Floor Plan Generation from Tracked Physical Objects

Thomas Poyet, Hugo Queinnec, Lucas Teissier, Gonca Yilmaz
ETH Zürich, Switzerland
{tpoyet, hqueinnec, lteissier, gyilmaz}@student.ethz.ch

Abstract

The ability to track 3D physical objects and integrate them with mixed reality has numerous potential applications, particularly in the field of architecture. In this paper, we introduce FloorPlanGen, a new application for generating floor plans from tracked physical objects. We tackle tracking, user experience, integration of a server, and generation of the plan. Through evaluation and discussion, we show that our application does reach its goal but further research is needed to optimize the experience. We deploy our application to HoloLens and invite non-expert users to test our demo. We show that adopting floor plan generation with physical objects can improve the immersive experience for the users, and include non-experts in decision-making for floor plan design.
Our code is publicly available on GitHub.

1. Introduction

Mixed reality applied to the field of architecture has the potential to change the way we interact with our environment. In this paper, we tackle the problem of generating a home layout on top of 3D-printed furniture. This allows the user to think differently and first choose the position and orientation of the furniture before the ones on the walls. As such, a user could decide to place the bed in front of the rising sun without being prevented by the layout of the house for instance. Floor plan generation from tracked physical objects includes three essential components: the live tracking of the 3D objects, the communication with the server, and the generation of the layout. We also focused on usability to make the application user-friendly. Hence, we discuss the user interface, evaluate our application and show some limitations.

2. Related Work

2.1. 6-DoF Estimation

FoorPlanGen relies on tracking 3D-printed furniture models, in order to virtually display the user interface over them. Tracking 3D objects in space is generally referred to as *6-degree-of-freedom estimation* in the literature, as tracking involves estimating the 3D position and orientation of the object. To have an idea of the state of the art in the field, the BOP challenge [1] is a great resource. In this challenge, different datasets are used to evaluate the models. Since our application involves tracking symmetric and textureless objects, we focus on the scores obtained on the T-Less dataset. Among the 26 models evaluated in the BOP challenge [1], the one that performed best on this dataset is CosyPose-1view [4]. This latter is able to retrieve the positions and orientations of several objects in a scene. It can be used either in a single-view scenario or in a multiple-view scenario with unknown camera positions. However, in order to apply such a model in an application, it would need to be finetuned on a dataset consisting of the objects one wants to track. To avoid this work, we decided to use more user-friendly tracking models such as those proposed by Vuforia SDK [10].

2.2. Mixed Reality for Design and Construction

In recent years, there have been several studies on the use of mixed reality (MR) for design. These works have focused on various aspects of MR, including interactions, frameworks, sketching and drawing, configurators, gestures, mixed 2D and 2.5D environments, and output to 3D printing.

Arora et al. (2018) introduced “SymbiosisSketch”, a hybrid sketching system for 3D design in AR that combines drawing in air (3D) and on a drawing surface (2D) [9]. Kwan and Fu (2019) presented a system for sketching in 3D using AR on a mobile device, but their system is geared towards users with good drawing skills [2]. Millette and McGuffin (2016) developed a ”Draw-and-Drop” system for editing 3D AR scenes by sketching and modeling objects on a mobile phone. Reipschläger and Dachselt (2019) used a touch screen with a pen and a HoloLens to model basic shapes in AR [5].

Roo and Hachet (2017) presented a framework for interacting with physical objects and virtual content in different modalities, including object scale modes and immersive en-

vironment 1:1 mode [7]. However, their immersive mode is purely representational and does not support editing. In contrast, our work aims to enable users to edit designs in both object and 1:1 modes.

Son et al. (2020) developed a system that uses projection mapping and physical objects to find and display floor plan layouts matching user-defined criteria such as the number of rooms and boundaries. The system also allows users to try different wall textures on their designs [8]. However, it only works at the object scale and does not offer a 1:1 walkthrough mode. It also has limitations in terms of the types of elements it can handle (e.g., walls and room types but not doors and windows).

There are also some existing methods for generating a floor plan from a given structure to automate the construction and include non-experts in the design process. Kwiecinski et. al presented a method for computer-aided participation in the design of single-family houses, using an interactive user interface on a tablet and a generative design system based on context-free grammar. The method, called HOPLA-Home Planner, allows for flexible configuration of the house layout and was tested through usability studies with non-expert users. The findings suggest that increasing the scope of users' participation in the design process can increase user satisfaction [3]. Despite of the promising results, an immersive interactive environment can increase user engagement leading to more quality outcomes.

Overall, these previous works have demonstrated the potential of MR for design and automation of design construction, but there are still challenges and open problems to be addressed, such as developing more intuitive and precise interactions, supporting a wider range of design tasks and scales, and enabling seamless integration with 3D printing. Our work employs the use of physical objects and virtual content to retrieve the user input and support 1:1 walkthrough mode.

3. Methodology

When loading FloorPlanGen, the user is asked to place an empty home layout on a surface and start the design phase. During this phase, the user is expected to gaze at one piece of furniture to activate the object tracking.

Once the tracking starts, the user can place the furniture piece within the layout. Having placed the object within the layout, the user inputs the dimensions of the rooms and their connections. The home layout is saved as a graph: each room represents one node in the graph and room connections represent the edges. After the placement of all objects with their corresponding inputs, the user can click on the floor plan generation button. This action sends the graph to the server and retrieves the 3D mesh of the generated plan, which is then displayed to the user.

In addition to the floor plan generation, our application also offers a 1:1 walkthrough mode, where users can explore the interior of their generated house in a 3D world by moving the 3D-printed camera object in the physical world. This provides an even more immersive and realistic way to experience their design, allowing them to move through the space and get a sense of the layout and flow of their home.

To make this possible, FloorPlanGen is composed of three main elements: *object tracking*, *user interface*, and *plan generation*. The first two elements are managed locally on the HoloLens and the last one is done on a server. The two devices communicate via the internet using Photon Engine available on Unity. In the following sections, we will detail these major building blocks and explain their implementations.

3.1. Furniture Tracking

Vuforia SDK [10] is chosen, rather than Azure Object Anchors because it helps deal with another major challenge for tracking 3D objects: the size of the objects (10cm long in our case). 2D and 3D object tracking were both explored and will be discussed with their own advantages and limitations in the following paragraphs.

3.1.1 2D Tracking

Vuforia Image Targets [11] allows tracking multiple images in real-time. However, for performance, it is recommended not to track more than 5 objects simultaneously. A 2D printed image is stuck to each 3D model and then tracked to augment virtual objects on top of the furniture. This requires meeting specific conditions and has some limitations. First, furniture need to be flat to be able to stick a flat 2D image on them. Thus, it works better for tables and beds than for complex 3D geometric shapes such as toilets. Also, we need to look at the furniture from above to look at the images, in our case, it is not necessarily a real limitation because it is an intuitive angle. To track images, Vuforia detects keypoints on them. Tracking works better when images are well-lit, with good contrast and well-distributed features such as well-scattered unique patterns. Comparison for two different hand-made textures is shown below in Figure 1. In the right image, flowers are not well detected because the contrast is low (light blue on white) and because the shape does not contain sharp corners but mostly circles.

3.1.2 3D Tracking

The 3D tracking is performed using Vuforia Advanced Model Targets [12]. We opted for the advanced model and not for the regular one because the former did not require a first step of alignment between a virtual outline of the object and the physical object to start the tracking. In addition, we noticed that Advanced Model Target was better at tracking the object, it lost tracking less frequently when the object or

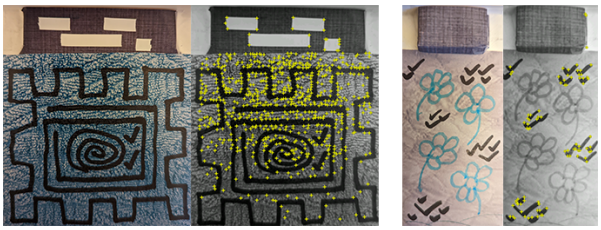

Figure 1. Good texture (left) vs. bad texture (right).

camera was moving. The use of Vuforia Advanced Model Targets requires a training step as it is based on a deep learning approach. The training is done on the Vuforia servers. For all our models, 6 in total, it took about 2 hours.

However, even with Advanced Model Targets, our initial tracking tests were disappointing. The tracking was difficult to activate and could be lost easily. By reading the Vuforia Model Targets' best practices, we realized that our 3D models were too simple. Indeed, according to Vuforia: "Geometric complexity is key to distinguish an object from other shapes in the environment.". After identifying the problem, it was easily solved by making more complex 3D models of the tracked objects. For example, in Figure 2 a comparison between the original 3D model and the complex model of the dining table can be seen.

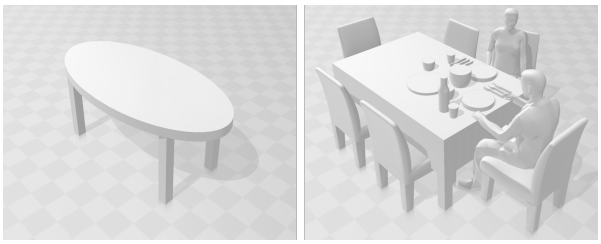

Figure 2. Comparison of the dining table model. On the left is the original one, and on the right is the complexified one.

During these initial tracking tests, we also noticed that Vuforia Model Targets was only able to track one object at a time and that the tracking worked less well on the HoloLens than with a webcam. The first problem is a fundamental limitation of Vuforia that we were not able to overcome. The second is due to the fact that the HoloLens camera is located above the user's eyes, so it does not see exactly what the user sees. Placing the object in the center of what the camera sees helps Vuforia initialize the tracking, which means the user has to look a little lower than the object to get it in the middle of the camera frame. To circumvent these problems, we developed a custom interface between Vuforia and our application.

Regarding the first issue, this interface allows us to check which object is being tracked; this information is used by the user interface to give feedback to the user. It also allows us to disable tracking of all objects except the one the user specified, which is useful when the user only wants to track a specific object near another.

This interface also allows us to update the position and orientation of the user interface augmenting each piece of 3D-printed furniture so that it is located at the last tracked position of the objects and always oriented towards the user. Such localization of the user interface forces the user to look lower at the object, thus circumventing the second problem.

3.2. User Interface

The user interface (UI) of our application can be divided into three main parts: starting the scene and placing the outline on a flat surface, moving the physical objects and interacting with the virtual UI to adjust room sizes and connections, and generating the floor plan and managing visualization settings. This section will focus on the UI attached to the physical objects during the phase of object manipulation to place the rooms.

Virtual UI elements are attached to each physical object to facilitate intuitive and easy interaction. These elements include a slider to adjust the size of the future room, text to display the name of the object, and a "diamond sphere" that serves as both a tracking indicator and a tool for creating connections between objects. The diamond sphere can be grasped and released near other objects to create links, which will represent connections such as doors between rooms in the generated floor plan. This is shown in Figure 3.

There were several challenges to be addressed in the design of the virtual UI.

First, the UI had to be attached to the physical objects and always within the user's reach. To achieve this, we placed the UI elements according to the position information received from the object tracking system and made them always face the user, rotating around the center of the physical object. This way, the user can always interact with the UI elements without the need to move somewhere else.

Second, the UI had to handle the fact that the object recognition system often loses track of the objects and can only track one object at a time. To address this, we placed the UI elements a little "below" the physical object, which orientates the HoloLens camera directly towards the 3D model and significantly improves tracking accuracy and speed. The diamond sphere also serves as a tracking indicator, rotating in green when the object is actively tracked and remaining stationary in blue when it is not. If tracking is lost, the UI stays at the last tracked position, allowing the user to seamlessly move on to manipulating the next object by simply looking at it and repeating the process. This ensures that the first object remains in place as expected while the user is not looking at it.

Finally, the UI had to notify the user if any objects were

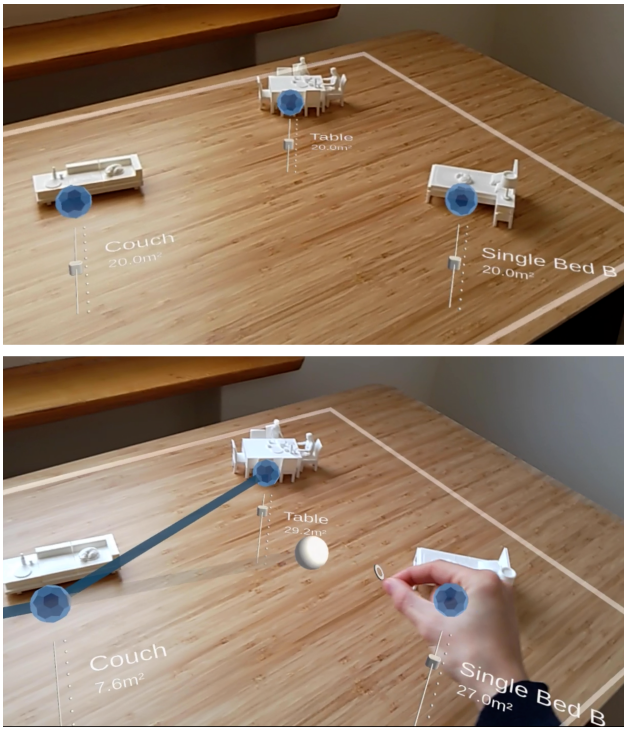

Figure 3. The top image shows what the user sees through the HoloLens when he has placed several pieces of furniture. In this picture, no object is actively tracked. The bottom image shows the user creating a link between two objects. On both images, "diamond spheres" can be seen in blue.

placed outside the outline of the floor plan, as these will not be considered in the final generation. To test if an object, represented by its 2D position in the 2D plan, is inside the floor outline, represented by a polygon, we have to solve a point-in-polygon problem, and we used the crossing number algorithm. The diamond sphere turns red if the object is outside the outline, providing a visual warning to the user (shown in Figure 4).

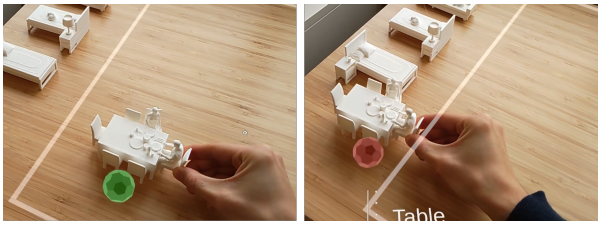

Figure 4. Visual indicator of an object being inside or outside the plan outline. Left: inside (and actively tracked) - Right: outside.

3.3. Communication

We knew from the beginning that we would need a backend server along with our application. This decision was motivated by two main factors. Firstly, the floor plan generation algorithm, which is expected to eventually be a large machine learning model, cannot run on the HoloLens itself. Instead, it must be housed on a distant server and communicate with the HoloLens through a predetermined interface. Secondly, we included a physical camera model along with the furniture, which will need to be tracked and its position transmitted to the distant server in order to view the generated house model from that perspective.

To facilitate this communication, we have implemented a system diagram (shown in Figure 5) that utilizes the third-party module Photon Unity Networking [6]. Originally developed for multiplayer game functionality, this module enables the physical camera model seen through the HoloLens and the virtual camera on the distant server to move in sync. To get a first-person view of the interior of the house, the user can place the camera model inside the virtual house plan. Its position and orientation are tracked by Vuforia Model Targets, and this information is used to move the virtual puppet camera within the coordinate frame of the house plan. Using Photon Engine, the position and orientation of the puppet camera is mirrored to the Unity instance running on the server. Moreover, using a code template provided by our supervisors, Photon Engine allows for the direct transmission of data such as the model graph or generated mesh. In order to use Photon Unity Networking as explained, we have set up a Windows backend server running a backend Unity scene (shown in Figure 6).

In addition to facilitating communication between the HoloLens and the distant server, the backend Unity scene also serves as a mediator between the floor plan generation module and the rest of the app. This is accomplished through the use of files with predefined names, allowing for easy replacement of the generation method if desired. Currently, the floor plan generation algorithm utilizes Rhino, but it is expected to be replaced by a Python machine-learning model in the future (which will be the work of our architecture supervisors).

3.4. Floor Plan Generation

Once all objects have been tracked and positioned and all inputs have been set, the floor plan generation phase begins. To initiate this process, the user must click a button on the hand menu (shown in Figure 7), which appears next to the hand when it is raised and flat. This menu allows the user to have options always close at hand without cluttering their field of view with a permanent menu. If desired, the hand menu can also be grasped and placed in a fixed location in space.

Upon clicking the button, a request is sent to the backend server. Currently, the backend server generates a simple floor plan (a Voronoi diagram) as a proof of concept using Rhino. This process of sending the request, generating the

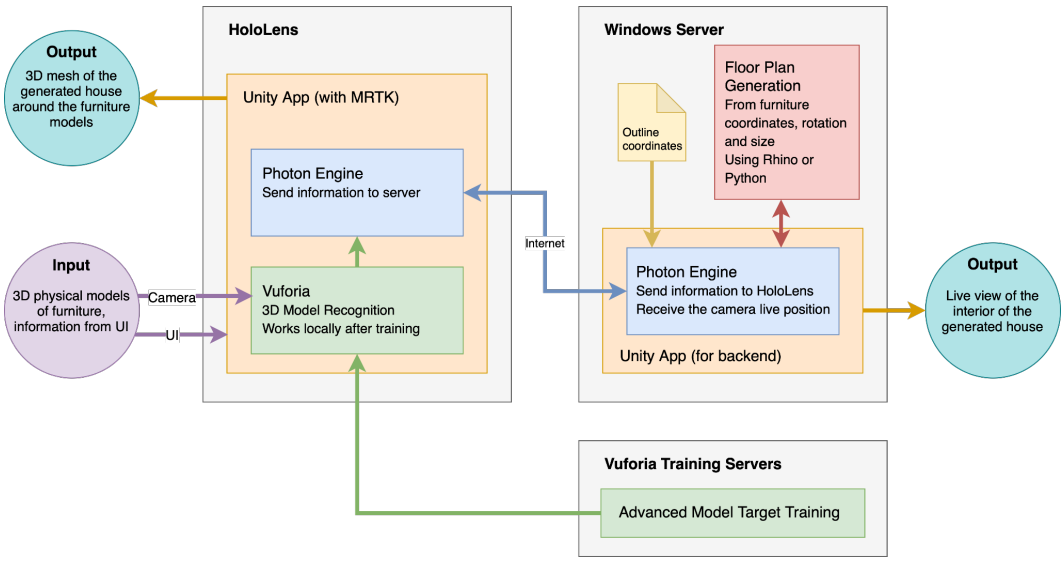

Figure 5. Simplified system diagram showing communication between the HoloLens, its environment, and the backend server, as well as communication within the components.

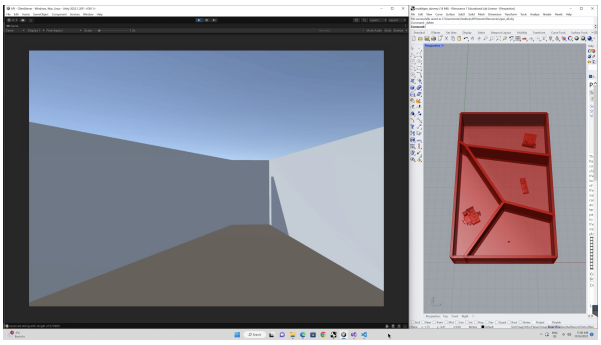

Figure 6. Screenshot of the Windows backend server. On the left, the Unity view received from the movement of the physical camera in the plan. On the right, the Rhino view of the floor plan generation.

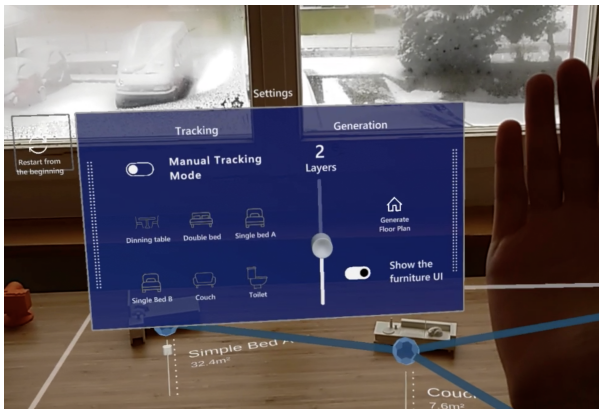

Figure 7. Hand menu, containing settings for manual tracking on the left, and for floor plan generation and display on the right.

floor plan, and receiving the response (the mesh of the generated floor plan) takes approximately 10 seconds.

Once the mesh is received on the HoloLens, it is displayed around the objects on top of the virtual outline. The mesh is divided into four layers, which can be selectively added using a slider in the hand menu. This allows the user to choose how much of the plan to display, from just the base layer to the complete walls. Showing only the first base layer makes it easier to manipulate the furniture while showing everything allows the user to observe the plan details such as doors and windows (once they are implemented by our supervisors in the mesh returned by the generation algorithm). The UI elements attached to the physical objects can also be deactivated from the hand menu to remove clutter and better appreciate the house model. This is shown in Figure 8.

Finally, depending on the user's satisfaction, they can continue to move the furniture and adjust their inputs, regenerating the floor plan until they find the perfect one for their needs.

4. Evaluation

During our demo session, we invited users to test our application and we observed their interactions. The participants were all non-experts in architectural design and construction. The main evaluation component was task completion. The participants were required to

- Overlay the empty home outline on a surface

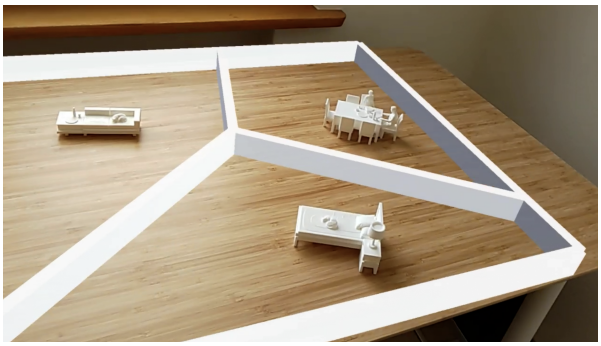

Figure 8. View of the floor plan generated from the HoloLens, here with one layer on four, and with the UI of the objects hidden.

- Start the object tracking and interact with 3D-printed furniture
- Change the room size using sliders and add room connections
- Generate the floor plan
- Try 1:1 walk in the floor plan by using a small 3D-printed camera object

Participants were able to accomplish most of the tasks. However, some parts were perceived as more difficult than others. In terms of satisfaction, most users had a problem with the sliders and stated that it was hard to drag and drop them. The sound effects during the interaction, however, were perceived to be very helpful to improve the immersive experience. Participants also found it helpful to see a loading animation during the generation of the floor plan as it eased the waiting time. The use of real furniture during the design phase instead of virtual was found very intuitive.

In regards to efficiency, the most time spent was on the second task. Due to the limited capabilities of 3D object tracking, the users could only interact with one piece of furniture at a time and they needed to gaze at them for a while before Vuforia could start the tracking. The tracking of some furniture pieces was easier than others, resulting in different time intervals spent on tracking different pieces of furniture.

For effectiveness, we observed whether tasks were completed. All tasks, except for the second and last, were fully completed by users. In the second task, due to the reliance on object tracking, some objects were not captured or recognized by the Vuforia engine resulting in partial completion of this task. For instance, all users were able to activate the tracking for the dinner table object, whereas only a few were able to activate it for the couch object. Another outcome of our observations is that some were not able to try the last functionality because of time limitations we had for each participant or because the algorithm was not able to track the camera object reliably as it was smaller than the ideal size and the users could easily occlude the object while moving it with their hand.

Our user study showed that the incorporation of mixed reality in interior design can greatly benefit non-experts by allowing them to more easily participate in the design process and create their dream home. The use of physical objects and their placement proved to be highly intuitive for users. However, there are still areas for improvement in terms of tracking certain furniture pieces, the camera object, and the collection of room sizes using sliders. Overall, there is potential for further development and optimization of the user experience in our application.

5. Discussion

5.1. Furniture Tracking

The 3D tracking of objects is not perfect. Only one object can be tracked at a time, but by adapting the user interface we were able to work around this limitation.

It can be quite difficult to activate tracking, but once the object is recognized by Vuforia, the tracking is quite stable as long as the user holds it without much occlusion. To help Vuforia recognize the object when the user is looking at it, we adapted the user interface so that the user looks lower than the object, which places the object in the center of the HoloLens field of view.

The quality of the tracking also depends on the lighting conditions, it works much better in a bright environment. It doesn't seem to be affected much by the background color. During our test, the tracking worked just as well on a light background as on a dark one.

5.2. User Interface

Overall, the user interface was functional and intuitive for the end user, provided they received some initial explanations on how to use it, particularly in regards to interpreting the tracking status of the objects. We observed that the rotating UI on one object can sometimes lead to overlap with the UI of the other objects, making it difficult to use. However, this issue rarely occurred in practice and the user was intuitively able to position themselves in a way to avoid overlap.

User testing also revealed that it was sometimes hard to grab the slider due to its orientation and the fact that it was close to the table, causing the HoloLens to have difficulty recognizing the pinching gesture. This placement was necessary for optimal tracking of the 3D models, as explained in Section 3.1.2.

5.3. Communication

In terms of communication, we found that using Photon Engine, which was not originally intended for direct data

transmission, made it quite slow to send and receive data due to its limited capability of sending data as a string with a character limit too low to transmit a mesh in a single round. However, this limitation did not pose a significant issue as the primary bottleneck in this process is expected to be the time required for floor plan generation, which will increase as the complexity of the generation model increases.

It is also worth noting that this application requires an internet connection and does not currently support a local setup, although this is largely a positive feature as it eliminates the need to bring a separate computer in order to run it alongside the HoloLens.

6. Conclusion

In this paper, we presented FloorPlanGen, a mixed-reality application for the HoloLens that aims to make floor plan generation more user-friendly and accessible. Through a user study, we demonstrated the effectiveness of incorporating mixed reality in interior design and how it can allow non-experts to more easily participate in the design process. However, there is still potential for improvement in terms of optimization and user interaction. Further research could focus on optimizing the application, including the use of better hardware to enable real-time floor plan generation and better sensors to improve tracking quality, particularly in low-light environments. In addition, we plan to explore the use of deep learning techniques to generate more realistic and consistent house layouts, rather than relying on Voronoi cells. Finally, a key goal for future work will be to track multiple 3D objects simultaneously, similar to how we currently track 2D images.

Acknowledgments

We would like to extend a special thank you to Anton Savov and Wenqian Yang for their supervision, guidance and support throughout this project. Their contributions, including the provision of 3D models, 3D printing, and assistance with setting up the API and server-side logic, were essential in the successful completion of this project. We are grateful for their contributions and support.

References

[1] Tomáš Hodaň, Martin Sundermeyer, Bertram Drost, Yann Labbé, Eric Brachmann, Frank Michel, Carsten Rother, and Jiří Matas. Bop challenge 2020 on 6d object localization. In *European Conference on Computer Vision*, pages 577–594. Springer, 2020. 1

[2] Kin Chung Kwan and Fu Hongbo. Mobi3dsketch: 3d sketching in mobile ar, 2019. 1

[3] Krystian Kwieciński. Interactive generative system supporting participatory house design, Nov 2022. 2

[4] Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic. Cosypose: Consistent multi-view multi-object 6d pose estimation. In *European Conference on Computer Vision*, pages 574–591. Springer, 2020. 1

[5] Alexandre Millette and Micheal J McGuffin. Dualcad: Integrating augmented reality with a desktop gui and smartphone interaction, 2016. 1

[6] Photon. Photon Unity Networking. https://www.photonengine.com/pun (Jan. 2023). 4

[7] Joan Sol Roo and Martin Hachet. One reality: Proceedings of the 30th annual acm symposium on user interface software and technology, Oct 2017. 2

[8] Kihoon Son and Hwiwon Chun. C-space: An interactive prototyping platform for collaborative spatial design exploration: Proceedings of the 2020 chi conference on human factors in computing systems, Apr 2020. 2

[9] Rahul Arora University of Toronto, Rahul Arora, Rubaiat Habib Kazi, Tovi Grossman, George Fitzmaurice, and Karan Singh. Symbiosissketch: Proceedings of the 2018 chi conference on human factors in computing systems, Apr 2018. 1

[10] Vuforia. Vuforia API Reference. https://library.vuforia.com/sites/default/files/references/unity/index.html (Jan. 2023). 1, 2

[11] Vuforia. Vuforia Image Targets Introduction. https://library.vuforia.com/objects/image-targets (Jan. 2023). 2

[12] Vuforia. Vuforia Model Targets Introduction. https://library.vuforia.com/objects/model-targets (Jan. 2023). 2