

Using Hand Tracking in Mixed Reality to Reproduce Optical Systems for Mid-air Images

LUCA BARRE^{1,a)} RYUGO HAYASHI^{2,b)} KOIZUMI NAOYA^{2,c)}

Abstract: To develop optical systems for mid-air images, several methods have been proposed that reproduce them in an application simulating their behavior. However, most of these methods require time and lack practicability in the real environment. Therefore, we propose a method that can simulate mid-air images and the necessary optical system in these potential environments using a combination of hand tracking and mixed reality. We are contributing to the development of interacting with and creating optical systems for mid-air images by providing an intuitive way to prove new ideas.

1. Introduction

Mid-air imaging is a promising way of combining virtual world with the real world by providing a new type of displaying images. These images are not shown on a monitor or on a wall but rather produced into the air by an optical system. This allows the user to see and interact with the image without having to wear any equipment like a Head-mounted Display (HMD).

However, mid-air imaging faces two major problems. First, the optical system consists of at least one mirror with very specific properties that influence the way of reflecting and refracting light. To achieve these specific properties, a special manufacturing process is necessary that comes with high costs and long construction times. Second, some optical systems only exist in theory as there is no material in the real world, that has the properties necessary to create that desired optical system. As a result, developing new optical systems and new ways of interacting with mid-air images or simply installing an optical system in a certain environment is not practical.

To deal with the latter, Mixed Reality applications can help to overcome that problem. Mixed Reality (MR) is a technology similar to Virtual Reality (VR) and Augmented Reality (AR) that combines the real world with an application. While VR applications are replacing the real world with a virtual world, restricting interaction to the virtual one, AR applications enhance the real world by overlaying it with virtual elements, allowing the interaction with both worlds. Similarly, MR applications also allow the user to

interact with both the real and the virtual world and can enhance the real world with virtual elements. However, with MR, the user can use real objects to interact with the virtual world by anchoring them to virtual elements. For example, they can use their hands using hand tracking or their water bottle as a wand to move virtual objects.

This brings new possibilities in designing application in different fields like entertainment, industry and research. For example, it makes proving the practicability of a system in their intended environment easier, as the user can simulate their application in that environment using MR instead of creating a simulation in their laboratory. Therefore, it is also suitable for designing optical system for mid-air images. Designing these optical systems is still not practical due to the high costs and the absence of the materials with the necessary properties. However, using a technology to design these optical systems without the drawbacks of creation while also being able to apply them "directly" in the real world, is very useful. Moreover, mid-air imaging is still not a widely used projection technique and not very well known. Thus, most people can hardly imagine how installed optical systems look like, what their scaling are and how high the costs would be. Therefore, being able to apply these optical systems in a potential environment using a MR application can help people to understand mid-air imaging better.

In this paper, we propose a method to create, scale, position and set the parameters like the distance and view area of mid-air images and the optical systems by using hand tracking in real-time. This allows the user to virtually create and adjust their desired mid-air imaging system at their intended location in the real world, using an intuitive interaction method, so that that they proof whether it is feasible in that environment before acquiring the real system.

¹ University of Bremen

² University of Electro-Communications

a) luca.barre16@gmail.com

b) hayashi@media.lab.uec.ac.jp

c) koizumi.naoya@uec.ac.jp

2. Related Work

For mid-air imaging various optical systems exist, that generate the images in different ways which result in different properties of the mid-air images. One of these systems are called Micro-Mirror Array Plates (MMAP) [1]. As MMAPs can generate mid-air images on their own, they only rely on the properties of the light source and allow the easy creation of mid-air imaging environment due to their simple configuration. Compared to other optical systems it can generate bright mid-air images, resulting in a good visibility. However, the viewing angle of the mid-air image is narrow, so it can only be seen from limited positions. The system by Maekawa, S. et al. is using a Dihedral Corner Reflector Arrays (DCRA) [2] which is working similar to MMAPs and can form reflection-symmetric mid-air image without any distortion. However, this device still has some drawbacks like the low resolution of the image.

There are also systems that make use of retro-reflection such as the systems Yamamoto, H. et al. [3] and Koike, T. et al. [4] describe in their respective papers. These systems can be summarized as Aerial Imaging by Retro Reflector (AIRR). Mid-air Images created by AIRR systems can be seen by a wide viewing angle for up to 90 degrees and they are visible at room light. However, they are not as bright as the mid-air images from MMAPs and also require multiple parts to be set up correctly, resulting in harder configurations.

Because of the simple configuration compared to other mid-air imaging systems and their ability to generate mid-air images on their own, we decided to use MMAPs for the optical systems in our proposed method.

To deal with the problems that the optical systems for mid-air images like the MMAP share, several methods to use computer application for simulating mid-air imaging have been proposed. In their paper Kiuchi et al. proposed a method to model a MMAP in order to simulate the appearance of a mid-air image [5]. To model the MMAP they were using the software Blender and to render the mid-air images they were using Blender's physically based production renderer Cycles. With their method, they were able to correctly reproduce the mid-air image created by a real MMAP. While their method is able to reproduce MMAP physically correctly, their rendering process takes time, making it not usable for real-time applications. Hayashi, R. et al. have proposed a method to reproduce and simulate the behavior of a MMAP using the Unity game engine [6]. While their approach is able to reproduce the behavior of a MMAP in real-time, it is only used in VR, leaving out relations to the real world such as size compared to the environment. Our system however, can simulate MMAPs in real-time and in MR while also providing an intuitive operation with hand tracking.

For proving the benefits of using VR, AR and MR applications to simulate real world behaviors and to develop new

ways of interacting with them, several methods have been proposed. With VRception Gruenefeld, U. et al. have proposed a concept to prototype cross-reality systems [7]. Their concept allows to easily prototype VR, AR and Augmented Reality applications in a VR environment by using either the Unity Engine or their own predefined editor. Therefore, they can overcome technical limitations as all the development takes place inside the VR application.

3. Proposed Method

Our proposed method is using Hand tracking to allow the user to create the desired mid-air image and the necessary optical system in mixed reality.

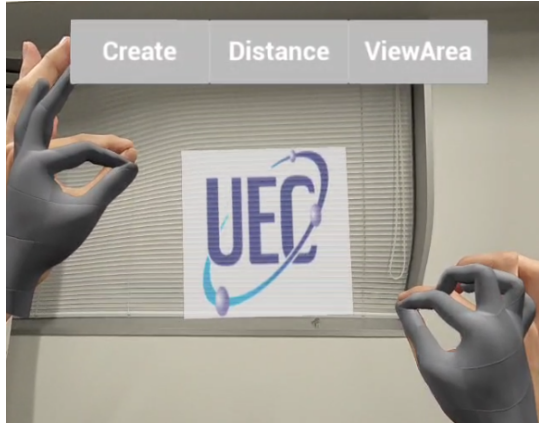
3.1 Interaction

When using our application, the user will find themselves in a mixed reality world which means that they can see and interact with both the real and the virtual world. At the top border of their view port is a menu bar that allows the user to either create a mid-air image, adjust the distance from the mid-air image to the MMAP or adjust the viewing area in which the mid-air image can be seen. (see Figure 1).

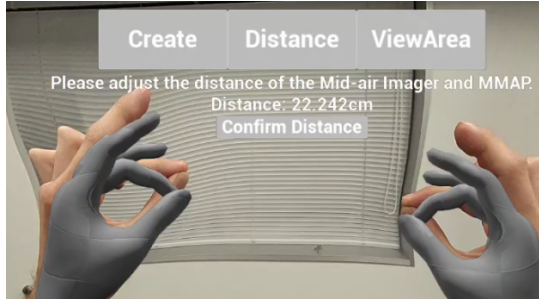
When clicking on the "Create"-button, the user can create a mid-air image by forming and holding a "index-pinch" gesture with their left hand first and then their right hand (see Figure 1a). When shifting the positions of their hands, the user can also change the position and size of the mid-air image. However, the mid-air image will always face the user during that process. When the user has moved the mid-air image to desired position and scaled it to their desired size, they can release their right pinch gesture to set the mid-air image in its place and stop the creation. Now, the necessary optical system will be created behind the mid-air image according to the procedure described in subsection 3.2.

When they are clicking on the "Distance"-button a dialog below the menu bar appears as seen in Figure 1b. By forming "middle-pinch"-gestures with both hands and change the distance between them, the user can now adjust the distance between the mid-air image and the MMAP. By clicking on the "Confirm Distance"-button they can confirm the distance. If the user has not created a MMAP yet, the distance is changed for future MMAPs. If the user has created a MMAP, the distance of future MMAPs and the current MMAP will be changed. The current one will be repositioned and resized accordingly.

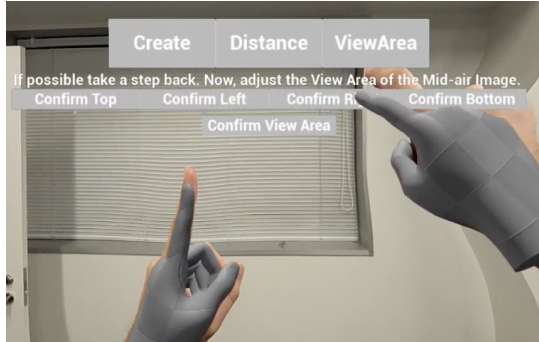
When the user clicks on the "View Area"-button the dialog seen in Figure 1c appears. This allows them to adjust the view area in which the mid-air image can be seen. By moving the finger tip of their left index finger into position, the user can click on the "Confirm Top", "Confirm Left", "Confirm Right" and "Confirm Bottom" -buttons with their right index finger to specify the top, left, right and bottom border of the view area respectively. Clicking on the "Confirm View Area"-button confirms the view area and the MMAP will be resized accordingly. The user can now repeat either of these steps to adjust the optical system to their desires.



(a) A mid-air image can be create by forming index-pinch gestures with both hands



(b) The distance can be adjusted by forming middle-pinch gestures with both hands



(c) The view area can be adjusted by moving the left index and confirming its position as the respective view area border with the right index finger

Fig. 1: An overview of the interaction methods

3.2 Implementation

To run our proposed method we are using the Meta Quest 3. It provides Hand tracking and provides the "Passthrough" feature. This allows the user to see their environment through the HMD, allowing for the mixed reality experience. To implement the application, we are using the Unreal Engine 5.2^{*1} as it allows us to create MR applications easily and has a good integration with common HMDs such as the Meta Quest.

Table 1: Overview of the parameters and their description used for the scaling calculation of MMAP

Parameter	Description
$Direction$	Direction from mid-air image to place the MMAP
Pos_{MidAir}	Position of mid-air image in world space
Pos_{Eye}	Position of Camera in world space
Pos_{MMAP}	Position of MMAP in world space
$RightVector_{MidAir}$	Local Y-Axis of Mid-air Image in world space
$YScale_{MidAir}$	Scaling of Mid-air Image on local Y-Axis
$UpVector_{MidAir}$	Local Z-Axis of Mid-air Image in world space
Pos_{Corner}	Position of one mid-air image corner in world space

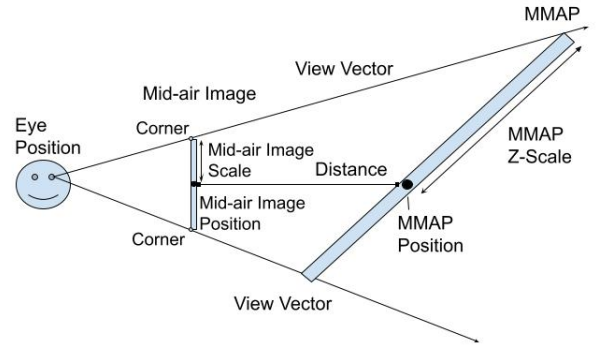


Fig. 2: The overview of calculating the size and location of the mid-air image, MMAP and light source

When the user has either decided the position and size of the mid-air image, has changed the distance between the mid-air image and the MMAP or has adjusted the view area in which the mid-air image can be seen, the MMAP has to be created at the correct position, with correct size and rotation. An overview of this process can be seen in figure 2. The parameters that are given after creating a mid-air image, are the camera position, the mid-air image position and the mid-air image scaling. An overview of the parameters we use is given in Table 1. As the MMAP needs to be created behind the mid-air image with respect to the camera position, we can use the following equations to calculate the position of the MMAP:

$$Direction = \frac{Pos_{MidAir} - Pos_{Eye}}{\|Pos_{MidAir} - Pos_{Eye}\|} \quad (1)$$

$$Pos_{MMAP} = (Direction * Distance) + Pos_{MidAir} \quad (2)$$

The first equation returns the normalized direction vector in which we want to place the MMAP. The second equation calculates the position on the vector from the position of the mid-air image. For the rotation of the MMAP, we are using the rotation of the mid-air image and add a negative 45 degree rotation to the local Y-Axis. This gives us a correct MMAP rotation for every mid-air image. To calculate the scaling of the MMAP, we proceed with the following steps. First, we calculate the corner points of the mid-air image and use the following equations:

*1 <https://www.unrealengine.com/en-US/unreal-engine-5>

$$Y = RightVector_{MidAir} * YScale_{MidAir} \quad (3)$$

$$Z = UpVector_{MidAir} * ZScale_{MidAir} \quad (4)$$

$$Pos_{Corner} = Pos_{MidAir} + Y + Z \quad (5)$$

Depending on which corner point we are calculating, also have to multiply Y or Z with -1 respectively. Second, for each corner point we calculate a vector that starts from the camera position and passes through the corner point. For that we use the following equation:

$$ViewVector = Pos_{Corner} - Pos_{Eye} \quad (6)$$

Third, we calculate the intersection of each vector with the plane the MMAP position and its forward vector create. For this, we use Unreal's built-in "Line-Plane-Intersection"-node to which we pass the camera position, the vector from the previous step multiplied by a length, the position of the MMAP as the plane origin and the forward vector of the MMAP as the plane normal. As the length needs to be long enough to allow big scaled MMAPs, we decided to go for a length of 100 m. Fourth, we calculate the differences in the y and z coordinates of the intersection points and the MMAP position as they will serve as the MMAP scaling. To do so, we decided to use Unreal's built-in "Inverse Transformation"-node. This transform a given location from world space to the local space using a given transformation. So, we pass the intersection points and the transformation of the MMAP to this node. Then, we use the y- and z-coordinates in local space as our potential scaling for the MMAP. Fifth, we compare the four potential scaling for y- and the four potential scaling for the z-coordinates and choose the biggest ones. This ensures that all intersection points are on the actual MMAP. The light source is created and positioned similarly but with the MMAP as the related origin instead of the camera. Also, the light source will be scaled to the same size as the mid-air image.

To calculate the stray light of the MMAP we are using the method of Hayashi, R. [6]. This method can calculate the position and degree of distortion of the stray lights depending on the viewing angle. However, instead of using an affine transformation, we are using a homographic (perspective) transformation as it provides better results.

4. Results

To proof that our proposed method is working as intended, we are running three test scenarios for which we want to use our method. First, we want to check what MMAP our method generates for a very big mid-air image. In reality, we would need a MMAP that does not fit into a regular room and is not practical in an office for example. Second, we want to check what MMAP our method generates for a mid-air image that can only be seen by very small people like children. We would expect an optical system where the MMAP is (almost) parallel to a wall with a low height. That would result in a viewing area only viewable by small people. Third, we want to check that the method

changes the size of the MMAP according to the view area. In reality, the view area of a MMAP is narrow, so we need a larger MMAP for a larger viewing area.

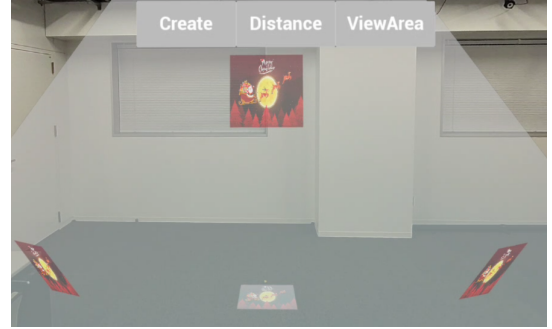


Fig. 3: The result of the first scenario. A big mid-air image results in a big MMAP

Figure 3 shows the result for the first scenario where we create a large mid-air image by spanning our arms as far as possible. Thus, the distance between camera and mid-air image is low which led to a large angle for the viewing vectors. That resulted in a very large MMAP which does not fit practically in our room, as we expected. For the second scenario we created a mid-air image when sitting on the floor and placing it at approximately 45 degrees above our head. Figure 4 shows the mid-air image at the described position and that the MMAP has been created almost parallel to the wall / window, which is as expected. The figure also proofs that it can be seen by small people. To check whether the mid-air image can be seen by a tall person like an adult, we stood up at the same position from where we created the mid-air image. From there, we looked at the position where the mid-air image would be.



Fig. 4: Perspective when crouching and looking at the MMAP.

Figure 5 shows the view from that position and proofs that the mid-air image cannot be seen by tall people. Only the MMAP and the light source are visible which is what we expected.



Fig. 5: Perspective when standing and looking at the MMAP.

For the third scenario we created a small mid-air image in front of us. Figure 6 shows our perspective when we moved our head a small distance to right/left. It shows that the mid-air image starts to disappear and the stray light is not visible. That proves that the viewing angle of the mid-air image is narrow. So, we extended the viewing area by approximately 15 cm in each direction using the procedure described in 3.1.

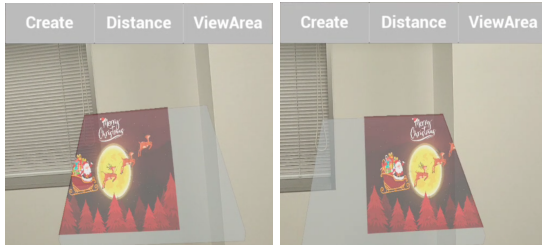


Fig. 6: Perspectives after creating a small mid-air image. The view area is very small

Figure 7 shows our perspective after changing the view area from the bottom border. It proves that the larger view area requires a larger MMAP to display the mid-air image. Also, the borders work correctly as the mid-air image should be seen at the top edge of the MMAP when we look from the bottom border of the view area.

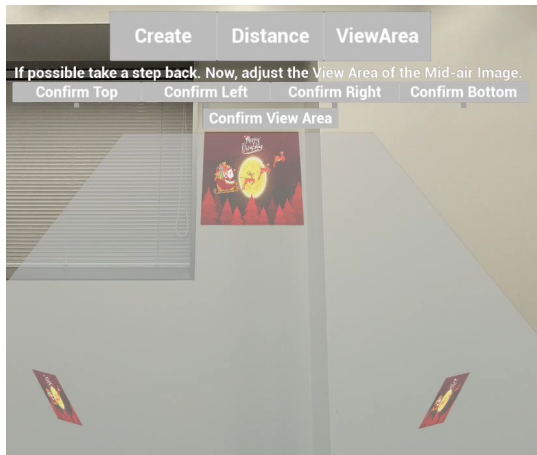


Fig. 7: Perspective after changing the view area. Mid-air image can be seen in a wider area. Stray light also visible

5. Discussion

As we can see in the result section our proposed method allows the easy creation of mid-air images and the necessary optical system using simple hand gestures and movements. As it is a mixed reality application, the user can also directly see the optical system in a real space without the need to find a workaround for transporting it from the virtual world to the real one. This gives the user the opportunity to check whether the optical system they desire, can fit into the environment they wish to place it in.

However, our proposed method still faces some limitations. First, the optical system is not influenced by lighting

environment of the real world. In reality, other light sources like lamps or the sun influence the visibility of the mid-air image and the stray light. The more light there is the harder they are to see. Second, the light source does not influence the mid-air image and stray light. In reality, the light source shoots light through the optical system to generate the mid-air image and stray light. So, there is a direct connection between both of them. However, in our proposed method the light source does not influence neither the mid-air image nor the stray light as we create them separately. To reproduce the mid-air image's generation by the light source, we would need ray tracing for MR which is not computational possible in real-time yet.

6. Conclusion

In the end, we were able to propose a new method that allows the user to reproduce an optical system in mixed reality by forming certain hand gestures. That allows the user to verify whether their imagined version of the installed optical system is feasible in their respective environment. By running three scenarios we showed that our proposed method is working as intended and that it can be applied to different situation, proving that it is very useful for designing mid-air imaging systems in the real world.

Acknowledgement

This work was supported by JST FOREST Program, Grant Number JPMJFR216L.

References

- [1] M. Otsubo, "Optical imaging apparatus and optical imaging method using the same," Apr. 22 2014.
- [2] S. Maekawa, K. Nitta, and O. Matoba, "Transmissive optical imaging device with micromirror array," in *Three-Dimensional TV, Video, and Display V* (B. Javidi, F. Okano, and J.-Y. Son, eds.), vol. 6392, pp. 130 – 137, International Society for Optics and Photonics, SPIE, 2006.
- [3] H. Yamamoto, Y. Tomiyama, and S. Suyama, "Floating aerial led signage based on aerial imaging by retro-reflection (airr)," *Opt. Express*, vol. 22, pp. 26919–26924, Nov 2014.
- [4] T. Koike and Y. Onishi, "Aerial 3d imaging by retroreflective mirror array," in *Proceedings of the 2018 ACM Companion International Conference on Interactive Surfaces and Spaces, ISS '18 Companion*, (New York, NY, USA), p. 25–29, Association for Computing Machinery, 2018.
- [5] S. Kiuchi and N. Koizumi, "Simulating the appearance of mid-air imaging with micro-mirror array plates," *Computers & Graphics*, vol. 96, pp. 14–23, 2021.
- [6] R. Hayashi, A. Saito, and N. Koizumi, "Reproducing the position and deformation of stray light in a retrotransmissive optical element using a game engine (jp)," 2022. Title is manually translated from Japanese. Original is in Japanese.
- [7] U. Gruenefeld, J. Auda, F. Mathis, S. Schneegass, M. Khamis, J. Gugenheimer, and S. Mayer, "Vrception: Rapid prototyping of cross-reality systems in virtual reality," in *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems, CHI '22*, (New York, NY, USA), Association for Computing Machinery, 2022.