
A Toolkit for Building Collaborative Immersive Multi-Surface Application

Cooper Davies¹

Cooper.Davies@agilesoftware
engineering.org

Jade White¹

whitejk@ucalgary.ca

Alec McAllister¹

Alec.McAllister@agilesoftware
engineering.org

Adam Saroka¹

asaroka@ucalgary.ca

Omar Addam¹

omaddam@ucalgary.ca

Fatemeh Hendijani Fard¹

fhendija@ucalgary.ca

Frank Maurer¹

fmaurer@ucalgary.ca

¹ University of Calgary
2500 University Drive NW
Calgary AB, Canada

Abstract

The paper describes a toolkit that integrates spatially-aware multi-surface systems with mixed-reality approaches to create immersive collaborative environments. The toolkit integrates multiple digital displays and multiple Microsoft HoloLens devices with

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author. Copyright is held by the owner/author(s).
ISS '16, November 06-09, 2016, Niagara Falls, ON, Canada
ACM 978-1-4503-4248-3/16/11.
<http://dx.doi.org/10.1145/2992154.2996879>

multiple Kinects. The HoloLens' allow several users to look at the same virtual hologram while the Kinects enable them to use body movements to interact with these holograms as well as with other digital surfaces in a space. Effectively, the toolkit enables its users to build collaborative applications that utilize digital displays as well as the space between them to interact with information. Our approach also facilitates the management of virtual objects and overcomes gesture restrictions of HoloLens. We use the toolkit in a prototype application for improving the efficiency and response time in emergency management.

Author Keywords

Augmented Reality; gestures; multiple devices; interaction; hand tracking.

ACM Classification Keywords

I.3.6

Introduction

Interacting with 3D mixed-reality models requires substantial amounts of technology and/or usability trade-offs. While head mounted devices such as HoloLens offer a limited gesture set for interactions [1], depth cameras such as Microsoft Kinect provide a full body interaction along with a variety of gestures that

can be useful for interacting with 3D models. The usability of complex multi-surface systems can be enhanced if this society of devices is treated as a single environment [2,3]. To go beyond showing information on digital surfaces, the 3-dimensional space between different surfaces can be utilized for mixed-reality models. To help developers, we developed a tool that manages 3D mixed reality models and enables users to interact with these virtual objects. This paper focuses on how we use the Microsoft HoloLens and Kinects to create the immersive collaborative experience.¹ Combining the interface of the HoloLens with the functionality of the Kinect has introduced many opportunities for information handling or manipulation. We have created an infrastructure by which the two devices are able to communicate. The toolkit utilizes multiple HoloLens and/or Kinect devices to enable multiple users to view the same data within the same space. Our framework can be expanded even further to include multiple other devices, restricted only by their ability to connect to our server. By defining a standard message format, there are potentially a limitless number of devices that can be connected and integrated. We have previously published part of this toolkit (SoD) [3]. We use the toolkit in a collaborative industry project to prototype technology for an Emergency Operations Centre (EOC) of the Future to supports analyze and plan emergency situations.

Related Work

Various applications and frameworks have been developed that use head mounted devices, wearables, Kinect, etc., to interact with the 3D mixed reality models and create seamless interaction of devices.

¹ See ***SoD ITS 2015 paper*** for details on the multi-surface aspect of the approach.

Mayer and Soros use smart glasses and watches to select objects and interact with them in the real world [4]. XDKinect is a framework that uses Microsoft Kinect as a mediate for cross device interaction (allow users to interact with multiple devices) [5]. In [6] authors study how to manipulate or interact with virtual objects using various gesture control systems or gesture interactions via Kinect or other devices. Different projects are explained in [7] that separately use Kinect and GearVR headset as well as other devices to interact with virtual objects. The most similar work to ours is HoloR [1], an augmented reality system that uses projectors, goggles, and Kinect. It tracks the data and forwards the Kinect primary gestures to the app. However, the use of HoloLens and the details of Kinect and lens interactions are not mentioned in [1]. The main difference of our work with HoloR is the ability to use SoD built in gestures on the HoloLens to manage a virtual object and overcome the restriction of HoloLens to create new gestures.

Engine

The HoloLens is tied in with the Unity Engine for model creation. Unity uses a type of file (AssetBundle) which automates the process of 3D model serialization and deserialization. Integrating these capabilities allows us to shift focus from data optimization, to how the data is transferred.

Framework

To allow for cross-device communication, we use our "Society of Devices" (SoD) toolkit [3]. SoD is a message relay server to which multiple devices connect, and register themselves with. SoD relays

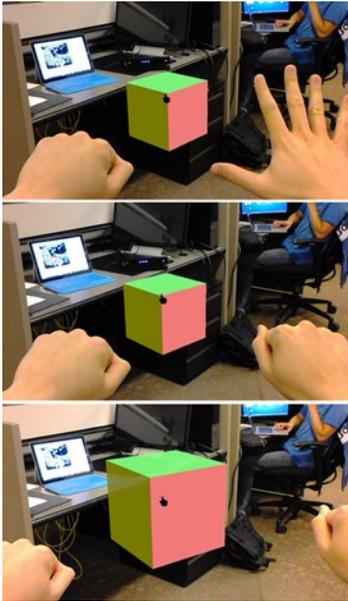


Figure 1. Real-time use of the Kinect in coordination with the HoloLens. In frame one, the user anchors the object by closing one hand. In frame two, the user positions his second hand in order to calculate relative distance between hands. In frame three, the user enlarges the object by widening his hand position.

messages between connected devices. Currently, similar to standardized network socket libraries, SoD contains commands such as Connect, On, and Emit, has the ability to exchange information, and implement additional commands (e.g. declaring the single target of a message, asking for the number of people the Kinect are able to see, etc.). SoD can be used as a file server. We serialize data from files and upload them through Emit messages to be stored on the server. Multiple HoloLens are then able to download these files, deserialize them, and use the information accordingly. This allows for the sharing of 3D models and their location in physical space amongst multiple HoloLens devices in real time.

Gestures

One of the limitations of the HoloLens (restrictions placed on its API by Microsoft) is that it does not allow for the creation or addition of new gestures and provides exactly two gestures. Because users feel comfortable using more than two gestures [8] we decided to integrate Kinect gestures that are already distributed by SoD to overcome this limitation. Using the Kinect, we can place the users' hands within the 3D realm of the HoloLens. By giving their real-world hands a virtual position in the same space as the mixed reality objects, interaction within that 3D world becomes possible. To interact with virtual objects, a "grab" gesture seems to be an obvious choice. The HoloLens already supports a grab gesture, but we found it to be very unnatural feeling during its use. It requires a very awkward, and deliberate motion, which goes against our intuitive approach. We felt this to be very contradictory, and attempted to create alternatives.

Along the way, we implemented a mechanism which determines the specified object we are attempting to

interact with. This raised a number of issues. If we were to use hand positions, we could not assume that the world the HoloLens views is comparable in scale or position as the world the Kinect views. Neither device knows how the other interprets the physical world. We cannot be sure that the HoloLens has the room mapped similarly to that of the Kinect to any degree of accuracy. SoD merely relays along what the Kinect tells it to. We use the standard gaze mechanism used in the HoloLens to decide which object the user is interacting with. A ray is cast from the current camera perspective, down the line of sight, until it collides with an object that we have tagged as interactable. The collided object becomes considered the current focused object. When a user performs a grab, the focused object is what they interact with. This cross-device integration enabled us to provide intuitive interactions with 3D models.

Interaction

Implementing the Kinect allows for a more fluid and natural interaction between the 3D world and the user. The user has to simply close their hand to instantiate a grab event and the system handles the rest. The Kinect invokes an emit function which sends a "Grab started" message to SoD passing the users/hand position, current gestures, etc. The HoloLens receives the message from SoD, looks at the hand positions, and then invokes its own grab event. If the User then opens their hand, invoking a "Grab released" event, this process is repeated. While the grab released event is not sent, the HoloLens will poll SoD every 100ms to ask the Kinect the current position of hands. Every time the HoloLens receives the hand positions from SoD, it uses these positions to determine how to alter the virtual world. For example, in Figure 1, we have the HoloLens set to recognize that a grab indicates we are

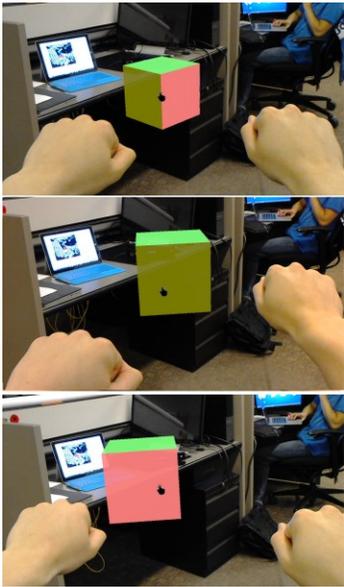


Figure 2. Real-time use of the Kinect in coordination with the HoloLens. Similar to Figure 1 except that the gesture is used to rotate the cube rather than enlarge it

attempting to interact with the focused object. We then begin polling to find the relative distance between the users' hands with respect to their original distance. We can also do the same for rotating as shown in figure 2. By implementing more gestures via the Kinect, there is essentially a large number of body movements to signify commands. By extending the model from that of a simple 3D cube, we can allow for users to manipulate individual buildings on a map. By introducing different gestures, the user can clone, move, orient, scale, and rotate buildings with our current design.

Conclusion & Future Work

Based on current state of the project, we believe that there is a significant number of applications for this technology. The toolkit will substantially decrease the

amount of time taken build immersive multi-surface applications. In the future, we will explore immersive analytics applications with a focus on enhancing EOCs.

In addition, we will integrate Google Tango tablets to view the same virtual holograms. This would be a low cost alternative to look at virtual holograms but will also require to rethink gesture interactions. On the technical side, we will store 3D objects and their positions in a database, enabling us to show or hide parts of an object. By storing the information as opposed to an AssetBundle file, we would be able to download the models to a non-unity engine if need be.

Acknowledgements

We would like to thank our industry partner, C4i Consultants Inc. for their support.

References

- [1] Schwede, C. and Hermann, T. HoloR: Interactive mixed-reality rooms. *CogInfoCom, IEEE 2015*, 517-522.
- [2] Rädle, R., Jetter, H.C., Schreiner, M., Lu, Z., Reiterer, H. and Rogers, Y. Spatially-aware or spatially-agnostic?: Elicitation and evaluation of user-defined cross-device interactions. In *Proc. Human Factors in Computing Systems, ACM (2015)*, 3913-3922.
- [3] Seyed T, Azazi A, Chan E, Wang Y, Maurer F. SoD-Toolkit: A Toolkit for Interactively Prototyping and Developing Multi-Sensor, Multi-Device Environments. In *Proc. ITS 2015, ACM (2015)*, 171-180.
- [4] Mayer, S. and Sörös, G. User Interface Beaming--Seamless Interaction with Smart Things Using Personal Wearable Computers. In *Wearable and Implantable Body Sensor Networks Workshops, 2014 11th International Conference, IEEE (2014)*, 46-49.
- [5] Nebeling, M., Teunissen, E., Husmann, M. and Norrie, M.C. XDKinect: development framework for cross-device interaction using kinect. In *Proc. 2014 ACM SIGCHI Symposium on Engineering interactive computing systems, ACM (2014)*, 65-74.
- [6] Fernandes, L., Nunes, R.R., Matos, G., Azevedo, D., Pedrosa, D., Morgado, L., Paredes, H., Barbosa, L., Fonseca, B., Martins, P. and Cardoso, B. Bringing user experience empirical data to gesture-control and somatic interaction in virtual reality videogames: an exploratory study with a multimodal interaction prototype. In *SciTecIn15, 2015*.
- [7] Perlin, K. Future Reality: How emerging technologies will change language itself. *IEEE Computer Graphics and Applications, 36.3 (2016)*, 84-89.
- [8] Chan, E., Seyed, T., Stuerzlinger, W., Yang, X. D., & Maurer, F. User Elicitation on Single-hand Microgestures. In *Proc. CHI 2016, ACM Press (2016)*, 3403-3414.