HeadPhones: Ad Hoc Mobile Multi-Display Environments through Head Tracking

Jens Grubert
Coburg University
jg@jensgrubert.de

Matthias Kranz
University of Passau
matthias.kranz@uni-passau.de

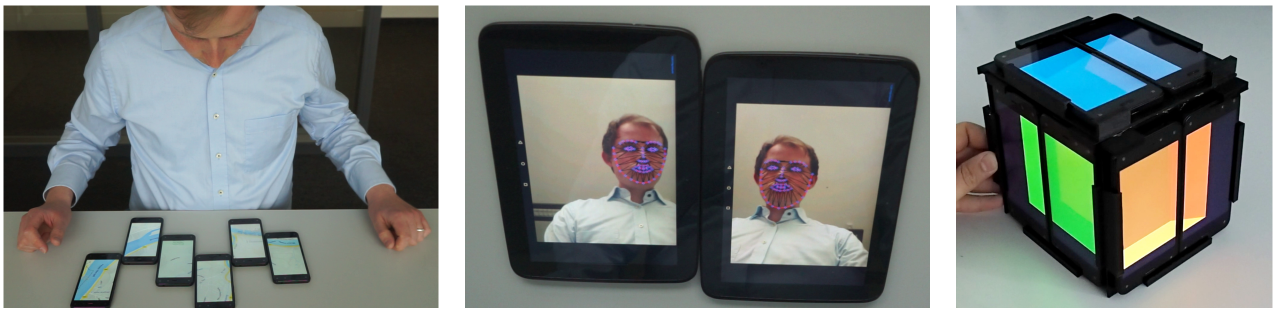

Figure 1: Left: HeadPhones allows dynamic registration between mobile devices using the human head as reference frame. Middle: Head pose estimation and concatenation is used to determine the relative positions of the devices to each other. Right: a perspective cube display viewer assembled from multiple smartphones.

ABSTRACT
We present HeadPhones (Headtracking + smartPhones), a novel approach for the spatial registration of multiple mobile devices into an ad hoc multi-display environment. We propose to employ the user's head as external reference frame for the registration of multiple mobile devices into a common coordinate system. Our approach allows for dynamic repositioning of devices during runtime without the need for external infrastructure such as separate cameras or fiducials. Specifically, our only requirements are local network connections and mobile devices with built-in front facing cameras. This way, HeadPhones enables spatially-aware multi-display applications in mobile contexts. A user study and accuracy evaluation indicate the feasibility of our approach.

ACM Classification Keywords
H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

INTRODUCTION
Large displays are compelling for many work tasks, but are typically non-mobile. In contrast, many mobile users own at least one smartphone, along with a tablet and potentially a smartwatch. When people meet, the number of concurrently available mobile screens easily reaches a dozen or more. Collectively, these individual small displays could be used to create large-scale displays in an ad hoc manner.

CHI 2017, May 06-11, 2017, Denver, CO, USA
ACM 978-1-4503- 4655-9/17/05.
http://dx.doi.org/10.1145/3025453.3025533

Still, creating these ad hoc multi-display environments is typically cumbersome. Besides initially *binding* or associating devices into a group (i.e. each device is aware of the existence of nearby devices), devices generally lack the knowledge about their *spatial position* relative to the devices around them. While a number approaches for ad hoc device binding already allows for easy mobile use [1], determining the spatial position of devices in an ad hoc mobile multi-device environment remains challenging [3]. So far, approaches for the *spatial registration* of multiple mobile devices have either required external infrastructure such as cameras [17, 13] or fiducials [9], required an instrumentation of the user [4], resulted in low spatial registration accuracy [6, 7] or allowed for only static device configurations (i.e. devices could not be freely moved after initial registration [5, 10]).

Our system, HeadPhones[1], instead, makes use of the user's head as external reference frame, allowing to overcome the limitations of previous approaches. The system solely assumes that the devices have a front facing camera, which is used for head tracking. To the best of our knowledge, HeadPhones is the first system for the spatial registration of mobile devices in an ad hoc multi-display environment, which does not rely on external infrastructure such as cameras, allows for dynamic repositioning of devices after initial registration and achieves comparable accuracy of existing camera-based and gesture-based approaches.

RELATED WORK
Binding and spatially registering multiple mobile devices into multi-display environments has been of interest to the research

[1] source code available under https://gitlab.com/mixedrealitylab/HeadPhones

community for a while. Chong and Gellersen present a recent overview of binding or device association techniques [1].

HeadPhones primarily builts on previous work in the domain of spatial registration in mobile multi-display environments. A popular approach for spatial registration is to use finger or pen gestures across display boundaries [5, 10]. However, those approaches typically do not allow for repositioning of individual devices after the initial registration step. IMU-based approaches are often employed for dynamic peephole navigation on mobile devices [11], but might be combined with gesture-based approaches to enable dynamic repositioning. Still, those approaches are prone to sensor drift. There are various camera-based approaches, e.g., [17, 13, 16]. However, they often require an external camera or depth-sensor to be available and, hence, might constrain the mobility of device registration. Radio-frequency-based (e.g., [6, 7]), sound-based (e.g., [12, 22]) or inertial sensor-based approaches (e.g., [19]) also allow for (partially) pose estimation of devices, but typically result in low spatial accuracy, limited number of degrees of freedom or impact the user experience through audible sounds.

The two closest approaches to our work were presented by Li and Kobbelt [9] and Dearman et al [2]. Li and Kobbelt propose to use a fiducial marker (similar to [16]) which is tracked using the front camera of mobile phones. However, their approach requires a fiducial to be printed and carried around and the fiducial either has to be held with one hand during dynamic repositioning or has to be mounted in the environment, potentially leading to a substantial setup effort. In contrast, we propose to use the user's head to enable spatial registration of multiple mobile devices, requiring no further setup costs. Dearman et al. allow the recovery of 2D orientation between devices when being pointed to a textured floor in a server-based image stitching approach that does not scale well with increasing number of devices. HeadPhones allows use on surfaces (like tables) and free space with six degrees of freedom (6 DoF). Our runtime does not increase with the number of devices as each device computes its pose locally and a server is only used to manage all pose entries.

DISPLAY REGISTRATION THROUGH HEAD TRACKING

The core idea of HeadPhones is to enable the spatial registration of multiple mobile devices with front cameras through head tracking (HeadPhones: Headtracking + smartPhones). After an initial display binding step using existing approaches such as entering a server IP or photographing a QR code [1], each device estimates the pose of the user's head relative to its own coordinate origin. To determine the pose of two devices relative to each other the following transformation takes place in a right-handed coordinate system with the device coordinate system in the screen center (in portrait mode: x-axis pointing right, y-axis pointing to the top of the phone, z-axis pointing out of the phone), see also Figure 2, left:

Let T_i be the pose of the user's head in the coordinate frame of device i and T_j the pose of the head relative to a second device j. The pose of device j in the coordinate frame of the first device i is then given by:

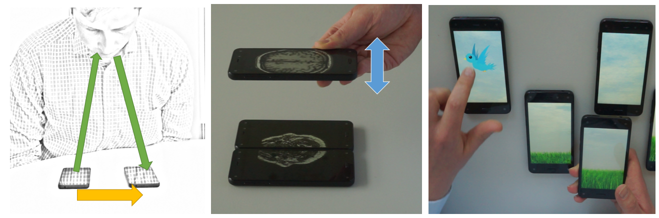

Figure 2: Left: Transformation between two devices is computed by chaining the (inverse) transformations between each device and the user's head. Middle: Browsing through cross sections in medical imaging data. Right: Catch the bird game.

$$T_{ji} = T_i * T_j^{-1} \text{ where } T = [R|t]_{4x4}$$

Our approach supports tracking using six degrees of freedom (DoF). Many head tracking approaches offer less precise orientation tracking around x- and y-axis compared to rotation around the z-axis (pointing out of the display) due to smaller appearance changes in facial features when rotating around x- or y-axis. Hence, we utilize the integrated orientation sensors of smartphones (with a typical accuracy of 1°) to obtain rotation values around these axis. Further, position and orientation variances can be incorporated using common dynamic tracking approaches like Kalman filtering

Registration between multiple devices is accomplished as long as the face is in the joint field of view (FoV) of the devices. The maximum number of devices that can be horizontally (analog for vertically) aligned for a single user is given by:

$$numDevicesHor = ((d/2 - headWidth) * 2)/deviceWidth$$
where $d = tan(a/2) * 2 * z$, a: FoV, z: z-distance of user's face

For example, a single user (head w: 15.0 cm, h: 20.0 cm) sitting in front of Amazon Fire Phones (w: 6.5 cm, h: 13.8 cm) with a horizontal field of view of 97° (vertical 108°, diagonal 120°) at z-distance of 40 cm results in 9 devices horizontally, 5 vertically. Other common mobile phones typically have a front facing camera with ca. 60 – 80 degrees FoV. Using common wide-angle lens adapters for mobile phones, this sensing range can be substantially increased. Also, our approach works with devices with different screen sizes. The devices only need to synchronize their screen sizes at the beginning for initializing the virtual camera positions and canvas size in the renderer. Regarding other spatial configurations the max. number of devices increases as the phones are aligned on a convex surface and decreases on a concave one.

In multi-user scenarios, the sensing range can be extended further. For example, one head serves as a reference frame for a first subset of devices, a second head for a different set of devices. As long as two heads are jointly visible in some devices the mutual transformations between all devices can be determined through the concatenation of multiple transformations. If multiple faces are visible at a time, the relative poses of devices are theoretically independent of the currently tracked face. Practically, the approach can still lead to jitter due to measurement noise when switching faces. Hence, the face tracker (FT) can be combined with a face detector (e.g., [21]) to ensure that only the pose of the first visible face is used; 2D image regions of further identified faces can be masked out to

Figure 3: Map applications multiple map layers can be browsed by changing the height of individual devices.

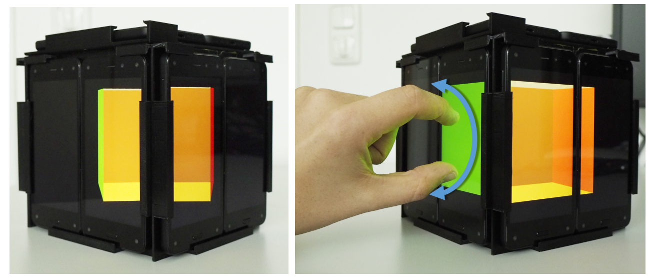

Figure 4: Left: a virtual 3D cube rendered perspectively correct across multiple screens. Right: Separation of spatial manipulation (here rotation) axes through the orthogonal touch surfaces.

avoid tracking. Also, one could explore fusing multiple pose estimates from more than two devices or multiple faces (e.g., using Kalman filtering).

We implemented HeadPhones as client/server application[1]. The clients can join a multi-display group by entering an IP. The server can run on any of the mobile devices resulting in a infrastructure-free, completely mobile solution. One of the devices determines the coordinate origin of the virtual display (typically the device to join first, changeable at any time later).

To work on common mobile devices with a single front camera, we combine a 2D deformable face tracker [15] with a solver for the perspective-n-point problem [8] (Figure 1, middle). In a first step, 2D image points of facial landmarks are estimated using deformable model fitting. For the second step, we use a rigid 3D model which is mapped to selected image points of the 2D model (eyes, nostrils, temples). Alternatively, Fire Phones can be used, giving easy access to a head tracking API.

The head tracking is implemented in C++ (using OpenCV). Applications are written in HTML5 + JavaScript, which are displayed in an embedded browser window within the mobile application, similar to existing approaches [17]. The head tracking data is injected from the C++ level into JavaScript for further use. The applications presented in the next section communicate via WebSocket.

On an Amazon Fire Phone, the applications, including head tracking and registration run at 30 frames per second (fps) using the Amazon API and at 20 fps using our monoscopic face tracker.

APPLICATIONS

We demonstrate the applicability of our approach with a set of applications.

Map navigation: Browsing maps on mobile devices typically requires frequent zooming between levels of detail. Using multiple mobile screens, the extend of a current map level is extended mitigating the need for zooming (to a certain level). Furthermore, the mobile devices can be dynamically repositioned to explore even further map areas, see Figure 1, left. Additionally, the user can pick up individual devices and switch between rendering modes of maps, such as standard view, satellite or traffic views (see Figure 3). This potentially allows viewing different rendering modes of the same map area simultaneously (e.g., if the user looks from the side).

Photo Viewing and Sharing: Similar to the map application, we have implemented a tiled photo viewer which allows the distribution of photos across multiple screens. In addition, users can share photos between devices by first putting their devices next to each other and then flicking the images from one screen to the next (see accompanying video).

Medical Imaging: We have created an image viewer for browsing medical image data sets. A base slice of the image is displayed at the mobile devices lying on a table. The user can pick up an additional device and browse through image slices, see Figure 2, middle.

Shooter game: In spirit of popular aim and shoot games such as crazy chicken ("Moorhuhn"). The goal of the game is to shoot as many birds as possible in a given time, see see Figure 2, right. At the beginning of the game available devices are distributed across the virtual playground. Then, the bird is searched and selected by touch. The search can happen through various interaction techniques. Either, by handholding a single device in air, which constantly hovers over the playground, by sliding a single device on a table, or by moving two devices simultaneously. The game can be turned into a co-operative, multi-player shooter game, with each player controlling up to two devices.

Cubic Display: Staveness et al. have demonstrated a perspective cubic display for Virtual Reality experiences [18], which has inspired a number of follow up works. With HeadPhones, a perspective cubic display can be achieved by assembling the phones into a cube shape with 2 phones on each side see Figure 1. To ensure perspective correct rendering of 3D content, the user's head is used for determining the virtual camera position and frustum for each phone. The touch screens of the mobile phones can be used as interaction surfaces, e.g., for object selection via raycasting or object manipulation via multitouch gestures. The orthogonal arrangement of the touch displays lends itself for separation of manipulation axed. This way, translation, rotation and scale can be independently performed along the x-, y-, z-axis without the need for a mode switch.

EVALUATION

We conducted a technical evaluation and a user study to collect and learn from some initial user reaction and comments on the concept and initial implementation.

Technical evaluation

For measuring the position accuracy, we positioned two Amazon Fire Phones next to each other in portrait mode and gathered accuracy data using 5 faces for side-by-side configurations. For the monoscopic facetracker, the mean deviation from the ground-truth distance between two neighboring devices was 2.07 cm (sd = 1.46) for the vertical offset (y direction), 4.96 cm (sd = 1.08) for the horizontal offset (x direction) and 11.47 cm for the z-direction (sd = 12.00). For the Amazon Fire Phone API the mean error in x was 0.02 cm (sd = 0.5), for y 1.8 cm (sd = 0.3) and for z 4.2 cm (sd = 5.0).

We also empirically determined the range of facial poses. For the Amazon Fire Phone API a face at 40 cm z-distance could be recognized in x-direction [-30..+30 cm], y-direction [-35..+35 cm]. At x = 0, y = 0 the z-range in which a face could be recognized was [10..85 cm]. The orientation range around the x-axis was [-55..+55°], y-axis [-45..+35°], z-axis [0..360°]. For the monoscopic facetracker a face at 40 cm z-distance could be recognized in x-direction [-15..+15 cm], y-direction [-25..+25 cm]. At x = 0, y = 0 the z-range in which a face could be recognized was [15..100 cm]. The orientation range around the x-axis was [-2..+25°], y-axis [-35..+30°], z-axis [-25..+25°].

While we did not formally measure robustness, we empirically found the Amazon face tracking API to be robust to severe backlighting, dark and bright environments as it relies on 4 infrared cameras and emits light through infrared LEDs. Our employed monoscopic facetracker has similar constraints as most monoscopic RGB camera-based techniques (e.g., fails if face is very dark due to strong backlighting). Those properties are not specific to our approach but to face tracking in general, and have been studied before e.g., using the Faces in the Wild database in the work by Saragih et al. [15]. Also, new face tracking algorithms potentially increase the range of facial poses further and the robustness relative to partial face occlusions (e.g., when a face is too close to a camera to be fully seen) [20].

User study

Ten participants (4 male, 6 female, mean age: 23.3 years sd = 3.2) took part in the study. The study was an exploratory study to collect qualitative feedback. We demonstrated the applications to participants and participants could try out HeadPhones by themselves on up to six Amazon Fire Phones. They were asked to think out aloud while exploring. They commented on and rated the ease of use (how easy it was to use the application) and usefulness (how useful they find the application) of the map, game and medical applications, through 2-item questionnaires with 5-items Likert scales. With a questionnaire on social acceptability [14], we also asked participants about suitable locations and audiences for using HeadPhones. After trying the applications, they were asked about the challenges and merits of HeadPhones in semi-structured interviews. The study lasted approximately 60 minutes. Participants where compensated with 10 Euros. When being asked about the potentials and drawbacks of HeadPhones in semi-structured interviews, users highlighted the fact that the idea was "great" and "simple" as "you only need your head, which you carry around anyways". Another user said that this system is designed for the "dumbest assumable user". All but two users described the system as comfortable to use. The other two users mentioned that they felt a little uncomfortable looking towards the camera. One user explicitly mentioned privacy concerns "as the cameras are always watching you". Please note that our approach faces similar privacy concerns as other camera-based interaction techniques, but it does not constantly record the user's face for registration, only when the devices move. The accuracy of the position estimates were concern to one of ten users, who noticed a slight jitter in the applications. Another user speculated about the robustness of the system in backlight situations, as face tracking might not work properly. Participants also speculated about potential other applications, with one mentioning a planning tool for construction sites and another one as collaborative sports analytics application, in which the devices on the table provide an overview, whereas specific details (such as an individual player in a football match) could be picked up and subsequently be investigated in detail.

DISCUSSION AND CONCLUSION

The accuracy measurements indicated that the position estimates in the x and y directions stay low most of the time, whereas there is a larger error in the z-distance. This can have an impact on the design of applications. As long as the mobile devices are co-planar (e.g., lying on a table) applications such as maps can easily be deployed on multiple devices. If applications only need to distinguish between few z-layers (e.g., switching between base, satellite and traffic view in a map), HeadPhones can be employed as well. However, fine grained z-adjustments, are not possible, as of now. Larger deviations in z are due to the smaller image space changes when the z value of 3D points change compared to larger changes in x, y. This is an inherent property of perspective cameras. Jitter can be mitigated by only enabling pose updates when the devices actually move. In our application we turn off the pose updates after the devices did not move for 3 seconds (as indicated by IMU measurements). Still, in these situations the facetracker can be used to determine the position of a virtual camera for user-perspective rendering as in the perspective cube example. The user study indicated that participants liked the idea of using distributed map and medical applications. The game application was not found as very useful, albeit most users indicated that it was fun to play. The study also indicated that most users would use HeadPhones in public spaces, some even while walking. To conclude, we presented HeadPhones, a novel approach for the spatial registration of multiple mobile devices into an ad hoc tiled display. Our approach allows for dynamic repositioning of devices during runtime without the need for external infrastructure such as separate cameras or fiducials. Specifically, our only requirements are local network connections and mobile devices with built-in front facing cameras. This way, HeadPhones enables spatially-aware multi-display applications in mobile contexts. An accuracy study and a user study indicated the feasibility of our approach. In the future, we want to explore new applications for HeadPhones and deploy it to real-world scenarios.

REFERENCES

1. Ming Ki Chong and Hans Gellersen. 2012. Usability Classification for Spontaneous Device Association. *Personal Ubiquitous Comput.* 16, 1 (Jan. 2012), 77–89. DOI:http://dx.doi.org/10.1007/s00779-011-0421-1
2. David Dearman, Richard Guy, and Khai Truong. 2012. Determining the Orientation of Proximate Mobile Devices Using Their Back Facing Camera. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 2231–2234. DOI: http://dx.doi.org/10.1145/2207676.2208377
3. Jens Grubert, Matthias Kranz, and Aaron Quigley. 2016a. Challenges in mobile multi-device ecosystems. *mUX: The Journal of Mobile User Experience* 5, 1 (2016), 5. DOI:http://dx.doi.org/10.1186/s13678-016-0007-y
4. Jens Grubert, Eyal Ofek, Michel Pahud, Matthias Kranz, and Dieter Schmalstieg. 2016b. GlassHands: Interaction Around Unmodified Mobile Devices Using Sunglasses. In *Proceedings of the 2016 ACM on Interactive Surfaces and Spaces (ISS '16)*. ACM, New York, NY, USA, 215–224. DOI: http://dx.doi.org/10.1145/2992154.2992162
5. Ken Hinckley, Gonzalo Ramos, Francois Guimbretiere, Patrick Baudisch, and Marc Smith. 2004. Stitching: Pen Gestures That Span Multiple Displays. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI '04)*. ACM, New York, NY, USA, 23–31. DOI: http://dx.doi.org/10.1145/989863.989866
6. Haojian Jin, Christian Holz, and Kasper Hornbæk. 2015a. Tracko: Ad-hoc Mobile 3D Tracking Using Bluetooth Low Energy and Inaudible Signals for Cross-Device Interaction. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. ACM, New York, NY, USA, 147–156. DOI: http://dx.doi.org/10.1145/2807442.2807475
7. Haojian Jin, Cheng Xu, and Kent Lyons. 2015b. Corona: Positioning Adjacent Device with Asymmetric Bluetooth Low Energy RSSI Distributions. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. ACM, New York, NY, USA, 175–179. DOI: http://dx.doi.org/10.1145/2807442.2807485
8. Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. 2009. Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision* 81, 2 (2009), 155–166. DOI: http://dx.doi.org/10.1007/s11263-008-0152-6
9. Ming Li and Leif Kobbelt. 2012. Dynamic Tiling Display: Building an Interactive Display Surface Using Multiple Mobile Devices. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia (MUM '12)*. ACM, New York, NY, USA, Article 24, 4 pages. DOI:http://dx.doi.org/10.1145/2406367.2406397
10. Heidi Selmer Nielsen, Marius Pallisgaard Olsen, Mikael B. Skov, and Jesper Kjeldskov. 2014. JuxtaPinch: An Application for Collocated Multi-device Photo Sharing. In *Proceedings of the 16th International Conference on Human-computer Interaction with Mobile Devices & Services (MobileHCI '14)*. ACM, New York, NY, USA, 417–420. DOI: http://dx.doi.org/10.1145/2628363.2633569
11. Michel Pahud, Ken Hinckley, Shamsi Iqbal, Abigail Sellen, and Bill Buxton. 2013. Toward Compound Navigation Tasks on Mobiles via Spatial Manipulation. In *Proceedings of the 15th International Conference on Human-computer Interaction with Mobile Devices and Services (MobileHCI '13)*. ACM, New York, NY, USA, 113–122. DOI: http://dx.doi.org/10.1145/2493190.2493210
12. Chunyi Peng, Guobin Shen, Yongguang Zhang, Yanlin Li, and Kun Tan. 2007. BeepBeep: A High Accuracy Acoustic Ranging System Using COTS Mobile Devices. In *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems (SenSys '07)*. ACM, New York, NY, USA, 1–14. DOI: http://dx.doi.org/10.1145/1322263.1322265
13. Roman Rädle, Hans-Christian Jetter, Nicolai Marquardt, Harald Reiterer, and Yvonne Rogers. 2014. HuddleLamp: Spatially-Aware Mobile Displays for Ad-hoc Around-the-Table Collaboration. In *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces (ITS '14)*. ACM, New York, NY, USA, 45–54. DOI: http://dx.doi.org/10.1145/2669485.2669500
14. Julie Rico and Stephen Brewster. 2010. Usable Gestures for Mobile Interfaces: Evaluating Social Acceptability. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. ACM, New York, NY, USA, 887–896. DOI: http://dx.doi.org/10.1145/1753326.1753458
15. J. M. Saragih, S. Lucey, and J. F. Cohn. 2009. Face alignment through subspace constrained mean-shifts. In *2009 IEEE 12th International Conference on Computer Vision*. 1034–1041. DOI: http://dx.doi.org/10.1109/ICCV.2009.5459377
16. Arne Schmitz, Ming Li, Volker Schönefeld, and Leif Kobbelt. 2010. Ad-hoc multi-displays for mobile interactive applications. In *In Proceeding Eurographics 2010*, Vol. 29. 8. http://dx.doi.org/10.2312/ega.20101006
17. Julia Schwarz, David Klionsky, Chris Harrison, Paul Dietz, and Andrew Wilson. 2012. Phone As a Pixel: Enabling Ad-hoc, Large-scale Displays Using Mobile Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 2235–2238. DOI: http://dx.doi.org/10.1145/2207676.2208378

18. Ian Stavness, Billy Lam, and Sidney Fels. 2010. pCubee: A Perspective-corrected Handheld Cubic Display. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. ACM, New York, NY, USA, 1381–1390. DOI: http://dx.doi.org/10.1145/1753326.1753535

19. Zheng Sun, Shijia Pan, Yu-Chi Su, and Pei Zhang. 2013. Headio: Zero-configured Heading Acquisition for Indoor Mobile Devices Through Multimodal Context Sensing. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '13)*. ACM, New York, NY, USA, 33–42. DOI: http://dx.doi.org/10.1145/2493432.2493434

20. Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. 2014. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1701–1708.

21. P. Viola and M. Jones. 2001. Robust real-time face detection. 2 (2001), 747–747. DOI: http://dx.doi.org/10.1109/ICCV.2001.937709

22. Z. Zhang, D. Chu, X. Chen, and T. Moscibroda. 2012. SwordFight: Exploring Phone-to-Phone Motion Games. *IEEE Pervasive Computing* 11, 4 (Oct 2012), 8–12. DOI: http://dx.doi.org/10.1109/MPRV.2012.78