Tech-note: Device-Free Interaction Spaces

Daniel Stødle* University of Tromsø, Norway
Olga Troyanskaya† Princeton University, USA
Kai Li‡ Princeton University, USA
Otto J. Anshus§ University of Tromsø, Norway

Abstract

Existing approaches to 3D input on wall-sized displays include tracking users with markers, using stereo- or depth-cameras or have users carry devices like the Nintendo Wiimote. Markers makes ad hoc usage difficult, and in public settings devices may easily get lost or stolen. Further, most camera-based approaches limit the area where users can interact.

This paper presents Interaction Spaces – a distributed, optical sensor system for 3D input that lets users interact without needing markers or hand-held devices. An Interaction Space is created that covers the display wall. Inside it, objects like hands or fingers are tracked in 3D. This enables actions like moving or zooming a view on the wall. The added depth dimension allows images to be zoomed using a single hand instead of the two-hand "pinch" gesture used in other systems. The system's distributed aspect enables simple scaling to cover smaller or larger areas.

The system is built using four computers and eight web cameras mounted along the floor. Each camera image is divided into vertical slices. Each slice is processed to detect 1D object positions, before 2D positions are determined using triangulation. The 3D position of an object can be inferred from its corresponding 2D positions in each slice. The system is currently being used to control a microarray visualization on a 2x2 display wall. The system's accuracy has been evaluated, and is shown to be about 1 cm.

Index Terms: I.3.1 [Computer Graphics]: Hardware Architecture—Input devices

1 Introduction

There are many approaches to provide 3D input to applications running on wall-sized displays. A user's hand- or body movement can be tracked using cameras that identify and position a set of passive markers mounted on the user. Users can also carry devices like a 3D mouse or the Nintendo Wiimote, or her location can be determined without markers using stereo- or depth- cameras. These approaches are limited in different ways. The use of markers makes ad hoc usage difficult. Users must spend time mounting the markers to their body, or wear special clothes with embedded markers for full-body tracking. In public settings, a 3D mouse or Wiimote may easily get lost or stolen, and most camera-based approaches limit the area in which interaction can take place.

This paper presents a distributed optical sensor system for 3D multi-point input. The system removes the need for markers and hand-held input devices, enabling the user to interact freely along a wall-sized, high resolution tiled display. The system creates a 3D Interaction Space that is as long and as tall as the display wall itself, and up to about 35 cm deep. The width of the Interaction Space is mainly limited by the number of optical sensors used. Inside the Interaction Space, objects - like a user's hands - are discovered and their 3D position determined. Each object's position is sent to applications in events which can be used for various purposes like moving, zooming, and rotating a view on the display wall.

Using the depth dimension, it is possible to zoom images using a single hand instead of the two-hand "pinch" gesture used in other systems like the Apple iPhone. The system is not limited to detecting hands and using them for input; it can detect any object that gives sufficient contrast to the mostly static background, including for instance the user's elbows, head or other body parts. Since the system avoids using markers or special hand-held devices, ad hoc usage is possible with no preparation on the user's part. The system's intrinsic distributed aspect makes it easily scalable by adding additional cameras and computers, creating larger Interaction Spaces.

The system is currently being used with different applications on two wall-sized displays, including a parallel multi-image viewer and a genomics-application to explore relationships between different microarrays, shown in Figure 1(a). To evaluate the system, experiments measuring the accuracy on a per-slice level have been performed, demonstrating that the system has an average accuracy of about 1 cm for the slice closest to the display wall. The main contribution of this paper is a 3D input system that makes scalable interaction in 2D and 3D possible using commodity components, while still maintaining reasonably good accuracy. Users do not need special devices or markers to interact with the system.

2 Related work

There has been much work on input devices for display walls. The VisionWand [2] provides input by optically tracking a wand-like object in 3D using two cameras, but is limited in that it requires markers and a known object (the wand) to operate. Further, the area in which interaction can take place is limited. When Nintendo introduced the Wii console, they also made the first "mass-market" 3D input device in the "Wiimote." The Wiimote combines accelerometer data with tracking of up to four infrared dots to provide input in 3D, and its potential for "hackability" has enabled the creation of cheap DIY multi-point input devices [7]. The VisionWand and the Wiimote are examples of systems that use markers to provide 3D input.

Another class of 3D input systems employ image recognition to determine the pose of hands and fingers without using markers directly. The Visual Touchpad is an example of this, where a user's hand can be positioned over a specially designed touchpad [9]. Other approaches include using stereo cameras combined with infrared illumination [12], and depth-cameras that capture both color and depth for each pixel in an image [14]. Both systems can provide device-free interaction, but suffer from a lack of large-area coverage. This is a commonality for most camera-based systems: The user has to be inside the camera's field of view, which introduces scalability problems as the size of the area one wishes to interact with goes up. Further, it is not clear how existing systems could be extended to increase the area of interaction. The Interaction Spaces system is designed to be scalable, and is currently used with two different display walls measuring 6x3 and 2.7x2 meters.

The Interaction Spaces system is similar to a number of other multi-touch systems currently available, like the Microsoft Surface [1] and TouchWall, and the Diamondtouch tabletop [3]. Jeff Han pioneered multi-touch sensing using frustrated total internal reflection

*daniels@cs.uit.no
†ogt@genomics.princeton.edu
‡li@cs.princeton.edu
§otto@cs.uit.no

IEEE Symposium on 3D User Interfaces 2009
14-15 March, Lafayette, Louisiana, USA
978-1-4244-3812-9/09/$25.00 ©2009 IEEE

of infrared light [4], which has been commercialized by Perceptive Pixel as a "collaboration wall" and used extensively by CNN to cover the 2008 US presidential election. In [13], the authors describe a system that detects hands and fingers interacting with a whiteboard using a single camera. It is similar to our system in its use of simple image differencing to segment the foreground and background. In [10], a few custom cameras are used to triangulate the position of objects on the SMART Board. This approach differs from ours in its use of custom cameras with on-chip image processing to do object detection. None of these systems provide input in 3D. GestureTek is a company that offers both 2D and 3D camera-based, device-free input solutions [6]. The scalability of their products is unclear, as it appears that all the cameras cover the same (limited) region of interest, but from different angles. In the Interaction Spaces system, different sensors cover different but overlapping regions, cooperating to create a larger space in which interaction can take place. Further, the Interaction Spaces system does not require high-end synchronized cameras, but can operate using commodity web cameras.

3 Design and Implementation

The design of the Interaction Spaces system is based on the following permeating principle: It is more important to determine *where* an object is, as opposed to *what* the object is. This approach is fundamentally different from other camera-based systems. Instead of trying to determine what different objects are – a hand, a finger, a pen, and so on – the system only seeks to discover that an object has entered the Interaction Space, and determine where that object is. The system is based on earlier work that only provided object positions in 2D [11], and uses a set of optical sensors to detect the presence of objects in each optical sensor's field of view. By using information about the relative position of these objects in each optical sensor's view, the object's location can be determined.

To extend this design to 3D, each sensor divides its field of view into a number of distinct slices. Within each slice, the sensor locates foreground objects. As a result, each object's 1D position and extent in a slice is determined for each sensor. A coordinator can then determine an object's 2D position by collecting 1D positions from all the sensors. By treating each 1D position in a slice as a beam from the sensor's position and up, the 2D positions of possible objects can be found using triangulation at the intersections of beams from different sensors, as illustrated in Figure 1(b). To support multi-point interaction and avoid false positives, an object must be initially tracked by at least three sensors. The sensors do not attempt to associate objects from different slices with each other. Instead, the coordinator uses the object extent and calculated 2D position to associate objects from different slices with each other.

The Interaction Spaces system has been implemented using eight commodity web cameras (Unibrain Fire-i @ 640x480 pixels in grayscale) and four computers (Mac mini @ 1.83 GHz). The system is used to interact with applications running on two display walls: One 2.7x2 meter 4-projector, 2048x1536 pixel wall, and one 6x3 meter, 28-projector 7168x3072 pixel wall. The latter is extended with 4 additional Mac minis and 8 additional cameras to cover the entire width of the wall. The software consists an image processing component and an analysis component.

The image processing component runs on each Mac mini to capture images from the cameras, and processes them to detect the presence of foreground objects. Each image is divided into 25 independent vertical slices, as shown in Figure 1(c). Foreground objects are detected using image differencing, thresholding and a dynamically updated background image. This results in clusters of white pixels where objects have been detected. The center position and extent of each cluster is then transmitted using a network event system to the analysis component, along with the slice index in which the object was detected.

The analysis component receives data for each slice from each sensor, and uses it to triangulate object positions in 2D per slice. Since many objects can be detected in a given slice, each individual object must initially be detected by at least three cameras for the triangulation to be successful. This is necessary to avoid false positives caused by the presence of other objects, which would create a number of "ghost objects" where imprints created by one object intersect the imprints from other objects. Such potential false positives are highlighted in Figure 1(b).

To detect an object's 3D position, the analysis component first gathers the information it has about 2D object positions in all available slices. It then begins at the outer-most slice (farthest from the display wall), and assigns 2D objects to new or existing 3D objects. A new 3D object is created any time a 2D object from one of the outer-most slices appear that are considered too far from any already existing 3D objects, or if there are more 2D objects in a given slice than there are existing 3D objects. At present, the system is limited to associating a single 2D object from each slice to a given 3D object; this means that an arm that extends into the Interaction Space and then divides into an open hand with spread fingers will only use one of the finger positions, instead of incorporating all of them into the same 3D object.

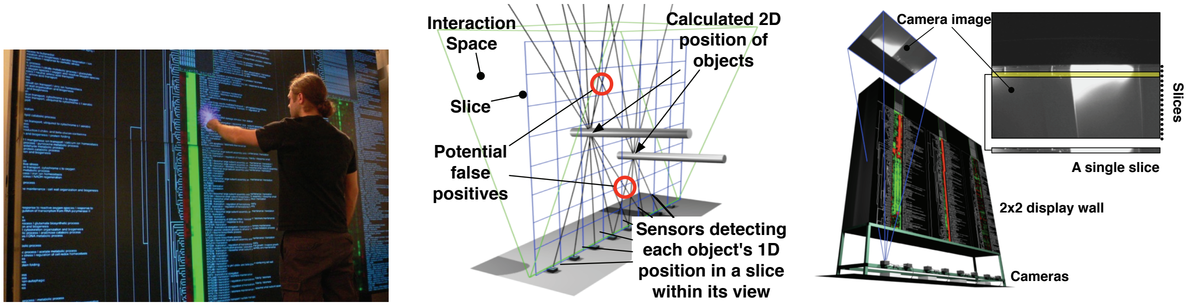
Once a 3D object has been detected, an event is created containing the location of the object's tip in 2D, and its depth position, as well as events for the raw 2D locations for each object in each slice. These events are used by applications to enable interaction. Currently, the depth position is a direct translation from the slice index, with a value of 0.0 corresponding to touching the wall, and a value of 1.0 being the outer-most slice that is recognized; in the future this will correspond to the actual depth in real world units. The 2D position is reported in centimeters relative to the left side of the display wall and the floor.

To provide accurate output, the system is calibrated using a limited camera model with the following parameters: position, field of view, left-right rotation and distortion (further refinement is planned). To calibrate the cameras, an operator touches 18 target points with known real world coordinates on the display wall. Each camera records the position of the object it detects, if any. When all the targets have been touched, each camera will have a set of detected objects and their associated real world coordinates. Each camera is then automatically adjusted by iteratively changing different parameters with the goal of minimizing the error between known target location, and where that object would be placed given the current camera parameters.

4 Applications

Interaction Spaces is currently being used to interact with a system for visualization of genomic microarray data, shown in Figure 1(a). The viewer a custom display wall version of HIDRA [5]. In HIDRA, users explore different datasets by selecting genes in one dataset, and observing where they are located in other datasets. Genes in close proximity to each other in the microarray indicates that they might be correlated.

To use the viewer, the system must support navigation and selection of genes. In Interaction Spaces, these actions are mapped to moving ones hand in the space in front of the display wall. The depth dimension is used to control what action is performed. If the user touches the display, the genes under the user's finger will be selected. Otherwise, the view is panned according to the user's hand movements. To avoid accidental panning when the user intends to select genes, the system prevents panning if the object is seen to move closer to the wall. The viewer can also be controlled using an iPhone. The depth dimension is also used to control zoom in an image viewer application, where the view zooms closer as the user's hand approaches the wall, and zooms back out when the hand is moved away.


Interaction Space
Slice
Potential false positives
Calculated 2D position of objects
Sensors detecting each object's 1D position in a slice within its view
Camera image
Slices
A single slice
2x2 display wall
Cameras

(a) (b) (c)

Figure 1: (a) Interacting with a microarray visualization. The bright spot under the user's finger is in reality a set of animated particles that tells the user that he is giving input to the system. (b) An object's 2D position in the center slice is found by triangulating the 1D positions detected by the different sensors. The two circles indicate potential false positives if only two sensors were to be used in determining the position of the object. (c) A sample image from one of the cameras, and its relation to the world.

5 Evaluation

There are two important technical performance metrics when evaluating input devices: Latency and accuracy. Latency is important to help users create a connection between the actions they make, to what they see happen on screen [8]. Good accuracy is important for selecting small targets or do other tasks that require precise input. However, a system could in principle be used even in the face of great inaccuracies if the applications were designed to expect noisy and inaccurate input. Previous work [11] has shown that the latency of the Interaction Spaces system is about 115 ms. This evaluation will focus on accuracy. In the Interaction Spaces system, there are many variables that together determine the total system accuracy. The distributed nature of the system means that sensors covering different areas may combine to produce very different accuracy levels for the areas they cover.

It is difficult to design an experiment that objectively and empirically measures the accuracy of an input system such as the one presented in this paper, while enabling other researchers to reproduce the results in a consistent manner. Without designing a mechanical arm or similar that can be made to consistently produce the exact same movements, the only option left is to have one or several users test the accuracy of the system. However, such tests are very hard to reproduce, and will inevitably be affected by the different characteristics of each user. Thus, such tests do not help in methodically exploring how changes to the system affect its accuracy. For instance, to quantify the effect of varying lighting conditions or changing the foreground/background segmentation algorithm, it is essential that the experiment be repeatable and identical to earlier trials. For this case, users are not useful, as they will be hard-pressed to conduct the exact same movements time and time again.

In spite of these concerns, the system's accuracy in positioning an object at the innermost slice was measured by having a user interact "mechanically" with the system. The user touched 100 target points on the display wall in turn. For each target, the system recorded the currently detected object's position 30 times. Each target was shown alone on the display wall as a white square on a black background. To provide the user with feedback about what the system detects, a fountain of particles appear at the location where the system thinks the object is. Once the system has gathered enough samples for the target, the screen briefly flashes to indicate its readiness to sample the next target, and the next target appears on the display wall.

The results are shown in Figure 2. The accuracy of the system is measured in centimeters. The X axis indicates the offset from the left side of the display wall, which measures 272 cm in total. The Y axis shows the offset from the bottom of the display wall (not the floor) to the top, which measures 202 cm. All the targets were located within an interior rectangle that measured 167x60cm. The size of this area was chosen based on the typical area in which the system is used for interaction; anything much above is usually too far up to reach without effort, and the system does not support initial touches to the far left and far right (which is what the experiment tests), as these objects are only seen by two cameras. This is one less than the three that are required to get a positive lock on the object (note that once the system has acquired an object, it can be tracked even if only seen by two cameras).

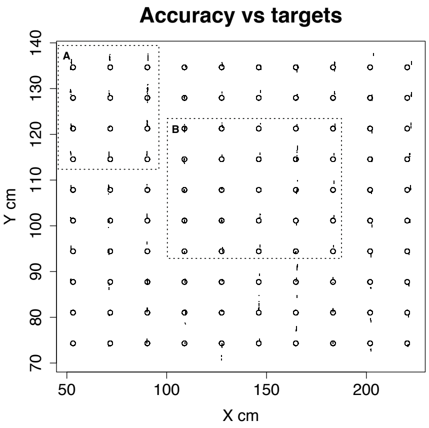


Figure 2: 100 targets (circles) and the positions detected by the system for each target (dots). The X and Y axis show the horizontal and vertical location on the 2.7x2 m display wall. Boxes A and B highlight areas where the system exhibits low and high accuracy.

The plot indicates that the system is more accurate along the horizontal axis than the vertical axis, with the mean horizontal delta (dX) between target and observed location being -0.21 cm, and the mean vertical delta (dY) -0.47 cm. The mean distance from observations to actual targets is 1.1 cm, with a 0.72 cm standard deviation. Further, 90% of the targets had a vertical standard deviation less than 0.5 cm, and 93% of the targets had a horizontal standard deviation less than 0.1 cm.

Two boxes are highlighted in Figure 2. Box A shows an area where the system exhibits low accuracy. The detected object's location flickers up and down (as indicated by the vertical spread of the dots), while the object's position along the X axis remains fairly constant. Inside box B, the system appears to be more accurate: Apart from a few problem spots, most samples are very close to their targets. Why does the system exhibit such differing accuracy behaviour? Some answers to this question will be given in the discussion.

6 Discussion

The evaluation has shown that the system exhibits differing accuracy-levels depending on where the user interacts. To analyze why this is the case, it is necessary to know which factors impact the system accuracy. The factors that govern accuracy in the Interaction Spaces system are: (i) Timing of data from the sensors, (ii) object speed, (iii) lighting, (iv) precision of the foreground/background segmentation, (v) object extent, (vi) physical placement and alignment of the sensors, and (vii) system calibration.

Timing of sensor data is important when the objects being tracked are moving. Since the system relies on unsynchronized web cameras, one step of the triangulation may rely on data from different cameras separated by up to 33 ms. The experiment was designed to eliminate both timing and object speed as factors, by keeping the object to track stationary.

The precision of the foreground/background segmentation is one factor that helps explain some of the observations made in the evaluation. Typically, when the position of a detected object exhibits much vertical jitter, a single camera "flickers" between detecting and not detecting the object within that particular slice, or detecting it at two slightly offset 1D positions due to random noise or lighting effects. This causes the position to jitter up and down.

As an object's extent grows wider, the extent of the lines projected from the cameras grow. If segmentation was perfect and the cameras perfectly synchronized, the object extent would not play a role in the system's accuracy. While not entirely eliminated in the evaluation (the extent of a user's finger may vary slightly depending on the exact finger pose), it is not a major factor. As part of the evaluation, the object extent was recorded for each sample, varying between 2.34 and 6.49 pixels.

The cameras' physical alignment is important to give a best possible starting point for determining object positions. Since perfect alignment is difficult to achieve, the system requires calibration. In the experiment, most samples had a standard deviation less than 0.1 cm horizontally and 0.5 cm vertically. Thus, while the system's accuracy varies from target to target, it is consistent in where it positions objects relative to the targets, pointing to calibration as the cause of the varying accuracy, since different cameras may have been calibrated more or less accurately.

Some ways to improve the system accuracy would include using infrared illumination coupled with visible-light filters on the cameras, or using cameras with higher framerates or lower noise. Lighting also affects the quality of the segmentation. The results presented in the previous section were gathered under fairly ad hoc lighting conditions, with two lamps mounted in the ceiling giving good backlight to parts of the scene, but not covering it completely, leaving large dark regions (visible in Figure 1(c)). Despite the varying light levels, the system yields an accuracy of about 1 cm.

The evaluation has only touched on the accuracy of positioning an object within a slice, and not on how this relates to the depth dimension. Since every slice is processed in the same way, it is reasonable to assume that the resulting accuracy would be roughly the same. However, conducting an experiment to prove this is very difficult, as it is hard for users to accurately point at targets shown on a display wall as much as 35 cm away. For this reason, an evaluation of the slice accuracy further away from the display wall is left as future work. Support for recognition of 3D object pose (such as pointing direction) and a more refined camera model is also planned, as well as an evaluation of the system's accuracy when tracking moving targets.

7 Conclusion

This paper has presented the Interaction Spaces system for providing 3D input to applications running on wall-sized displays. The system allows one or several users to interact with applications simultaneously using one or both hands, or any other object or body part they might see fit to use. Unlike most existing systems, the Interaction Spaces system can easily be extended to provide interaction along larger areas. It is non-intrusive, in that it does not require users to wear special markers or carry devices to interact. The system is in use with several different applications, including an application for microarray visualization used to study relationships between genes. In this application, the depth dimension can be used to differentiate between navigating the visualization and selecting genes. The experiments have demonstrated that the system has an average accuracy of about 1 cm for the innermost slice.

Acknowledgements

Supported by the Norwegian Research Council (159936, 155550), NSF (DBI-0546275), NIH (R01 GM071966, T32 HG003284), NIGMS (P50 GM071508). Thanks to Matthew Hibbs, Tor-Magne S. Hagen and Espen S. Johnsen.

References

[1] S. Bathiche and A. Wilson. Microsoft Surface, 2007. http://www.microsoft.com/surface/.

[2] X. Cao and R. Balakrishnan. VisionWand: interaction techniques for large displays using a passive wand tracked in 3D. In *UIST'03: Proceedings of the 16th annual ACM Symposium on User Interface Software and Technology*, pages 173–182, 2003.

[3] P. Dietz and D. Leigh. DiamondTouch: a multi-user touch technology. In *UIST'01: Proceedings of the 14th annual ACM Symposium on User interface Software and Technology*, pages 219–226, 2001.

[4] J. Y. Han. Low-cost multi-touch sensing through frustrated total internal reflection. In *UIST'05: Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 115–118, 2005.

[5] M. Hibbs, G. Wallace, M. Dunham, K. Li, and O. Troyanskaya. Viewing the Larger Context of Genomic Data through Horizontal Integration. *IV '07: Information Visualization*, pages 326–334, July 2007.

[6] E. Hildreth and F. Macdougall. Multiple camera control system, June 2006. US Patent no. 7058204.

[7] J. C. Lee. Hacking the Nintendo Wii Remote. *IEEE Pervasive Computing*, 7(3):39–45, 2008.

[8] I. S. MacKenzie and C. Ware. Lag as a determinant of human performance in interactive systems. In *CHI'93: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 488–493, 1993.

[9] S. Malik and J. Laszlo. Visual touchpad: a two-handed gestural input device. In *ICMI'04: Proceedings of the 6th Int'l Conference on Multimodal Interfaces*, pages 289–296, 2004.

[10] G. D. Morrison. A Camera-Based Input Device for Large Interactive Displays. *IEEE Computer Graphics and Applications*, 25(4):52–57, 2005.

[11] D. Stødle, P. H. Ha, J. M. Bjørndalen, and O. J. Anshus. Lessons Learned using a Camera Cluster to Detect and Locate Objects. In *ParCo'07: Proceedings of Parallel Computing: Architectures, Algorithms and Applications.*, volume 15 of *Advances in Parallel Computing*, pages 71–78. IOS Press, 2008.

[12] W. M. Vieta and M. Bell. WaveScape: a practical robust display with a 3D gesture interface. In *IPT/EDT'08: Proceedings of the 2008 workshop on Immersive projection technologies/Emerging display technologiges*, pages 1–2, 2008.

[13] C. von Hardenberg and F. Bérard. Bare-hand human-computer interaction. In *PUI '01: Proceedings of the 2001 workshop on Perceptive user interfaces*, pages 1–8, 2001.

[14] G. Yahav, G. Iddan, and D. Mandelboum. 3D Imaging Camera for Gaming Application. *ICCE'07: Int'l Conference on Consumer Electronics.*, pages 1–2, Jan. 2007.