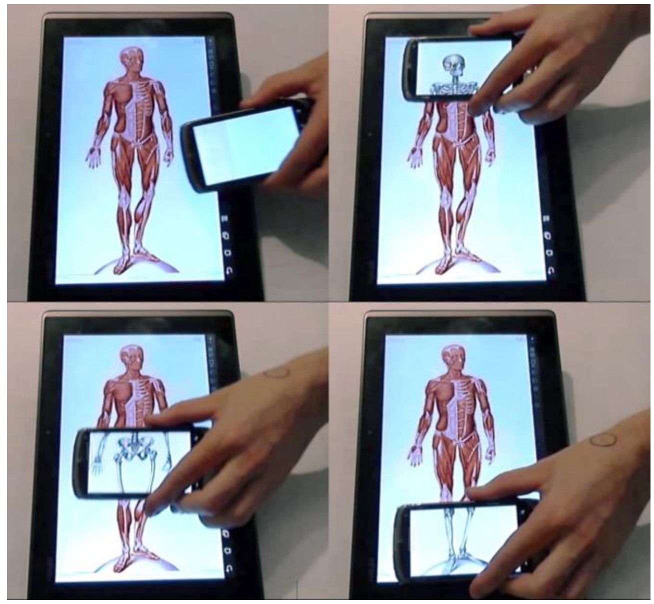

Figure 1- Magic Lens application using DisplayPointers

Physically interacting with multiple devices provides us with a rich, multi-modal, interaction space.

We believe that cross-device interactions should leverage this rich interaction space, allowing users to utilize the physical properties of objects instead of navigating a GUI or gestural interface. Examples of cross-device interactions which leverage the affordances of physical objects include products such as Bump which implements research by Hinckley [14] or resonant bits demonstrated by Bennet et al. [2].

In this paper we discuss how physical affordances of multiple mobile devices can replace activities which traditionally require explicit input or menu navigation. Additionally, we present methods that support the design of this type of cross-device interaction. Our system of DisplayPointers can be easily implemented on everyday devices, while enabling a large set of interactions that typically require controlled environments or extensive instrumentation. The interactions presented in this paper are intended to be an exploration of what a touch interaction between two devices means; a step towards a physical interaction language of physical cross-device interactions.

RELATED WORK

Multi Device Environments
As stated by Sellen and Harper [26], physical media have affordances that are difficult to achieve with digital

solutions. These include the haptic and kinesthetic experience of manipulating documents as well as the possibility to access and juxtapose multiple documents simultaneously, both for side by side reading and for concurrent reading and writing. Chen et al. have shown that multi-display environments are beneficial for active reading [5].

Various projects have focused on obtaining similar affordances by means of digital tools. Wellner's DigitalDesk [31] and PaperWindows by Holman et al. [15]used projection mapping to integrate paper with the digital world. PaperTab by Tarun et al. [27] expands on this interaction space by using physical displays and introducing spatial interaction methods. These include pointing with displays and other proximity based interactions.

Prototypes such as FoldMe [16] by Khalilbeigi et al. or Codex by Hinckley et al. [12] attempt to take advantage of these findings by providing users with multi-display devices in various configurations. PaperFold by Gomes et al [10] provides dynamic configuration of screen real estate.

While the rationale behind these prototypes is promising, form factors, as well as technological constraints, limit the type of casual, lightweight interactions [20] we are accustomed to from interacting with physical documents.

Position Aware Handheld Devices

In Marshall and Bly's study of casual reading [20], participants chose a couch, arm-chair or bed as a preferred place to read. Mobile devices might be able to provide some of the light-weight affordances traditional digital systems lack: their form factor allows physical manipulation, and we can use them wherever we go.

Early explorations which specifically explored the affordances of devices which can easily be moved in 3D space include position-aware handheld devices presented by Rekimoto [24] and Fitzmaurice [9]. These systems used instrumented environments combined with optical and inertial tracking, respectively.

Voida et al. [29] presented an iPodLoupe, which is linked to a tabletop display via gestural commands. They implemented interactions suggested by Bier et al. [3] as well as Ware and Lewis [30].

Two recent explorations of position-aware handheld devices are PhoneTouch by Schmidt et al. [25] and THAW by Leigh et al. [17]. They offer similar interactions to Voida, however they opted for a direct manipulation approach: Schmidt et al. [25] explored the use of mobile phones in a stylus-like fashion to interact with tabletop surfaces, while Leigh et al. [8] used the camera of a mobile device to detect on-screen visual patterns to track its relative position.

Even more recent projects, ACTUI [18] and VISTouch [33] have explored the relative orientation of devices while touching, creating 3D display configurations. Though these systems investigate interactions between touching devices, they do not discuss the affordances of a touch event between two devices, and how these affordances can be used for the design of interactions.

DESIGN RATIONALE

The motivation behind DisplayPointers was to create a multi-device system which minimizes barriers between mobile devices. Most technology users already own multiple devices. We would like to leverage this and create a solution that can be implemented on commodity hardware. We further envision a system that can be used in ad-hoc work spaces: in a waiting room, a train, the bedroom.

In terms of interaction, we wish to further explore what it means for a device to touch another device beyond simple data-transfer and authentication as presented in PhoneTouch, moving towards a true multi-device work environment. We believe that the multi-model experience of interacting with physical objects should be leveraged to avoid the use of menus or gestural commands. For example, the kinesthetic experience of making two devices touch provides a simple but powerful feedback mechanism that warrants exploration.

The interactions proposed in this paper should not replace interactions we are accustomed to. Instead the interactions should offer users additional ways of completing tasks. This should be achieved by keeping the input space of finger and phone touches identical. While touch interactions can be executed with fingers or with phones, executing them with phones may change the context in which they are executed.

APPARATUS

DisplayPointers enable seamless interactions between off-the-shelf display devices such as smart-phones, tablets or laptops. This is achieved by using smartphones as pointing devices on a larger mobile device, such as a tablet. Using a tangible display device as a pointer alongside regular touch input opens up the interaction space for an organic multi-device experience, enabling users to leverage the benefits of multi-modal feedback. Devices can move in and out of focus, can become vessels for intermittently presenting data off-screen, can become transparent tools providing alternative views, or can provide users with contextual information.

The approach described in this paper enables arbitrary device pairings. A tablet could be used as a pointer on a phone. A phone could point on another phone or a touch-enabled laptop, etc. For simplicity we will only use examples where the tablet is the device being touched and the smartphones are the devices used as pointers. This restriction is introduced for the sake of clarity, but is not in any way a restriction of the system proposed.

IMPLEMENTATION

Tracking using 'Hand Down' Capacitance

We use the capacitive screen of a tablet computer to track the position of the phones touching it. We do so by using 'hand down' capacitance. This method was used by Chan et al. [4] for detecting tangible objects on displays and is the principle behind capacitive styli. It was also utilized by VISTouch [33] and ACTUI [18]: A conductive object by itself is invisible to the capacitive screen. However, if it is touched by a human hand, the conductive object can then be tracked by the touch-screen. Many phones, for instance those designed by HTC and Apple, have aluminum casings that enable them to pass on human capacitance and register raw data of capacitive sensors. When using phones without aluminum casing or if one wishes to use the filtered capacitive touch output of most tablets and smartphones, a conductive marker can be attached to the device (as one can find on the tip of capacitive styli).

Simple ways of creating such a conductive marker include using a small piece of anti-static foam or covering a soft item (such as the eraser tip on the back of a pencil) with conductive fabric. These markers should then be connected with a conductive material such as conductive fabric, wire or copper tape, which encircles the bezel of the device. This ensures an electrical connection between the user's body and the touch point.

Differentiating Fingers and Phone

Because we wish to use the identical input space for finger and phones, we require a method of differentiating between them. We found it possible to distinguish between phones and fingers based on size and dynamics of the touch point. A rigid object or an object augmented with a conductive marker typically has a smaller touch-point than a finger. Additionally, the size of this touchpoint is more consistent than that of a finger (Figure 2).

Reliability

To provide a basic idea of how well touch events triggered by phones can be distinguished from regular touch, we conducted a preliminary evaluation. We asked 6 participants (4 male, 2 female) to execute a series of 4 touch gestures (tap, short swipe, medium swipe, long swipe) with both their finger and a phone. They performed this series of gestures 25 times for a total number of 600 touch events per input method. We found that the average touch size for the phone (M=0.0957, SD=0.0327) was significantly smaller ($t_{(846)}$= -25.7, p<0,001) than the touch size for the finger (M=0.1772, SD=0.0704) using a two sample t-test assuming unequal variance. With a high pass filter as a measure of touch-dynamics we found that the dynamics for the phone (in ‰ of the touch-size, M=0.0064, SD=0.0222) were significantly lower ($t_{(778)}$=-9.93, p<0,001) than those of a finger (M=0.0311, SD=0.0566). Using a simple cut-off value of 0.1407 for detecting phones based on maximum touch-size, we were able to correctly identify 96% of all touch events. Using a decision tree approach including the touch-dynamics we were able to correctly identify 98% of all touch events.

Identifying Multiple Phones

To fully utilize a multi-device environment, it is not sufficient to only detect that there is a touch event triggered by a phone, we also need to establish which phone the touch event is triggered by. We find that a reliable method of doing this is matching the touch information with the accelerometer readings of the phones. Like PhoneTouch [25] we use the accelerometer to detect contact between two devices. Additionally we match the acceleration of the device to the acceleration of the touch point (Figure 2).

Figure 2 - Differentiating between finger and two individual phones.

Tracking Orientation

Most mobile devices have a multitude of integrated sensors. Smartphones commonly have an accelerometer, gyroscope and magnetometer integrated. Using sensor fusion algorithms [22] [19], the data from these sensors can be used to calculate a heading for each device. This allows us to calculate the relative angle and, in consequence, the orientation between a phone and the tablet.

Frame of Reference

To calculate device orientation or to match acceleration data with touch data, devices require a common frame of reference. In our system we used the inertial information of the tablet to establish our frame of reference. When matching acceleration to touch information, we subtract the acceleration of the tablet from the acceleration of the phone before doing the comparison. Using the tablet as a common frame of reference is especially important when both devices are in motion.

Communication

All touches are detected and identified by the tablet. The tablet sends all touch information to the phones using Open Sound Control (OSC, [32]) The applications on the phones accept the touch information. The behavior of the phone in reaction to that touch information is determined by the active app on the phone.

INTERACTION METHODS

Our system can be used to implement various interaction methods suggest by Bier, Leigh, Schmidt, Ware and Lewis [3,17,25,30] and others. In this section we wish to highlight interaction concepts which previous work has not explicitly addressed.

Interactions based on Input Method

In a multi-display environment, the user requires a clear understanding of which device will respond to their input. When opening an application, ideally it should open on the device the user is currently focused on, and not on a secondary display which they are currently not attending to. Various solutions to this issue have been suggested, ranging from eye-contact based solutions [28], to heavily instrumented and assumption based environments as presented with PaperTab [27].

We use a relatively simple approach, based on distinguishing between input from either finger or phone. We can design a single UI item, which acts differently depending on how it is accessed. A simple icon used to open an application might have two different behaviors: If it is touched by a finger, the application launches on the tablet the icon was displayed on. If the same icon is touched by a phone, the application launches on the phone (Figure 3). This method of displaying information provides a clear mapping between input and output. It is also an intuitive method for the user to display information on a device of their choice: as the mapping is clear from the physical configuration of the devices, the user is not required to navigate menu items or use any particular gesture to select where to display the information.

Figure 3 - Two different ways of opening an e-mail application. In the center image it is opened using a phone in the right hand image using a finger.

Devices as special purpose tools

When presented with an e-mail, a user may want to react in different ways. They might wish to take action immediately by reading it or by deleting it. Alternatively they may want to put it aside for later, or they might want to access supplementary information such as the sender's full name or the time it was sent.

The user could open or delete the e-mail with touch gestures, or they could use a phone to open the e-mail and put it aside in order to read it later. A second phone could be running a context application. If the user selects the e-mail with the 'context phone', instead of displaying the content of the e-mail, it will display contextual information about it. Individual phones can therefore become special purpose tools, based on how they are configured (an example of using a 'Wikipedia tool' to explore a map can be seen in Figure 5).

These special purpose tools can be physical manifestations of linked views, a concept introduced by Ware and Lewis [30] and further explored by North and Schneiderman [21].

Linking Devices

In a multi-device environment, an action on one device might have consequences to information displayed on other devices. While such linked views allow for intuitive exploration of data [30], it is also important to unlink data, so a view one wishes to preserve for later reference becomes permanent. In order to provide the user with consistent behavior, which allows both dynamic browsing as well as physical sorting and arranging, we suggest a simple rule on linking and decoupling content.

If a user interacts with the touch-screen of a tablet while a phone is touching it, the touch interaction affects both the tablet and the phone. If the phone is removed from the tablet, the touch interaction no longer affects both devices.

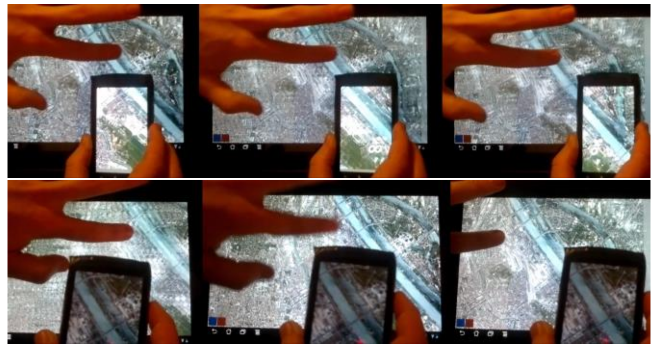

Figure 4 – Top: The phone is touching the tablet. Because the two are coupled the pan gesture is executed on both devices. Bottom: The phone is no longer touching the tablet. The pan gesture is executed on the tablet, while the phone remains static.

In a map application, for example, a user might place a second phone on the tablet. This phone could be used as a lens that highlights certain features of the map. The user can use touch gestures to pan and zoom the map to their area of interest. These panning and zoom actions occur simultaneously on the phone. Once the phone is removed from the tablet, the content of the phone becomes permanent and is no longer affected by further touch input on the tablet (Figure 4).

In digital systems representing and interacting with such links requires complex and nuanced software visualization approaches [7]. While these software implementations offer affordances and opportunities of their own, DisplayPointers do not require such considerations, as the linkage is clear based on the physical configuration of the devices.

Multiple Physical Clipboards

Research has shown that epistemic actions [8] such as clustering physical items [1], exploring, testing and restructuring [6] are invaluable when engaging with a task [8]. Traditionally, digital systems do not support these actions well. For example, the lack of tactile-kinesthetic feedback and the ability to spatially arrange documents in digital systems, negatively influences our reading behavior [26]. A multi-device environment allows users to regain some of these tactile-kinesthetic sensations and supports epistemic actions by physically arranging information.

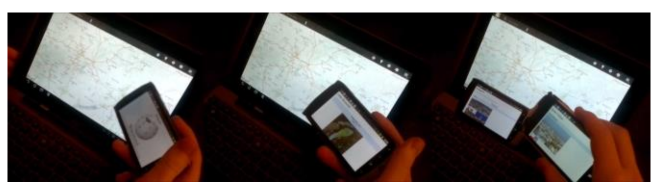

Figure 5 – A 'context' application shows Wikipedia articles which are linked to locations on a map. The user can use multiple phones to collect and arrange information.

Phones can be used as a physical clipboard to temporarily store information. The obvious advantage of using such a physical clipboard is that, unlike the more familiar copy-paste software, users have visual access to the content of the clipboard. Furthermore, users can use multiple clipboards simultaneously. This allows users to physically manipulate information and arrange it in physical space, which has been shown to support active reading (Figure 5) [5].

Relative Orientation

VISTouch [33] and ACTUI [18] have expanded interactions into a third dimension, using the relative orientation of two devices to manipulate their respective views. Relative orientation between devise can however also be used to manipulate parameters of the device as a special purpose tool. This is practical, as, when user is holding two devices that are linked by touch, actions such as pinch-zoom are difficult to execute with both hands occupied.

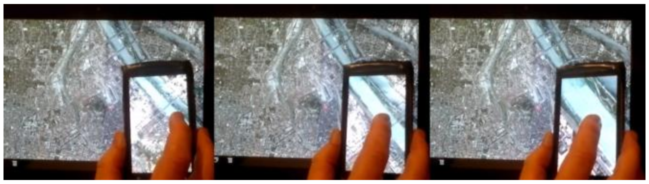

Figure 4 - Using a Touch & Tilt gesture for selecting zoom level.

As an alternative to touch gestures, the relative orientation between a phone and a tablet when touching can be used to modulate the relationship between them. The tilt of a phone can be used to modulate the mapping of the actions on the tablet to the actions on the phone. For example, touching the phones display, while simultaneously tilting the phone, can be used as single handed gesture for changing zoom levels, when using the phone as a lens for exploring content on the tablet (Figure 6). This type of input could be further extended by implementing multi-modal interactions suggested by Hinckley and Song [13].

LIMITATIONS & FUTURE WORK

DisplayPointers are limited by only supporting interactions with objects in physical contact. A next step to investigate is moving the interaction in the third dimension by using a contactless position tracking method. While we find THAW's [17] camera based system to be a step in the right direction, we believe that future work should additionally explore the use of mobile devices' inertial sensors for relative position tracking in 3D space.

We present anecdotes of how the affordances of touch interactions between physical devices can be utilized, however, a systematic exploration as well as evaluation of these interaction methods is warranted.

CONTRIBUTION

With DisplayPointers we propose interaction techniques to simplify cross-device multi-screen interaction. We leverage the physicality of devices to minimize interactions with menus or gestural commands. We focus on touch between two devices: the physical contact provides kinesthetic feedback that supports activities such as linking devices. We use touch together with relative position to tangibly copy selected content or open applications between devices, requiring no intermediate conceptual steps. We use relative orientation for fine-grained bimanual continuous input. We do not need to implement specific feedback systems, as multi-modal feedback is already provided by the physicality of the devices. While doing so, we do not interfere with interaction methods we are already accustomed to: If GUI elements are touched with a finger, the device behaves as users would expect it to.

DisplayPointers require minimal instrumentation. This is a significant contribution over previous projects that explored similar multi-device interactions: PaperTab [27], PaperWindows [15], DigitalDesk [31], etc. all require significant instrumentation and controlled environments.

DisplayPointers are mobile, not tied to a specific environment or location. This is a contribution over PhoneTouch [3], which is not designed to be mobile, and also over THAW [17], which is to some extent dependent on environmental factors due to its vision based system.

Finally, DisplayPointers are implemented using various off-the-shelf devices: tablets, laptops, smartphones etc. This makes it a convenient and accessible approach to prototyping and sketching ideas.

REFERENCES

1. Alissa Antle and Sijie Wang. 2013. Comparing motor-cognitive strategies for spatial problem solving with tangible and multi-touch interfaces. *Proceedings of the 7th International Conference on tangible, embedded and embodied interaction*: 65–72. http://doi.org/10.1145/2460625.2460635
2. Peter Bennett, Stuart Nolan, Ved Uttamchandani, Michael Pages, Kirsten Cater, and Mike Fraser. 2015. Resonant Bits. *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction - TEI '14*, ACM Press, 49–52. http://doi.org/10.1145/2677199.2680569
3. E.A. Bier, M.C. Stone, Ken Pier, William Buxton, and T.D. DeRose. 1993. Toolglass and magic lenses: the see-through interface. *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, ACM, 73–80. http://doi.org/10.1145/166117.166126
4. Liwei Chan, Stefanie Müller, Anne Roudaut, and Patrick Baudisch. 2012. CapStones and ZebraWidgets. *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI '12*, 2189. http://doi.org/10.1145/2207676.2208371
5. Nicholas Chen, François Guimbretière, and Abigail Sellen. 2013. Graduate student use of a multi-slate reading system. *Proc CHI*, ACM Press, 1799. http://doi.org/10.1145/2470654.2466237
6. A Clark. 1998. *Being there: putting brain, body and world back together again*. MIT Press.
7. Christopher Collins and Sheelagh Carpendale. 2007. VisLink: Revealing relationships amongst

visualizations. *IEEE Transactions on Visualization and Computer Graphics* 13, 6: 1192–1199. http://doi.org/10.1109/TVCG.2007.70521

8. Augusto Esteves, Saskia Bakker, Alissa N Antle, Aaron May, Jillian Warren, and Ian Oakley. 2015. The ATB Framework : Quantifying and Classifying Epistemic Strategies in Tangible Problem-Solving Tasks. 13–20. http://doi.org/10.1145/2677199.2680546
9. George W. Fitzmaurice. 1993. Situated information spaces and spatially aware palmtop computers. *Communications of the ACM* 36, 7: 39–49. http://doi.org/10.1145/159544.159566
10. Antonio Gomes and Roel Vertegaal. 2014. Paperfold. *Proceedings of the extended abstracts of the 32nd annual ACM conference on Human factors in computing systems - CHI EA '14*, 535–538. http://doi.org/10.1145/2559206.2574770
11. Peter Hamilton and Daniel J. Wigdor. 2014. Conductor. *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14*, ACM Press, 2773–2782. http://doi.org/10.1145/2556288.2557170
12. Ken Hinckley, Morgan Dixon, Raman Sarin, Francois Guimbretiere, and Ravin Balakrishnan. 2009. Codex : A Dual Screen Tablet Computer. *Design*: 1933–1942. http://doi.org/10.1145/1518701.1518996
13. Ken Hinckley and Hyunyoung Song. 2011. Sensor Synaesthesia: Touch in Motion, and Motion in Touch. *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11*, ACM Press, 801. http://doi.org/10.1145/1978942.1979059
14. Ken Hinckley. 2003. Synchronous gestures for multiple persons and computers. *Proceedings of the 16th annual ACM symposium on User interface software and technology - UIST '03*, ACM Press, 149–158. http://doi.org/10.1145/964696.964713
15. David Holman, Roel Vertegaal, Mark Altosaar, Nikolaus Troje, and Derek Johns. 2005. Paper Windows. *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '05*, ACM Press, 591. http://doi.org/10.1145/1054972.1055054
16. Mohammadreza Khalilbeigi, Roman Lissermann, Wolfgang Kleine, and Jürgen Steimle. 2012. FoldMe. *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction - TEI '12*, ACM Press, 33. http://doi.org/10.1145/2148131.2148142
17. Sang-won Leigh. 2015. THAW : Tangible Interaction with See-Through Augmentation for Smartphones on Computer Screens. *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction - TEI '14*: 89–96. http://doi.org/10.1145/2677199.2680584
18. Ming Li and Leif Kobbelt. 2015. ACTUI : Using Commodity Mobile Devices to Build Active Tangible User Interfaces. *MobileHCI*: 592–597. http://doi.org/10.1145/2786567.2792895
19. Robert Mahony, Tarek Hamel, and Jean Michel Pflimlin. 2008. Nonlinear complementary filters on the special orthogonal group. *IEEE Transactions on Automatic Control* 53, 5: 1203–1218. http://doi.org/10.1109/TAC.2008.923738
20. C.C. Marshall and Sara Bly. 2005. Turning the page on navigation. *Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries, June*, ACM Press, 07–11. http://doi.org/10.1145/1065385.1065438
21. Chris North and Ben Shneiderman. 1997. A Taxonomy of Multiple Window Coordinations. 97–18. http://doi.org/10.1.1.29.2424
22. William Premerlani and Paul Bizard. 2009. Direction Cosine Matrix IMU : Theory. Dcm: 1–30.
23. Roman Rädle, Hans-Christian Jetter, Nicolai Marquardt, Harald Reiterer, and Yvonne Rogers. 2014. HuddleLamp. *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces - ITS '14*, ACM Press, 45–54. http://doi.org/10.1145/2669485.2669500
24. Jun Rekimoto. 1995. The Magnifying Glass Approach to Augmented Reality Systems. *International Conference on Artificial Reality and Telexistence icat* 95, 4: 123–132. http://doi.org/10.1.1.50.5835
25. Dominik Schmidt, Fadi Chehimi, Enrico Rukzio, and Hans Gellersen. 2010. PhoneTouch: A technique for direct phone interaction on surfaces. *Proc. UIST*. http://doi.org/10.1145/1866029.1866034
26. Abigail J. Sellen and Richard H. R. Harper. 2002. *The Myth of The Paperless Office*. Massechussets Institute of Technology. Retrieved February 4, 2015 from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.91.6538&rep=rep1&type=pdf
27. Aneesh P. Tarun, Peng Wang, Audrey Girouard, Paul Strohmeier, Derek Reilly, and Roel Vertegaal. 2013. PaperTab. *CHI '13 Extended Abstracts on Human Factors in Computing Systems on - CHI EA '13*, 3131. http://doi.org/10.1145/2468356.2479628
28. R. Vertegaal and J. S. Shell. 2008. Attentive user interfaces: the surveillance and sousveillance of gaze-aware objects. *Social Science Information* 47, 3: 275–298. http://doi.org/10.1177/0539018408092574
29. Stephen Voida, Matthew Tobiasz, Julie Stromer, Petra Isenberg, and Sheelagh Carpendale. 2009. Getting practical with interactive tabletop displays.

Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces - ITS '09, ACM Press, 109. http://doi.org/10.1145/1731903.1731926

30. Colin Ware and Marlon Lewis. 1995. The DragMag image magnifier. *Conference companion on Human factors in computing systems - CHI '95*, ACM Press, 407–408. http://doi.org/10.1145/223355.223749

31. Pierre Wellner. 1993. Interacting with paper on the DigitalDesk. *Communications of the ACM* 36, 7: 87–96. http://doi.org/10.1145/159544.159630

32. Matthew Wright and Adrian Freed. 1997. Open Sound Control: A New Protocol for Communicating with Sound Synthesizers. *Proc. ICMC 1997*, International Computer Music Association, 101–104. Retrieved from http://cnmat.berkeley.edu/publication/open_sound_control_new_protocol_communicating_sound_synthesizers

33. Masasuke Yasumoto. 2015. VISTouch : Dynamic Three-Dimensional Connection between Multiple Mobile Devices. 89–92. Retrieved from http://doi.acm.org/10.1145/2735711.2735823