Manipulating Screen Elements in an Immersive Environment with a Wrist-Mounted Device and Free Body Movement

Gyanendra Sharma
Rensselaer Polytechnic Institute
110 8th Street, Troy, NY 12180
sharmg3@rpi.edu

Devavrat Jivani
Rensselaer Polytechnic Institute
110 8th Street, Troy, NY 12180
jivand@rpi.edu

Richard J. Radke
Rensselaer Polytechnic Institute
110 8th Street, Troy, NY 12180
rjradke@ecse.rpi.edu

Abstract

Large display environments supporting fluid user interactions are commonly supported in living lab environments such as situation rooms and industrial lobby spaces. However, conventional input mechanisms such as mouse and keyboard are poorly suited to interacting with such spaces due to their potentially huge spatial extent. In this context, we propose a novel technique for manipulating screen elements in large display environments. The method combines a wrist-mounted Leap Motion controller with user tracking from an array of ceiling-mounted Kinects. The overhead Kinects provide tracking coordinates in a global coordinate system along with the body orientation of the user, creating a dynamic and directed interaction screen space. The Leap Motion controller determines the location of the hand, and interprets it in the context of a large, wrap-around screen. We demonstrate the system in the CRAIVE-Lab, a large immersive environment (10m x 12m floor plan, with a 4.3m high, 40m long, 360 degree display wall).

Author Keywords

Immersive environments, large display, bi-manual interaction, mid-air pointing

ACM Classification Keywords

H.5.m. [Information Interfaces and Presentation (e.g., HCI)]: User Interfaces; Interaction Styles []

The research in this paper was partially supported by the US National Science Foundation under awards CNS-1229391 and IIP-1631674, and by IBM Research via Rensselaer's Cognitive and Immersive Systems Laboratory.
Living Labs: Measuring Human Experience in the Built Environment, in association with CHI'18., April 22, 2018, Montréal, Canada.

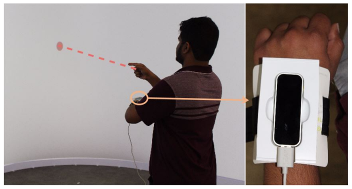

Figure 1: Interaction with screen elements in an immersive environment. (left) A user pointing at an element on the screen using a wrist-mounted Leap Motion controller and orienting his body towards the element. (right) Leap Motion setup.

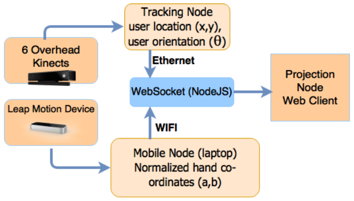

Figure 2: In the mobile node, the Leap Motion device is connected to a compact laptop. A dedicated computer connected to 6 overhead Kinect sensors sends the user's location (x, y) and body orientation θ to a nodeJS server via ethernet connection. A second dedicated workstation drives 8 projectors in the projection node to create the wrap-around 360-degree projection. A web client provides visualization and feedback, allowing users to manipulate objects on the screen.

Introduction and Related Work

As living spaces expand to contain multiple, large displays on different, spatially separated surfaces, instrumenting these spaces with natural interaction techniques is of paramount importance. Recent advancements in sensing technologies, such as LeapMotion and Kinect, have provided inroads towards optimizing interactions for occupants in such spaces. In this paper, we explore such techniques in the context of a large wrap-around immersive display, the CRAIVE-Lab [7]. The physical infrastructure is similar to that of the Cave [2] or CAVE2 [3]. It is well documented how such physical spaces can act as important collaborative and interactive zones, and may very well represent the future of various office and industrial spaces.

Many mid-air interaction techniques in various contexts including desktop screens, large screen systems, and virtual reality have been proposed over the years [8, 5]. Instead of using natural human gestures, now-ubiquitous mobile devices have also been used to enable interaction systems in large scale environments [4, 6, 1].

Simple tasks such as pointing and clicking with a conventional mouse are challenging and cumbersome for such environments, since the cursor may keep disappearing beyond the peripheral vision of the users. In order to interact with the system in a full-screen setting, users have to twist and turn their bodies, looking for the mouse cursor, to complete basic tasks such as pointing, selecting, dragging, and dropping. In this paper, we propose a technique to mitigate this problem and provide users with an alternative interaction technique for large, wrap-around environments. We repurpose the Leap Motion sensor as a wrist-mounted wearable device, as illustrated in Figure 1. Using overhead Kinect sensors in the environment, we determine the global spatial location and orientation of the user, which are used to define dynamic screen spaces. We conducted a user study to evaluate our proposed interaction system, and present quantitative comparisons of the speed of point and drag actions vs. a conventional mouse. We also discuss several accuracy issues of the designed system based on the results of a tracing experiment.

Point and drag with hand and body

In terms of implementation challenges to realize the interaction technique presented in this paper, in which the user-based Leap Motion device is used in tandem with global tracking and orientation data, many different components need to be tied together with a live communication protocol. A brief description of the overall system architecture is provided in Figure 2.

Pointing with Leap Motion

We propose a bi-manual interaction system with a wrist-mounted Leap Motion device. Its small size allows for repurposing it as a wearable system, which to our knowledge has not yet been explored.

The device is mounted on the non-dominant wrist using a band as shown in Figure 1. Participants perform bi-manual pointing actions by holding both hands outwards such that their dominant hand is in the interaction space of the Leap Motion device.

The reported (x, y) coordinates of the hand by the device are normalized with respect to its interaction area. The bottom left side of the interaction box is designated as the $(0, 0)$ coordinate. The position of the dominant hand in the interaction area of the device determines the corresponding screen coordinates. For example, in order to point at an element on the screen which is at the bottom left, a user would have to move the dominant hand towards the bottom left side of the Leap Motion interaction area.

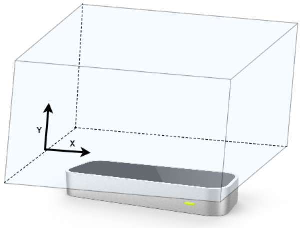

Figure 3: The (x, y) coordinates of the hand reported by the Leap Motion are normalized such that when the hand is close to the bottom left side of the device, it reports values close to $(0, 0)$.

Tracking and Orientation Estimation

- 6 ceiling-mounted Kinects facing the floor act as high-resolution time-of-flight sensors.
- Background subtraction is applied to filter out stationary objects and each depth stream is thresholded at human waist height.
- Rigid transformations calculated during calibration are applied to obtain user positions.
- An ellipse fitting method is used to compute user orientation.

Dynamic Interaction Space

In the context of our large immersive environment, we propose a system in which a user has access to the full wrap-around screen, but the active interaction space at a particular moment is constrained based on his/her spatial location and orientation, computed using the method shown in the box at left.

Since this region changes every time a user moves, we refer to it as the dynamic interaction space (Figure 4). For example, if a user wants to point at an element on the screen that is beyond the space defined in Figure 4, he/she can simply change body orientation, or move to a new region entirely changing the dynamic interaction space accordingly.

In order to successfully perform point and drag actions on the screen based on changing interaction areas, we made several design considerations so that small changes in orientation or tracking don't suddenly shift the interaction space. Instead of using continuous values, we divide the floor into a 2×2 grid and determine the user's location bin based on the tracking data. The orientation is also quantized into 4 bins. Therefore, there are 16 possible dynamic interaction spaces spanning the entire screen. The system is carefully calibrated to account for all 16 possible interaction spaces. This choice of quantization was made so that users can make explicit movements both in terms of orientation and location when they want to choose a different interaction space. In order to visually indicate the dynamic interaction space, it is rendered as two vertical lines that enclose a lightly-shaded gray area.

In order to avoid cases of random hand movements of users translating to pointing to target elements on the screen, we added a time condition so that the intention of the user is clear. We set this time constant to be 3/4th of a second, so that the user has to point at an element continuously for that duration to complete a select or grab action. A visual feedback with the text "GRABBED" appears on the window upon successful completion. Similar textual feedback is provided on completion of a drop action.

Users can freely move around the space and orient themselves as they see fit to manipulate screen elements. The same action is repeated to drop the element. Unlike a traditional mouse, users aren't required to drag the element continuously (e.g., "hold down a button"), which could be uncomfortable for drags spanning the extremely wide screen. The system creates an association of the grabbed element to the user and when the point and hold gesture is repeated, the element is dropped automatically.

Evaluation

We recruited 11 participants (7 male and 4 female); none of them had previously participated in any user studies or experiments in the space.

Experiment 1

In the first experiment, as shown in Figure 4, we wanted to evaluate the speed at which basic pointing tasks could be completed using the proposed system vs. the mouse. The 40m long wrap-around display screen was used in full screen mode, resulting in an effective display size of 1200×14500 pixels. In every trial, a blue dot and a pink square appeared at random points on the screen. The randomness was controlled so that these two artifacts appeared at a separation of at least 2/3rds of the full screen. The task was to select the blue dot, and drag and drop it on top of the pink square. We recorded the time it took for each user to select the dot, along with the duration it took for them to drag and drop the element to the destination.

In order to select the blue dot with a mouse, users had to

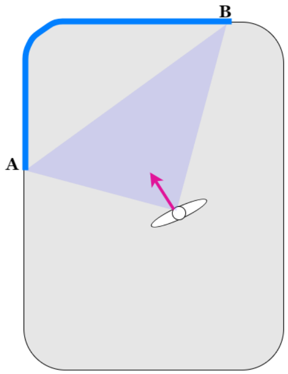

Figure 4: The red arrow represents the forward orientation of the user. Based on the tracking coordinates and orientation, the dynamic interaction screen space is defined, represented by the outer line AB.

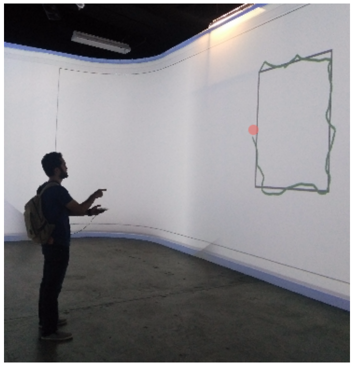

Figure 5: Experiment 2 in action, where a user employs our system to trace a small rectangle on the large screen.

move the mouse as they would generally do on a desktop computer and click on it. To be fair, the mouse cursor was re-designed to appear the same as the one in the Leap Motion setup. Once selected, they would move the mouse and drop it on top of the pink square by clicking on it. We used a wireless mouse for this setup, placed on a table at standing height. Users were asked not to move the table around, but were free to twist and turn their bodies as required to complete the actions.

In order to do the same with our system, users were told that they were free to move around and perform the mid-air pointing action as they saw fit. In order to select the blue dot, they had to turn towards it so that the dynamic interaction space enclosed the dot. Users were then asked to continuously point at the dot for 3/4th of a second, which would complete the selection action. For the drag action, they could simply turn towards the pink square or move to other locations from which pointing would be more comfortable. Users were asked to make sure the dynamic interaction space enclosed the pink square before they could repeat the process, i.e., pointing at the pink square for 3/4th of a second, which would drop the selected dot.

Experiment 2

In this experiment, we wanted to visualize the accuracy of the system. Users were asked to trace over two rectangles on the screen. One rectangle was 500×500 pixels, very small compared to the effective 14500-pixel width of the full screen. The other rectangle was 2200×500 pixels. Using differently-sized rectangles for tracing, we hoped to study the accuracy of our system when users stood close to the interaction screen vs. far from it. Both rectangles were traced using a mouse as well as the proposed system, and the time to complete each task was recorded. Similar to Experiment 1, we demonstrated the experiments to the users before asking them to do it. Each participant was asked to complete 2 trials. The experiment in action is shown in Figure 5.

Results

The results from Experiment 1 are shown in Figure 6. The average pointing time for the mouse was 4.17s with a standard deviation of 3.99s, while for the Leap Motion setup, the average pointing time was 8.80s with a standard deviation of 6.19s. The average dragging time for the mouse was 3.76s with a standard deviation of 1.94s, while the Leap Motion system reported an average dragging time of 5.90s with standard deviation of 2.45s. Considering that fact that all users for these experiments have decades of experience using a mouse in contrast to using the Leap Motion system for the very first time, the results, especially in terms of drag events, look promising. One of the best-performing Leap Motion users was able to accomplish point and drag tasks in 2.88s and 3.12s respectively towards the end of his experiment. These values are well below the average point and drag times for a mouse and thus suggest that with practice, mouse-like speeds are possible, as suggested in Figure 7.

We found that in Experiment 2 (tracing), the mouse tended to perform significantly better in terms of tight conformity to the rectangle. Again, mouse movements that are well-supported by a hard surface allow for greater accuracy in terms of tracing and movement. Despite the challenges that mid-air tracing poses in the absence of a static support, a clear outline around the rectangles shows promising signs towards users' adaptability to mid-air interactions. However, it may be that the proposed interaction technique is better suited to coarser pointing and manipulation than fine-detail work like tracing or drawing.

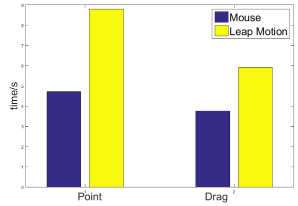

Figure 6: The average time taken to point and drag the elements on the screen for the mouse and Leap Motion systems.

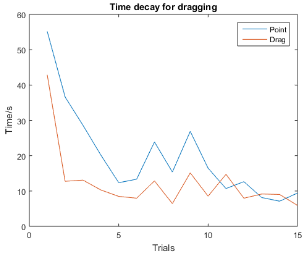

Figure 7: Decaying times to point and drag using proposed system for a sample user.

Discussion and Conclusions

Since users have grown up using a mouse for screen interactions and likely use one for several hours a day, it's not surprising that its performance is difficult to beat with an unfamiliar new input device. However, it is encouraging that the users were able to point and drag using the proposed system at comparable times to that of a mouse. If successful, such interaction techniques will aid in designing systems that allow for natural hand and body movements to manipulate elements on the screen.

Expanding the use of lightweight wearable devices such as the Leap Motion, which support the understanding of fine-grained gestures, will allow for more precise interactions, benefiting users in enclosed indoor spaces. The demarcation of global and local data scopes resulting from the Kinect and Leap Motion sensors, respectively, creates opportunities for user-localized interactions and visualizations, potentially across multiple users at once. Creating collaborative systems is one of the primary motivations for living lab environments, and robust implementations of natural user interaction techniques that go beyond traditional interaction devices is critical. Further instantiations of interaction techniques that leverage both local and global modalities will be necessary to optimize fluid and intelligent interactions in the context of living laboratories.

REFERENCES

1. V. Cheung, D. Watson, J. Vermeulen, M. Hancock, and S. Scott. 2014. Overcoming Interaction Barriers in Large Public Displays Using Personal Devices. In *Proc. of the Ninth ACM International Conf. on Interactive Tabletops and Surfaces (ITS '14)*. 375–380.
2. C. Cruz-Neira, J. Leigh, M. Papka, C. Barnes, S. M. Cohen, S. Das, and others. 1993. Scientist in Wonderland: A Report on Visualization Applications in the CAVE Virtual Reality Environment. IEEE Symposium on Research Frontiers in Virtual Reality, 59–66.
3. A. Febretti, A. Nishimoto, T. Thigpen, J. Talandis, L. Long, JD Pirtle, and others. 2013. CAVE2: a hybrid reality environment for immersive simulation and information analysis. In *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics.
4. S. Jeon, J. Hwang, G. J. Kim, and M. Billinghurst. 2006. Interaction Techniques in Large Display Environments Using Hand-held Devices. In *Proc. of the ACM Symposium on Virtual Reality Software and Technology (VRST '06)*. 100–103.
5. P. Koutsabasis and C. K. Domouzis. Mid-Air Browsing and Selection in Image Collections. In *Proc. of the Intl. Working Conf. on Advanced Visual Interfaces (AVI '16)*. ACM, New York, NY, USA, 21–27.
6. M. Nancel, O. Chapuis, E. Pietriga, X. Yang, P. P. Irani, and M. Beaudouin-Lafon. 2013. High-precision Pointing on Large Wall Displays Using Small Handheld Devices. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems (CHI '13)*. 831–840.
7. G. Sharma, J. Braasch, and R. J. Radke. 2016. Interactions in a Human-Scale Immersive Environment: the CRAIVE-Lab. In *Cross-Surface Workshop at ISS2016*. Canada.
8. S. Yoo, C. Parker, J. Kay, and M. Tomitsch. 2015. To Dwell or Not to Dwell: An Evaluation of Mid-Air Gestures for Large Information Displays. In *Proc. of the Annual Meeting of the Australian Special Interest Group for Computer Human Interaction (OzCHI '15)*. ACM, USA, 187–191.