Simulating Illumination in Mixed Reality - A Cross Reality Application

Jeremy Bitar
bitarj@ethz.ch

Simon Ebner
ebners@ethz.ch

Laura Nydegger
nylaura@ethz.ch

Abstract

This project presents a Cross Reality (XR) application that combines Mixed Reality (MR) and Virtual Reality (VR) for simulating lighting in real-time. Users can design and visualize lighting by placing objects and light sources in an MR environment and viewing the effects in a virtual replica of the same space. The system uses automated room scanning and dynamic lighting, offering a tool that can be used for interior design and planning illumination of buildings. A user study highlighted its potential while suggesting improvements to the interface and functionality.

1. Introduction

Real-time simulations of lighting in mixed reality have many applications. For example it can be used in interior design to simulate how furniture, decorations and light sources might look depending on the illumination, for mixed reality gaming and entertainment, where realistic light effects can heighten immersion or industrial design where designers could view their product under various lighting environments before moving to production.
Cross Reality (XR) is a term that is often used to describe applications that utilize a combination of both Mixed Reality (MR) and Virtual Reality (VR) [13]. Applications like these take advantage of overlays to extend the physical space but also provide the user access to virtual environments that grant experiences that are hard to achieve in Mixed Reality.
We present a novel approach to such lighting simulations where we provide a prototype application for the Meta Quest 3 that lets users switch between Mixed Reality and a completely virtual reconstruction of the same room. The user can place objects such as tables, lamps and other light sources in the mixed reality environment and simulate the illumination effects in the virtual copy where one can also change what time of day it should simulate. The benefits of a system like this is that you are able to place objects and lights easier in mixed reality, and by simulating the effects only in virtual reality, one does not have to consider conflicting existing light sources.

Figure 1. Switching between mixed and virtual reality

2. Related Work

2.1. Cross Reality Applications

In recent years, an increasing number of use-cases for cross reality functionality have emerged. It was used to create virtual co-working spaces, that combine physical and virtual space to facilitate easier collaboration for remote work.[9],using online laboratories to support students during their learning processes [5] or enhancing assembly operations for workers by using digital copies of machines. [10] The use of cross reality functionalities for virtual furnishing and lighting is largely unexplored. Most of the work done in that area of research is focusing on either mixed or virtual reality separately.

2.2. Furniture Placement

Placing objects such as furniture in virtual or mixed reality has been explored in various ways. Early solutions focused primarily on solutions in virtual reality by creating virtual showrooms where the objects could be previewed before making a purchase [7, 12]. Newer solutions primarily use mixed reality, where users can visualize objects directly in their current environment. This helps in estimating the scale and dimensions of an object. [8, 11]

2.3. Light Simulation in Mixed Reality

Illumination in mixed reality can be understood in two ways. Either applying the environment light to new virtual objects, or capturing the lighting effects of new virtual light sources and applying them to the current environment. To capture an estimation of the current environment illumination, most newer solutions use machine learning models such as CNNs to gain an estimate that can then be applied to virtual objects so that the models blend in seamlessly. [4]. Applying the light from virtual sources requires a virtual copy of all affected objects. You would then be able to calculate the light on that copy and apply the difference to the scene.[1]

3. Method

In order for the user to switch between MR and VR, a virtual copy of the room needs to be created and synchronized with the real version of the room. To achieve this, the room is scanned to receive information on the layout of walls and objects, as well as some labels pertaining to the nature of the objects present in the room. Then, a virtual version of the room is created by automatically placing wall and object models. Furthermore, windows are cut into the walls to allow simulation of sunlight.
Once the room is scanned, the user can freely switch between MR and VR and interact with the environment by placing objects and changing time of day in VR.

3.1. Room Scan

To receive information on the room layout, we use Meta's automatic Space Setup feature. The advantage of this is that the room information can be generated in a few minutes, enabling the application to be portable to any room. The Space Setup can be found under "Settings" > "Environment Setup" on the Meta Quest 3. Alternatively, the user will be prompted to scan the room if the application is opened without having done a Space Setup beforehand. In the application itself, we then use Meta's MR Utility Kit (MRUK)[6] to handle the information gathered during the Space Setup as well as the synchronization between the virtual representation and the real room. From the scan our app receives information on the location and size of several types of objects, labeled with one of the tags {floor, ceiling, wall, table, couch, door, window, storage, bed, screen, lamp, plant, wall-art, other}.

3.2. VR Representation

For the creation of the VR environment we replace the boxes given by the scan with appropriate object models. Again, there is a script in the MRUK for this purpose: AnchorPrefabSpawner. For a given label, it selects one prefab out of a list of prefabs provided by the user, according to the selected heuristic. However, this offers only two selection modes: "Random" and "Closest Size". We found that closest size selection is meant to select the model that is closest in volume to the target box, however in the current version of MRUK this function is bugged and does simply return the first object in the list. Furthermore, we found that simply selecting by volume didn't give satisfactory results as it does not account for the shape of the object. For these reasons we added another heuristic accounting for the ratio between depth and width as well as base area and height.

$$baser(obj) = \min\left(\frac{obj.x}{obj.z}, \frac{obj.z}{obj.x}\right) \tag{1}$$

$$heightr(obj) = \frac{obj.y}{\sqrt{obj.x \cdot obj.z}} \tag{2}$$

We then select the prefab p that has the closest ratio to the anchor a given by the room scan.

$$\Delta base(p) = |baser(p) - baser(a)| \tag{3}$$

$$\Delta height(p) = |heightr(p) - heightr(a)| \tag{4}$$

$$\min_{p}(\Delta base(p) + \Delta height(p)) \tag{5}$$

Using this heuristic, we found the selected objects to fit better into the anchors than before, resulting in less distortion and more convincing replacements. However, due to the lack of customizable labels, not all objects can be given a fitting prefab, as we can't distinguish two objects with the label "other" that have the same size.

3.3. Cutting Windows

Meta's room scan merely supports detecting anchors for walls and windows but does not directly offer them as a joined 3D object. If we want such a thing we will have to aggregate the models ourselves.
For this we use the scan's hierarchical structure, querying the child components that are automatically attached to a wall. Whenever Meta tells us that it detected a wall anchor we, as for all the other objects, spawn a prefab. Unlike the rest, this prefab has the WindowCutter script attached, which iterates all child anchors recognized as windows and subtracts the previously spawned geometry from that of the wall. While this is happening, the geometry of the window itself is removed since we are exclusively using it to manipulate wall-geometry.
Unity does not directly support boolean operations on 3D

objects. As to not re-invent the wheel, we employ the pb_CSG library [2]. Iterating all the windows in a wall, we subtract the geometry of one window after the other from the wall as demonstrated in the example provided in the README of the library.

3.4. Virtual Skybox

To simulate the lighting corresponding to a certain time of day we use Unity's skybox system. For this we set the camera background to "Skybox" while in VR. We can configure the finer details of the project's lighting in the respective window ("Window" > "Rendering" > "Lighting"). Unity will then render the sun based on the rotation of the specified primary directional light source of the scene. By manipulating the respective x-rotation (east-west) of the aforementioned light we can thus change the time of day displayed in the scene.
In order to handle the conversion between time of day (TimeSpan) and the Quaternion data type used by Unity (and many other engines) for rotations we add the TimeManager script to the directional light. It starts by storing the initial 3-axis rotation (euler angles) and determining the corresponding TimeSpan. When we want to manipulate the time of the scene we change the x-value of the stored rotation and re-apply the quaternion equivalent to the Transform. This needs to be done to avoid the gimbal lock that occurs when converting back and forth between eulers and quaternions.

3.5. UI and Controls

For completeness, we briefly describe the UI and actions the user can take in the application.

UI The UI consists of a 3 part menu. It contains a basic info panel, options for object placement and for the changing of time of day.

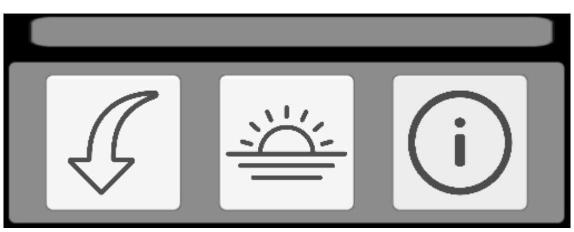

Figure 2. Main UI with buttons for object placement, time changing, and info

Time of Day Controlling the time of day is done by adjusting the value of a slider which controls the x-value of the sun rotation as described in 3.4.

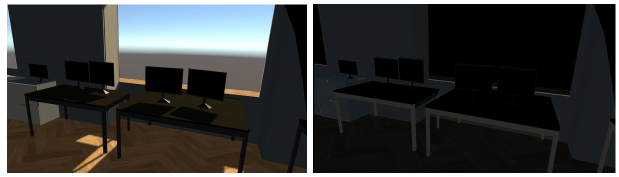

Figure 3. Switching between day and night time in VR

Object Placement The user can chose between different objects and place them at any desired point in the scene. Additionally, the objects can be scaled and rotated. Once placed, the user can either delete a specific object or clear the scene of all placed objects.

Figure 4. Placing a table in MR

Switch Between VR and MR Lastly, with the press of a controller button, the application switches between the MR and VR representation of the room.

4. User Study

To evaluate the effectiveness and intuitiveness of our prototype application, we conducted a usability study. The Participants consisted of 10 computer science students aged from 20 to 27 with varying levels of experience with mixed reality environments.

Experiment Setup All participants have received a short introduction to explain the purpose of the application and what functionalities are available.
Then they were asked to solve 5 different tasks that tested all functionalities of our application. We provided no assistance during the process of solving these tasks apart from an explanation of the controls that they could access at all times. This helped us in evaluating how intuitive the controls and user-interface are. We measured the time it took for each task to be completed. (see table 1)
After completing these tasks, all participants had a 3 minute time window where they could freely explore and test the app. In this window we offer assistance in case anything was still unclear. For further evaluation we asked each participant to fill out a questionnaire about their experience.

UI Intuitiveness The measured times per task and the perceived intuitiveness from the questionnaire seem to paint

Task completion times		
Task description	Average Time	Median Time
Switch from Mixed to Virtual Reality and back.	4.22s	3.03s
Place a pointlight and a lamp and check the effects in VR	68.53s	45:30s
Place a scaled up and rotated table	30.10s	20:40s
Delete the previously placed table	5.30s	4:40s
Change time of day to nighttime in VR.	8.37s	7:30s

Table 1. Measured times to complete each task

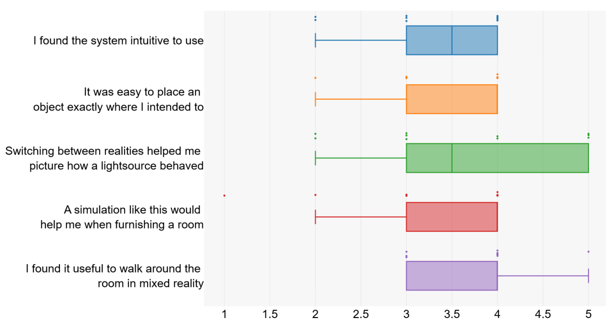

Figure 5. Questionnaire Answers

a conflicting image. Most participants reported that they found the system easy to use but the completion times for task 2 and 3 are comparatively long. One of the reasons for that becomes obvious when looking at the difference between median and average time. A small amount of participants encountered an issue that stalled them for a longer time and led to some significant outliers. The participants noted in particular that the buttons did not provide enough feedback, and that the controls were not completely obvious. In the free exploration part of the study, every participant overcame the issues that impaired them. So it seems that even though there are some hindrances, the user experience improves after a becoming familiar with the application.

User opinions At the end of the study we asked all participants a few open-ended questions on their experience. Whether they encountered problems or if they feel like important features are still missing. The shared sentiment across users was that the application served as a good prototype that can still be expanded upon.

> ”I could see myself use an application like this, if it had more options of objects and lights to place” - Participant # 3

Another Participant mentioned the wish for simulating lighting effects in mixed reality aswell.

> ”I think simulating the light in mixed reality aswell as virtual reality would be benficial” - Participant # 6

5. Outlook

As discussed in 4 the UI was perceived as somewhat unintuitive. An overhaul of the UI and controls to make them more user friendly should therefore be considered in the future. Areas to improve could include adding graphics to indicate the currently active buttons, slowing down the turning and scaling speed of objects and making the UI overall less disruptive on the screen.
Furthermore, the application could be extended with more functionality and be made more customizable. For example by adding more objects to place and allowing adjustment of objects already placed by the user. This could also include enabling the user to replace automatically placed objects in case they don’t align with their real counterpart.
Additionally the outlook from the VR representation could be expanded on with scenery objects outside the room. At the moment the virtual exterior is a completely barren endless void.
Lastly it could be interesting to explore other options for the underlying engine for different light simulation and performance trade-offs.

As of writing this there is a problem with the geometry library [2] which we use for window cutting. After boolean operations have been performed the resulting geometry has its pivot at the world origin. Consequently transformations don’t behave as they should, leading to accumulating errors with distance from the origin. Since neither the library nor Unity supports setting the pivot the problem would have to be fixed by switching geometry library or using another library to set the pivot afterwards. For the latter approach we already have a working prototype using Unity’s ProBuilder package but did not have the chance to test it on the device (see figure 6).

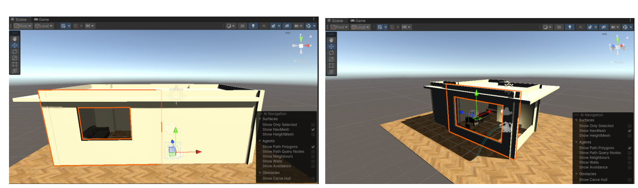

Figure 6. Pivot at world origin (left) and Pivot at wall-center (right)

6. Conclusion

The created application is a functional prototype and contains all necessary building blocks for the use case we intended to cover and shows a lot of potential that can be expanded upon. Future iterations could incorporate a larger catalog of objects, more customization and more realistic illumination.

The main motivation for using a cross reality approach was to simulate lighting conditions. However the dynamic lighting capability of Unity is limited in its realism. A different engine with more realistic dynamic lighting might be beneficial.

The user study provided valuable insights, revealing both the system's strengths and areas for improvement. Participants appreciated the smooth transitioning between MR and VR, as well as the flexibility that the object placement offered. However, the feedback indicated a need for a more intuitive user interface and enhanced visual feedback during interactions.

The repository with the full source code of the application can be found at [3].

References

[1] Stephen DiVerdi and Tobias Höllerer. Combining dynamic physical and virtual illumination in augmented reality. *Environment*, 5:12, 2004. 2

[2] Karl Henkel. pb_csg. `https://github.com/karl-/pb_CSG`, 2015. 3, 4

[3] Laura Nydegger Jeremy Bitar, Simon Ebner. Cross reality application. `https://github.com/jeremj22/cross-reality-app`, 2025. Accessed: January 6, 2025. 5

[4] David Mandl, Kwang Moo Yi, Peter Mohr, Peter M Roth, Pascal Fua, Vincent Lepetit, Dieter Schmalstieg, and Denis Kalkofen. Learning lightprobes for mixed reality illumination. In *2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 82–89. IEEE, 2017. 2

[5] Dominik May. Cross reality spaces in engineering education–online laboratories for supporting international student collaboration in merging realities. 2020. 1

[6] Meta. Mixed reality utility kit. `https://developers.meta.com/horizon/documentation/unity/unity-mr-utility-kit-overview`. Accessed: January 6, 2025. 2

[7] Hyunjoo Oh, So-Yeon Yoon, and Jana Hawley. What virtual reality can offer to the furniture industry. *Journal of Textile and Apparel, Technology and Management*, 4(1):1–17, 2004. 2

[8] Selcen Ozturkcan. Service innovation: Using augmented reality in the ikea place app. *Journal of Information Technology Teaching Cases*, 11(1):8–13, 2021. 2

[9] Derek F Reilly, Hafez Rouzati, Andy Wu, Jee Yeon Hwang, Jeremy Brudvik, and W Keith Edwards. Twinspace: an infrastructure for cross-reality team spaces. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 119–128, 2010. 1

[10] Bruno Simões, Raffaele De Amicis, Iñigo Barandiaran, and Jorge Posada. Cross reality to enhance worker cognition in industrial assembly operations. *The International Journal of Advanced Manufacturing Technology*, 105(9):3965–3978, 2019. 1

[11] Waraporn Viyanon, Thanadon Songsuittipong, Phattarika Piyapaisarn, and Suwanun Sudchid. Ar furniture: Integrating augmented reality technology to enhance interior design using marker and markerless tracking. In *Proceedings of the 2nd international conference on intelligent information processing*, pages 1–7, 2017. 2

[12] So-Yeon Yoon, Yun Jung Choi, and Hyunjoo Oh. User attributes in processing 3d vr-enabled showroom: Gender, visual cognitive styles, and the sense of presence. *International Journal of Human-Computer Studies*, 82:1–10, 2015. 2

[13] Cindy Ziker, Barbara Truman, and Heather Dodds. Cross reality (xr): Challenges and opportunities across the spectrum. *Innovative learning environments in STEM higher education: Opportunities, challenges, and looking forward*, pages 55–77, 2021. 1