Mixed Reality Simulation with Physical Mobile Display Devices

Mathieu Rodrigue[*] Andrew Waranis [†] Tim Wood [‡] Tobias Höllerer [§]

University of California, Santa Barbara

Abstract

This paper presents the design and implementation of a system for simulating mixed reality in setups combining mobile devices and large backdrop displays. With a mixed reality simulator, one can perform usability studies and evaluate mixed reality systems while minimizing confounding variables. This paper describes how mobile device AR design factors can be flexibly and systematically explored without sacrificing the touch and direct unobstructed manipulation of a physical personal MR display. First, we describe general principles to consider when implementing a mixed reality simulator, enumerating design factors. Then, we present our implementation which utilizes personal mobile display devices in conjunction with a large surround-view display environment. Standing in the center of the display, a user may direct a mobile device, such as a tablet or head-mounted display, to a portion of the scene, which affords them a potentially annotated view of the area of interest. The user may employ gesture or touch screen interaction on a simulated augmented camera feed, as they typically would in video-see-through mixed reality applications. We present calibration and system performance results and illustrate our system's flexibility by presenting the design of three usability evaluation scenarios.

Keywords: Augmented reality, virtual reality, large displays, immersive displays, mobile device, input device, interaction techniques

Index Terms: H.4 [Information Systems Applications]: Miscellaneous—;H.5.2 [Information Interfaces and Presentation (e.g. HCI)]: User Interfaces—Input devices and strategies, Interaction styles

1 Introduction

Evaluation and usability engineering of Mixed Reality (MR) interfaces and applications are inherently difficult to control [9] with users exposed to real-world confounds and sometimes brittle experimental technologies. This limits the systematic exploration of MR design spaces and therefore poses challenges for devising new systems, applications and interfaces. Previous MR simulation setups did not allow for realistic exploration of scenarios involving personal Augmented Reality (AR) displays such as hand-held displays or specific personal eyewear; a virtual representation of the mobile display device in VR causes the loss of important affordances. In this work, we take the approach of coordinating simulated backdrop displays (representing the real world) and augmented simulated camera streams (representing video-see-through AR) for MR simulation purposes.

Personal-display-based AR is of particular interest to be studied in a mixed reality simulator because of its increasing popularity and at the same time unclear ergonomics. In what situations and using which design parameters will hand-held AR magic lenses provide their best performance and meet highest user acceptance? What immersion factors (e.g. display field of view, tracking latency or jitter) impact near-eye AR user performance and satisfaction, and in what ways? When performing AR usability studies, many confounding factors exist in the world, such as lighting and visibility influences, dynamic environments, or sound interference. Additionally, real-world noise might affect crucial components of the AR system such as localization and registration, inhibiting accurate user data. MR simulation [11] can be used to overcome the traditional issues with mixed reality user studies by allowing systematic control over these variables.

Figure 1: Mixed reality simulation in our display environment.

MR simulation helps overcome confounding user study conditions including, but not limited to: poor tracking and registration, unsafe or impractical testing conditions, and the inability to use equipment that is, at time of evaluation, prohibitively expensive, not-yet fully functional, or immobile. Real-world variability needs to be controlled in order to run meaningful reproducible studies that truly expand knowledge in MR interface usability. At the same time, physical aspects of personal MR devices such as hand-held or near-eye displays are important evaluation factors, and if a prototype form factor exists it should be used for evaluation rather than being simulated in VR or approximated by passive props.

MR simulation is the process of simulating all aspects of a MR system to carefully control the pertinent variables in user studies. By simulating aspects of the hardware, software and the environment, one has full control over all factors in an AR experiment, which is often difficult or even impossible to achieve through conventional user testing. With MR simulation, one can design and compare a wide range of AR system variants, current models and futuristic possibilities alike, and at the same time minimize noisy experimental data caused by confounding variables.

Until now, MR simulation has mainly been implemented using fully immersive head-mounted displays. To our knowledge, this is the first design and implementation of a system consisting of large backdrop environment displays to simulate the real world, and actual physical mobile devices to simulate AR overlays (Figure 1). The system's infrastructure was designed to flexibly control and coordinate backdrop and AR views, and to be easily adopted and

[*]e-mail: mathieu@cs.ucsb.edu
[†]e-mail: andrewwaranis@cs.ucsb.edu
[‡]e-mail: fishuyo@mat.ucsb.edu
[§]e-mail: holl@cs.ucsb.edu

customized.

2 Related Work

Some of the first MR simulation work was conducted by Gabbard et al. [7] who explored the effects of lighting conditions on text legibility in a simulated AR setting. Lee et al. [9, 11], studied mixed reality simulation via several application and validation studies. In their implementation, a user was immersed in a virtual world by wearing a tracked head-mounted display and interacting with a tracked wand. A translucent window in the center of the display viewport acted as an AR magic lens and showed the virtual world as well as simulated AR augmentations. Both reality and AR augmentations were thus simulated in an HMD-based virtual environment.

Magic Lens Interaction. Magic lens interfaces have been a topic of interest in the AR community since its conception. Bier et al. [2] introduced magic lenses as tools to analyze complex 2D data by enhancing interesting data, diminishing distracting data or displaying hidden data. Szalavári and Gervautz [16] presented an early design and implementation of a magic lens interface for AR known as the Personal Interaction Panel (PIP). The PIP can be used to interact and augment the environment as well as navigate and view the augmentations that are placed in the world. Brown and Hua [5] present a system for a magic lens interface in augmented virtual environments. Cheng, Li, and Müller-Tomfelde [6], among others, present a system for viewing, interaction and collaboration of complex data with hand-held devices. Little is known about the true usability of such applications involving hand-held AR devices, and usability testing is challenging. Our MR simulator is set up to evaluate and compare interfaces and applications for AR magic lens setups, as now commonly employed on tablets and smart phones.

Display Simulation. It is common practice to simulate aspects of novel display technologies and evolve display parameters and features before actually building a first prototype or product [15, 1, 10, 12]. Grubert et al. [8] implemented a system consisting of mobile devices and backdrop displays, which was primarily focused on a novel type of user-perspective rendering for public display AR. Ostkamp et al. [13] implemented a system with mobile devices and backdrop displays that was meant for rapid prototyping and simulation. However, the system implemented by Ostkamp et al. is restricted to pre-recorded videos, which can't be configured or controlled as needed by the MR application developer or researcher. This system also does not simulate the camera feed, losing the ability to control the mobile device display and camera hardware parameters. In our implementation of a personal display-based MR simulator, we employ the physical form factors of existing hand-held or near-eye video-see-through AR displays, but simulate the camera input in coordination with the device's tracked pose and a fully configurable, simulated real-world backdrop.

Figure 2: Mixed reality simulation using a mobile device in our display environment. The simulated real-world is shown as a backdrop and the augmenting device is shown with a rendered view that was streamed wirelessly from the simulator server.

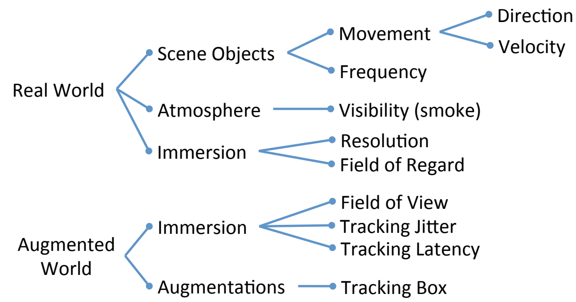

Figure 3: A taxonomy of the forest fire scenario.

3 MR Simulation Design

The MR simulator acts as a testbed to evaluate MR interfaces and scenarios in a controlled environment. The simulator is controlled through various parameters, allowing the experimenter to more easily design a realistic environment for a particular testing scenario. As an example project in basic research, an experimenter could use the simulator to evaluate various components of immersion [3] (e.g. field of view or resolution) to optimize certain task performance.

The experimenter must have control over the entire environment when evaluating a system's usability. The environmental conditions can be expressed as a structured set of parameters for a particular scenario. Generally, MR simulation can be split up into its "real" world and augmented world components which in our case are displayed on a large (and possibly surround-) display and augmenting device, respectively, whose displays are carefully coordinated in a controlled manner.

3.1 Real World

Three classes of parameters affect the real world: *scene objects*, *atmosphere*, and *immersion* (cf. partial taxonomy for a specific scenario in Figure 3).

Scene objects represent the real-world objects in the environment (e.g. trees, buildings). Each object in the scene has parameters, such as their position and orientation and any properties of physics (e.g. velocity, acceleration).

Atmosphere refers to the weather and lighting conditions in the world. One can control weather (sunny, rainy, foggy, etc.), lighting (position and intensity of the sun and other light sources or reflections), and external visibility factors (e.g., smoke or debris), among others. For example, an AR application that would help a firefighter keep track of other firefighter locations could be evaluated with high levels of simulated smoke.

Real World Immersion refers to the perceptual fidelity parameters of the simulation such as display size (influencing field of regard), stereo capability, or display resolution [3]. While replicating the real world as closely as possible, it is important to be able to control immersion parameters to better understand the testing conditions of the augmented world. When simulating MR training systems such as flight simulators or other equipment operation trainers [4], it becomes important to systematically evaluate the influence of immersion parameters for the representation of the real world backdrop.

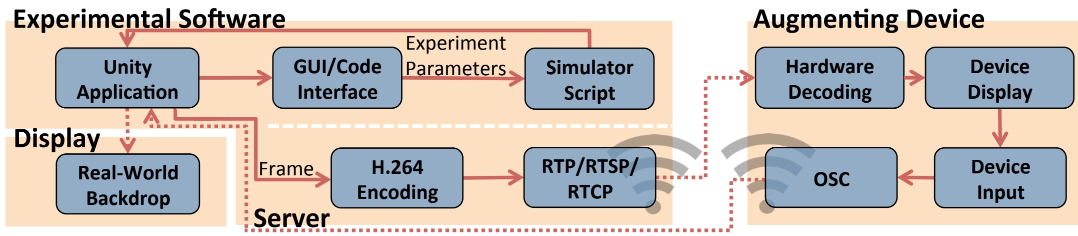

Figure 4: Mixed reality simulator architecture.

3.2 Augmented World

The primary parameters in the augmented world are: *immersion*, as well as *augmentations* and *interaction*.

Augmented World Immersion. By controlling the components of immersion for the augmented world, an experimenter may try different immersive configurations, often without needing extra equipment. For example, in simulation one can easily change the field of view of a simulated back-facing camera, or the frame rate of the simulated video stream.

Augmentations and Interaction are application specific. The designer or researcher will need to flexibly decide on the AR overlays to test the application with.

3.3 Devices

Currently, the most commonly used form factors for AR applications are video-see-through mobile devices (hand-held magic lenses or video streams in near-eye displays). The user of the mobile device may direct the camera lens toward some area of their environment, and a live stream of what the camera is capturing along with any augmentations can be viewed on the screen of the device.

Since a simulator is typically implemented indoors and does not require the user to physically travel long distances, the experimenter can use additional, often expensive or immobile, equipment to get more information about the user's experience. For instance, one might employ an eye tracking device to understand the regions of interest, an EEG device to better understand the cognitive workload required to use the interface, or other equipment that measures change in physiological activity. For studies in which mobility and ergonomics are crucial impact factors, such equipment should not be used.

A magic-lens based MR simulator has many advantages over HMD-based simulators. Body or head tracking is not needed in this environment as the user is intended to be at a far distance from what is being displayed on the large backdrop display, thus experiencing little to no parallax with movement. The head is not weighed down by heavy VR equipment, which could make head movements difficult or unrealistic. Real device affordances (e.g the weight, feel, and control elements of a particular smart phone or near-eye display) are taken into account, while retaining the flexibility of controlling immersion factors by simulating camera streams and overlays. The real body and limbs of the user may be fully observed without extra tracking equipment and reconstruction. Additionally, if one were to use extra equipment such as an eye tracker or EEG device, it might not be combined easily with a typical VR HMD system. For instance, the eye tracking device would need to be built into the HMD, and the EEG device may display harsh artifacts in the data due to the force of the HMD over the EEG electrodes, most likely resulting in unusable data.

4 Prototype Implementation

Our system is designed to be general and extensible. The implementation that we demonstrate here may guide others in constructing their own system for personal-device-based MR simulation.

Figure 5: Use of a smart phone as a stereo binocular display, using a VR2GO mobile viewer

There are four main elements of our implementation: the *backdrop display*, the *server*, the *augmenting device*, and the *experiment software*. A diagram of the system architecture is shown in Figure 4, an illustration of a use case is presented in Figure 2.

4.1 Backdrop Display

Our MR simulation implementation resides in a large spherical display which consists of two 5-meter-radius hemispheres with a 2-meter-wide bridge segment in between. Two Barco Galaxy NW-12 projectors with a combined 3575 x 1200 resolution were used. Our implementation has also used a single-projector wall display, and may be used with other displays such as CAVEs or cylindrical displays.

4.2 Server

The server was implemented on a machine running Lubuntu 13.04 with 2 8-core Intel Xeon CPUs, 2 Nvidia Quadro K5000 GPUs and 32GB of RAM. The Unity game engine was used to generate and run the virtual environment representing the "real-world backdrop" needed for a given task as well as the simulated augmentation layer. The server grabs each resulting video frame, at a resolution of 1024x526 pixels, of the composite view as seen by a virtual camera (controlled by tracking information from the augmenting device) in Unity. The frames are first encoded using a H.264 codec and then sent wirelessly over a 802.11n network to the augmenting device using RTSP, RTP and RTCP network protocols. By wirelessly streaming rendered video content instead of duplicating and synchronizing the virtual world on the mobile device, we achieve content independence and can render high quality 3D scenes without concerns of slowing the frame rate on the mobile display.

4.3 Augmenting Device

The purpose of the augmenting device is to simulate what the back-facing camera on the device would see on the real-world backdrop with added augmentations, but without actually using the camera.

(a) Medium visibility, high tracking jitter.

(b) Clear visibility, medium FOV.

(c) Low FOV.

Figure 6: Three implemented experiment scenarios: (a) a forest fire scenario, (b) a security team scenario in a city, and (c) an AR browser scenario in a park. The augmenting device view of each scene is displayed in the lower right corner.

While the mobile device does have to be prepared for low-latency display of wireless video streams, no content-dependent software needs to be installed on it. This enables the flexible use of (in our case Android) devices of different form factors (e.g. smart phones, tablets, Google glass).

For our reference implementation, a Samsung Galaxy S4 running Android Jellybean 4.2 was used as a magic-lens augmenting device, which can be used in hand-held mode as well as stereoscopic binocular mode using USC's VR2GO mobile viewer [17], cf. Figure 5. The augmenting device display content (backdrop plus augmentations) is streamed from the server and decoded with hardware acceleration provided by the Android SDK, to minimize the latency of decoding high quality and high resolution images. Users may view the scene on the device by positioning it so that it is directed toward the desired portion of the display. The orientation is calculated using a sensor fusion implementation for Android which is periodically corrected multiple times a second by a PhaseSpace Impulse X2 motion tracking system [14] that tracks phone-mounted infrared LEDs to correct for drift. This ensures that if the optical tracking is ever occluded, the device can still get a sense of orientation. Open Sound Control (OSC) packets are sent to the server wirelessly from both the mobile device and the optical tracking system, providing information about the orientation of the device as well as any interactions that took place on the touch screen, which then are immediately processed to alter the state of the virtual world in Unity. The orientation of the device controls a virtual camera in Unity, which streams its view of the virtual scene to the mobile device with an average of 95.75 ms of latency, as determined by high-speed camera observation. The latency is mainly due to the video streaming pipeline that occurs from server to augmenting device. Every frame must be encoded, packetized and transmitted, decoded, and rendered before being displayed on the augmenting device. Android employs triple buffering, and when combined with rendering to the augmenting device screen, may be the cause of approximately 50ms of latency in our system's pipeline. Use of the video-streaming approach rather than rendering over the actual camera feed feed enables us to simulate camera properties (such as FOV, resolution, exposure, or latency) in a controlled fashion, and eliminate refresh-rate interference artifacts with the backdrop projection.

4.4 Experiment Software

In our implementation, three different scenes (Figure 6) were designed to run simulated AR user studies. Scene or augmented objects may be manipulated by their position, orientation, and velocity. The augmented-world immersion parameters may be manipulated in terms of tracking jitter, tracking latency, and FOV. The real-world atmospheric conditions may be manipulated by the amount of visibility in a form that is specific to the scenario (e.g. smoke for the forest fire). While this is a very selective list of possible parameters, we feel that these parameters are significant to many experiments and can demonstrate the generality of the system.

Parameters are manipulated using an on-screen graphical user interface (GUI), or by interfacing with the scene by script (e.g. to automate randomized trials). We use Unity 3D along with C# Unity scripting in order to construct a framework that is modular and thus can be applied to any scene or scene object. The main simulator script, which handles all the parameters for the system, is composed of a list of methods that take a generic object (e.g. scene, atmosphere) as input and manipulates the object accordingly.

Virtual cameras, which are also considered objects, are devoted to the display of the real-world backdrop and the augmented view. Any necessary scene objects (e.g. virtual cameras, augmentations, or the main system object) are exported to what are known as 'prefabs' in Unity, which provides the ability to more easily import these objects into other scenes. Assuming that the experimenter has scenes and scene objects ready for use, the only setup required is to import the necessary prefabs and link the experiment parameters to objects of interest via code or our on-screen GUI.

5 Results

In this section, we first discuss a typical calibration procedure of our MR simulator and report the measured rotational error associated with our system's tracking, which shows how accurately it may be aligned with the real world display. Then, we describe three experiments we designed for use with our current prototype implementation, demonstrating the system's flexibility and ease of use. Lastly, we discuss current limitations of our MR simulation.

5.1 Calibration

To minimize image distortion, the aspect ratio and resolution of the Unity application that displays the real-world backdrop was set to the aspect ratio and resolution of our display environment. Additionally, the horizontal and vertical field of regard associated with the Unity-defined real-world backdrop display, must match the horizontal and vertical field of regard for the physical display system. In our case, these measurements were known ahead of time but may be obtained in a variety of ways, for instance using a Total Station.

5.1.1 Alignment

In order to have correct registration between the augmenting device and the real world display, the tracking system's coordinate system was aligned with the coordinate system of the virtual world (cf. Figure 7).

For the example scenarios driving our implementation, it wasn't our goal to exactly replicate the back-facing camera FOV of the Galaxy S4 for the augmenting device display, but instead to experiment with various camera FOVs. However, if the mobile display

	-70°	-60°	-50°	-40°	-30°	-20°	-10°	0°	10°	20°	30°	40°	50°	60°	70°
x_{err}	*	*	*	*	-0.4°	-0.1°	-0.1°	0.0°	-0.2°	-0.1°	-0.3°	*	*	*	*
y_{err}	-1.9°	-1.4°	-0.8°	-0.6°	-0.6°	-0.4°	-0.1°	0.0°	0.5°	0.9°	0.8°	-0.2°	-1.4°	-1.6°	-1.3°
z_{err}	*	*	*	*	0.2°	0.0°	0.0°	0.0°	0.0°	-0.1°	0.2°	*	*	*	*

Table 1: Rotation values and corresponding error as reported by the optical tracking system. Pitch was the rotation around the x axis, roll was the rotation around the z axis, and yaw was the rotation around the y axis. Entries containing an asterisk were not able to be measured, given the limits of the PTU device.

Figure 7: **Top image:** alignment of the augmenting device crosshair (green), with the real-world display crosshair (red). **Bottom image:** alignment of the back-facing camera crosshair (green), with the real-world display crosshair (red). The real-world display crosshair can also be seen in the background of the bottom image.

camera is to be simulated exactly, one must first obtain and match all intrinsic camera parameters and position the device's origin (reported from the tracker) where the back-facing camera is located.

5.1.2 Tracking Accuracy

To understand the inherent tracking limitations of a MR simulator tracking system, a comparison against ground truth is useful. We evaluated the rotation accuracy of our tracker with the help of a pan-tilt unit. The augmenting device and 4 rigid-body LEDs, were mounted onto a planar surface extending from a Directed Perception PTU-46-17.5. The PTU device was programmed to precisely rotate to a specific target angle along the x, y, and z axes of the tracking system, then the target angle was subtracted from the tracking system's measured angle to report error. Target angles and corresponding error are displayed in Table 1.

The largest per-axis rotations were constrained to not go beyond the full extent of the real-world display as well as the maximum range of the PTU. Overall, the results of this measurement show error of only a fraction of a degree. However, as seen in the extents of the y axis rotation, the error is approaching 1.5 to 2 degrees at the extreme points. Although larger, this amount of error is still hard to perceive by the naked eye, and it is not accumulating over time. It is most likely due to the tracked device reaching the angular limits of the tracking system, with few cameras seeing some infrared markers at those poses. Installation of more cameras and optimization of the tracking system is planned as a near-term update of our system.

5.2 Designed Experiments

The simulator is meant to be general and easily applied to a wide variety of scenarios and scenes. To demonstrate this, we designed three diverse scenario experiments to accompany the three scenes mentioned above. It took roughly 20 minutes to set up the experiment parameters with the objects of interest for each scene, preparing the simulator for user studies.

Scenario 1. A new AR tracking system is believed to help firefighters get a better spatial understanding of the position of helicopters in the air. These helicopters are intended to drop fire retardant, but need to be sure they are on target and that the ground is clear. Poor tracking registration may be particularly detrimental to this system, leading air traffic observers to have a skewed perception of helicopter position, particularly when air visibility is low. The simulator is set to have 4 conditions: low/high tracking jitter and latency, and low/high visibility (cf. Figure 3). Participants representing AR-equipped air traffic observers on an overlook are to draw flight trajectories on a map which displays the landscape from above. Sketches will be compared to the ground truth data to get a measure of task performance.

Scenario 2. AR technology is used to track members of a security team in city streets, to keep crowds under control at a festival. Use of the system allows the security team leader to keep accurate positions of all members, and provides the ability to respond to an incident efficiently. It is believed that, as more security members are deployed to control the crowd, the harder it is for the team leader to keep track of the team. Researchers want to get a better understanding of the demand on cognitive workload while performing this task, so that when the team leader's workload is at or near capacity, they know when to delegate responsibilities to other team members. The simulator is set with 3 conditions: small, medium, and large sized security teams, deployed around the festival randomly. Participants are allowed to view the scene for a predetermined amount of time and then are required to record as many of the tracked positions of the security team as possible. While the task is being performed, participants will wear an EEG recording device, which in real time classifies between cognitive states of low and high workload.

Scenario 3. An AR software development team is in the process of creating an AR browser for mobile devices, and would like the user experience to be easy and engaging. It is believed that a smaller FOV for the augmenting device, and high tracking latency may be detrimental to user performance during MR tasks. The team would like to find the optimal field of view as well as a threshold for the amount of latency that will provide an enjoyable experience. The team set up a simulated AR browser application that displays augmented icons over objects of interest in the scene. When a user points the device toward one of the icons, a dialog displays more information about the object of interest. For instance, an icon over a bench in a park may display the year it was built and to whom it may be dedicated. Subjects are presented with questions about the scene, ranging from easy to difficult, which they must answer by inspecting the icons. A between-subjects study design explores settings of small/medium/large FOV, and 100ms/300ms/500ms tracking latency. The user will be fitted with a mobile eye tracker to detect when eye fixations are away from the mobile device screen, as well as an EEG device to record real-time cognitive load information. The combination of task performance, off-screen eye fixations, and measures of cognitive load will help determine what type of FOV system is most useful in the presence of tracking latency.

5.3 Limitations

While the MR simulator makes most experimental variables easy to implement, some difficulties remain. Multicast server configuration to support multiple users, and stereoscopic rendering for the display backdrop are not currently part of the implementation, but may be included in the future. The MR simulator studies we have run so far were focused on inside-out wide-area AR, where most of the backdrop content surrounds the user at some distance. Indoor scenarios with many nearby objects may be more difficult to simulate realistically with this setup (without stereo). Many of our current setups rely mainly on orientation tracking, but our surround display allows for some movement in between the display hemispheres. Evaluating position tracking performance is left for future work. Lighting is one of the trickier, yet important [11, 7], variables to simulate given that it is very hard to get an accurate representation of real outdoor lighting in a simulated display environment. Similarly, as with all types of software-based simulation, the resolution of the display is not an accurate representation of how we perceive the resolution of the real world. The MR simulator has a certain base latency due to the transmission of high resolution video, making it difficult to simulate augmenting device displays with little to no latency. Considering that there is no latency to view the real world backdrop in our implementation (because we don't employ head-tracking and thus the backdrop doesn't change based on observer motion), this setup differs from HMD simulators [9, 11] that introduce latency on the real-world backdrop display. The 95ms latency would most practically be helped somewhat with a new video encoding standard, such as H.265 which is currently in the process of becoming mainstream. Hardware may be simulated, e.g. speed of processing in the form of display frame rate and device response time, however, simulating all aspects of hardware is difficult given the variety of form factors and specifications available. For example, it would be non-trivial for a single augmenting device to simulate auto-stereo displays, large displays, or wearable displays. Still, as mentioned earlier, many of these types of devices are equipped with an Android operating system and instead of being simulated these devices can directly be swapped in to the simulator.

6 Conclusion

In this paper, we introduced the design and prototype implementation of a system for simulating MR with large backdrop displays and controlled mobile AR devices. The MR simulation approach provides the ability to evaluate AR user interface design by offering full control over AR user experiments and application design. Additionally, it allows designers and engineers to implement novel systems and interfaces, while not being dependent on the shortcomings of current technology.

Acknowledgements

This work was partially supported by ONR grants N00014-14-1-0133 and N00014-13-1-0872, as well as NSF grant IIS-0747520.

References

[1] J. Arthur, L. Prinzel, K. Shelton, L. J. Kramer, S. P. Williams, R. E. Bailey, and R. M. Norman. Design and testing of an unlimited field-of-regard synthetic vision head-worn display for commercial aircraft surface operations. In *Proceedings of SPIE, Enhanced and Synthetic Vision*, Apr. 2007. Paper 12.

[2] E. Bier, M. Stone, K. Pier, W. Buxton, and T. DeRose. Toolglass and magic lenses: The see-through interface. In *Proc. SIGGRAPH*. ACM, 1993.

[3] D. Bowman and R. McMahan. Virtual reality: How much immersion is enough? *Computer*, 40:36–43, 2006.

[4] D. A. Bowman, C. Stinson, E. D. Ragan, S. Scerbo, T. Höllerer, C. Lee, R. P. McMahan, and R. Kopper. Evaluating effectiveness in virtual environments with MR simulation. In *Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC)*, pages 12075–1 – 12075–11, Orlando, FL, 2012.

[5] L. Brown and H. Hua. Magic lenses for augmented virtual environments. *IEEE Computer Graphics and Applications*, 26(4):64–73, 2006.

[6] K. Cheng, J. Li, and C. Müller-Tomfelde. Supporting interaction and collaboration on large displays using tablet devices. In *Proc. AVI*. ACM, 2012.

[7] J. Gabbard, J. E. Swan, and D. Hix. The effects of text drawing styles, background textures, and natural lighting on text legibility in outdoor augmented reality. *Presence: Teleoper. Virtual Environ.*, 15:16–32, 2006.

[8] J. Grubert, H. Seichter, and D. Schmalstieg. Towards user perspective augmented reality for public displays. In *Proc. ISMAR*. IEEE, 2014.

[9] C. Lee, S. Bonebrake, T. Höllerer, and D. A. Bowman. The role of latency in the validity of AR simulation. In *Proc. VR*. IEEE Computer Society, 2010.

[10] C. Lee, S. DiVerdi, and T. Höllerer. An immaterial depth-fused 3d display. In *Proceedings of the 2007 ACM Symposium on Virtual Reality Software and Technology*, VRST '07, pages 191–198, New York, NY, USA, 2007. ACM.

[11] C. Lee, A. G. Rincon, G. Meyer, T. Höllerer, and D. A. Bowman. The effects of visual realism on search tasks in mixed reality simulation. *IEEE Computer Graphics and Applications*, 19:547–556, 2013.

[12] S. Liu, D. Cheng, and H. Hua. An optical see-through head mounted display with addressable focal planes. In *Mixed and Augmented Reality, 2008. ISMAR 2008. 7th IEEE/ACM International Symposium on*, pages 33–42, Sept 2008.

[13] M. Ostkamp and C. Kray. Prototyping mobile AR in immersive video environments. In *Proc. MobileHCI*. ACM, 2013.

[14] Phasespace impulse x2 motion tracking system. http://www.phasespace.com/impulse-motion-capture.html, accessed September 2014.

[15] A. State, K. P. Keller, and H. Fuchs. Simulation-based design and rapid prototyping of a parallax-free, orthoscopic video see-through head-mounted display. In *Proceedings of the 4th IEEE/ACM International Symposium on Mixed and Augmented Reality*, ISMAR '05, pages 28–31, Washington, DC, USA, 2005. IEEE Computer Society.

[16] Z. Szalavari and M. Gervautz. The personal interaction panel - a two-handed interface for augmented reality. In *Computer Graphics Forum*, 1997.

[17] VR2GO mobile viewer, MxR, USC Institute for Creative Technologies. http://projects.ict.usc.edu/mxr/diy/vr2go/, accessed September 2014.