

CARNEGIE MELLON UNIVERSITY

School of Architecture

College of Fine Arts

Thesis

Submitted in Partial Fulfillment of the requirements for the degree of

Master of Science in Computational Design

TITLE:

VisionARy: An Open Source 3D Environment Segmentation
Toolkit for Augmented Reality

AUTHOR:

Yuechen Wang

ACCEPTED BY ADVISORY COMMITTEE:

Eddy Man Kim

August 10, 2023

Professor NAME

Principal Advisor

DATE

Abstract

This research is dedicated to making existing 3D semantic segmentation technologies for indoor data accessible to Augmented Reality (AR) developers and researchers, with a focus on enhancing Developer Experience (DevEx). We present VisionARy, an open-source toolkit that bridges the gap between AR applications and 3D segmentation by facilitating the adaptation of raw spatial mesh data and connecting the AR device to a computer.

In this thesis, we define a positive AR developer experience through two key elements: expectation and inspiration. This means developers should clearly understand their technologies' potential and limits, while also staying inspired throughout the AR development process. To embody this vision, we design and implement VisionARy toolkit. This thesis contributes to developer experience research by critically reflecting on the development process using the VisionARy toolkit. Reflecting on the project's outcomes and limitations, our work will inspire continued exploration and improvement in DevEx.

Acknowledgements

I would like to express my sincere gratitude to my advisor, Professor Eddy Man Kim. Thank you for all your support, guidance, and patience throughout this research. The time and effort I've spent with you are treasures I will cherish for a lifetime.

Thanks to the Frank-Ratchye STUDIO for Creative Inquiry and Doctor Harrison Apple for generously providing the essential devices for this research.

Table of Contents

1. Introduction.....	1
1.1. Augmented Reality and 3D Segmentation.....	1
1.2. What Is at Stake.....	3
1.2.1. The Need of Developers.....	3
1.2.2. Innovation Opportunities.....	4
1.2.3. Developers Experience Research.....	5
1.3. Thesis Summary.....	6
2. Background.....	8
2.1. AR Device and Platform.....	8
2.2. Relevant Projects.....	10
2.2.1 Applications with 3D Segmentation in AR.....	10
2.2.2 Open-source Toolkits for AR.....	15
2.2.3 Developer Experience Examples.....	19
2.3 Moving Forward.....	20
3. Hypothesis and Research Questions.....	22
3.1. Hypothesis.....	22
3.2. Research Question.....	23
3.3. Research Goal.....	23
4. Methods.....	25
4.1. Preliminary Experiments.....	25
4.1.1 Problem Formulation.....	29
4.2. Developer Roadmap.....	30
4.3. VisionARy Toolkit Overview.....	32
4.4. Application Case.....	36
5. Discussions.....	43
5.1. Contribution.....	43
5.1.1 Easy Onboarding and Adoption.....	43
5.1.2 Seamless Integration of Technologies.....	45
5.1.3 Encouragement of Innovation.....	46
5.2. Limitation.....	47
5.3. Future work.....	48
6. Conclusion.....	50
Reference.....	52

<u>Appendix.....</u>	<u>55</u>
<u>Appendix A. VisionARy User Manual.....</u>	<u>55</u>
<u>Appendix B. Video.....</u>	<u>63</u>
<u>Appendix C. Source Code.....</u>	<u>63</u>

1. Introduction

1.1. Augmented Reality and 3D Segmentation

Augmented reality is a subset of mixed reality (Milgram & Kishino, 1994). It refers to the cases that virtual objects are “augmented” to a real environment with computer graphics techniques. There are three main characteristics of Augmented Reality: integrates the real and virtual, supports interaction in real-time, and register the virtual objects in 3D (Azuma, 1997). Beyond simply constructing the virtual world and interactions in the real world, augmented reality enhances human perception and understanding of the real environment. (Mekni et al., 2014).

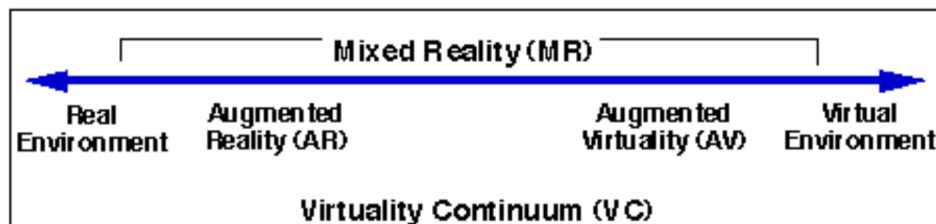


Figure 1. Representation of “virtuality-reality” continuum (Milgram & Kishino, 1994)

Detecting and understanding the real world is important for registering virtual objects in reality to realize augmented reality. As a solution, spatial mapping is effective with 3D digital reconstruction of the environment. Nowadays, there are Augmented Reality devices available for incorporating spatial mapping functionality. For example, Hololens and Magic Leap headset are both equipped with sensors to scan the environment and the devices are capable of processing data, reconstructing 3D geometry mesh that represents the real world settings using internal algorithms. This feature forms the technical foundation of various functions of Augmented

Reality, such as localizing the device or the user, enabling adaptation with room boundaries and the realization of occlusion and collision between virtual objects and environment.

The Hololens is designed to capture the 3D features of an indoor environment using its integrated depth cameras. As users navigate within an indoor space, the device progressively constructs a 3D digital mesh to represent the entire room environment. The device's sensors regularly retrieve depth maps, which the Hololens processor then uses to construct the mesh and integrate it with previous mesh data. While the mesh created by the Hololens isn't characterized by high-density triangles or textures, it remains sufficiently detailed to enable deep learning models to predict the semantic features and understand the space.

With 3D indoor data available from AR devices, deep learning datasets and models have a high potential in performing sentiment segments on this data. This enhances the potential for augmented interaction experiences, extending their capabilities beyond simple localization or interaction with the collective indoor data. In 2017, ScanNet was introduced, offering a RGB-D video dataset designed for deep learning segmentation in 3D scene analysis tasks. The ScanNet dataset includes 20 object class labels for classification. Following this, various models have been developed and benchmarked for segmentation tasks using indoor datasets, such as MinkUNet, CSC, and LGround, thereby further enhancing segmentation in indoor environments. These technological advancements highlight the potential of deep learning and augmented reality in indoor settings.

However, the potential of augmented interaction through segmented real-world data in AR applications has not been fully explored. Although there are some AR research projects employing 3D segmentation, the integration of sentiment segmentation in AR applications

remains a challenging task for most developers. Developers are required to custom build their own segmentation pipelines specific to their projects. This process is particularly daunting for those without a background in deep learning or computer vision. In this context, it becomes clear that there's a need for more accessible 3D segmentation tools that can be applied by a wide range of developers, including those without extensive experience in the area of computer vision or deep learning models.

In view of this, our research aims to offer an accessible solution for 3D sentiment segmentation for AR. We develop a toolkit that simplifies data adaptation and enhances the developer experience. For making 3D segmentation methods accessible to AR researchers and developers, we develop a toolkit that integrates with AR application development workflow, performs segmentation tasks across various settings, and offers intuitive developer interfaces. For developer experience study, we prioritize the aspects of expectation and inspiration as the guiding metrics for VisionARy.

1.2. What Is at Stake

1.2.1. The Need of Developers

At WWDC 23, Apple unveiled Vision Pro, a state-of-the-art AR wearable device designed to provide immersive experiences for daily use, work settings, and entertainment. This innovation offers users the ability to remain present in their environment while augmenting their reality. The introduction of Vision Pro sparked immense enthusiasm within the developer community and presented an exciting glimpse into the future of augmented reality lifestyle. With International Data Corporation forecasting a significant growth in the AR headset market over

the next four years, and a projected 57% growth in the AR software market by 2028, there is a surging interest in Augmented Reality among developers and researchers. This trend is evident in the multitude of research projects and innovative developments emerging in recent years. Accessible tools for certain fields of computer science that applied to Augmented Reality are well accepted and support various kinds of research. Taking the “Hololens2 Research Mode” as an example, this paper is cited by 80 and the project tool received hundreds of forks and stars on the open-source platform GitHub (Ungureanu et al., 2020). Thus, making existing techniques accessible in Augmented Reality development will benefit the large community of AR researcher and developers, and will support the emergence of AR projects. 3D segmentation of AR-reconstructed environments is a field yet to be explored to unlock new interaction opportunities between virtual objects and real-world surroundings.

1.2.2. Innovation Opportunities

With 3D semantic segmentation available in existing Augmented Reality development platforms and workflow, the developers can be more free to explore new opportunities to utilize the data of real world surroundings and reconstruction data, regardless of the domain knowledge in the field of 3D geometric processing or deep learning. As sensor technology and segmentation algorithms continue to advance, such a toolkit bridging the gap between AR application development and 3D computer vision techniques can evolve for certain application cases and continue providing insights.

1.2.3. Developers Experience Research

By improving the accessibility of existing techniques for developers, the research also explores how to provide a satisfying Developer Experience (DX/DevEx). Developer experience is an idea that reflects how developers perceive their working activities, assuming that improving the experience of development will benefit their project outcomes (Fagerholm & Münch, 2012). Microsoft identifies the four perspectives of measuring developer experience as build, test, start and debug.

Good developer experience impacts the success of an open-source project. NumPy, for example, is an open source Python library for scientific and numerical computing. It provides developers with less space and time consuming computation, as well as scientific functions such as linear algebra and vectorized calculation. NumPy is easy to use for developers for its efficiency working with large data, scientific functions it supports, and intuitive and consistent API(Application Programming Interface) that is easy for developers to learn and use. The document of NumPy helps developers to adapt NumPy for their specific needs, and there is a large community that contributes issues, updates and troubleshooting advice for developers.

In this research, we aim to explore how we can cultivate an optimal developer experience through the creation of an AR development toolkit. Our primary focus is on empowering developers to manage their expectations and derive inspiration. We are also dedicated to enhancing the accessibility of AR technology, making it easy for developers and researchers to get started and adapt to a wide range of use cases. Furthermore, we seek to understand how we can foster a thriving community around this project in the future.

1.3. Thesis Summary

Firstly, this thesis conducts a conceptual investigation into defining a positive developer experience with a focus on AR application development. We examine various relevant projects that involve interaction with 3D environmental data, to identify the gap between these projects and the expected utilization of segmentation technologies. We also explore other open-source projects that contribute to AR accessibility and enhance the developer experience, to conclude what constitutes a good developer experience in this context.

Secondly, informed by preliminary experiments and an analysis of development needs and developer experience pain points, we design the framework of an open-source toolkit. This toolkit is suitable for AR application development, and it centers around the ideal developer experience we aim to provide, prioritizing expectation management and inspiration. We evaluate and adapt certain existing 3D segmentation algorithms for integration into the toolkit. This section provides a comprehensive account of the toolkit's design and development emphasizing on its various functional modules.

Lastly, this research evaluates the developer experience VisionARy offers through a case study involving the development of an AR application using the toolkit. It assesses the process developers follow to complete their projects, with particular emphasis on how they initiate and complete the essential aspects. The quality of documentation and tutorials are also used as metrics to evaluate the toolkit's accessibility for developers and researchers. In doing so, we provide a comprehensive analysis of how effectively VisionARy facilitates and enhances the AR

application development process and how it demonstrates expectation and inspiration in developer experience.

2. Background

2.1. AR Device and Platform



Figure 2. Microsoft. *Hololens (1st gen) hardware.* Updated November 22, 2021. Accessed August 1, 2023.

<https://learn.microsoft.com/en-us/hololens/hololens1-hardware>

In this research, we select Microsoft HoloLens as the targeted AR device. HoloLens is an industry-leading AR device from Microsoft that performs well within the Microsoft ecosystem and has a large community of developers. Not only does the HoloLens provide an immersive experience of virtual objects in the real world, but it's also an open platform for researchers and developers to create custom applications. Microsoft maintains detailed documentation, developer guide, and comprehensive instructions for getting started and troubleshooting. Developers can easily create a project using accessible development platforms such as Unity or Unreal on a personal computer and deploy it to the HoloLens headset. With existing toolkits and example projects, prototyping augmented reality applications can be easy for researchers, regardless of

their technical background. Moreover, as the Hololens presents a vision of AR's applicability across industry, entertainment, and everyday use scenarios, our contributions to the Hololens ecosystem not only hold lasting value but also benefits a broader range of scenarios.

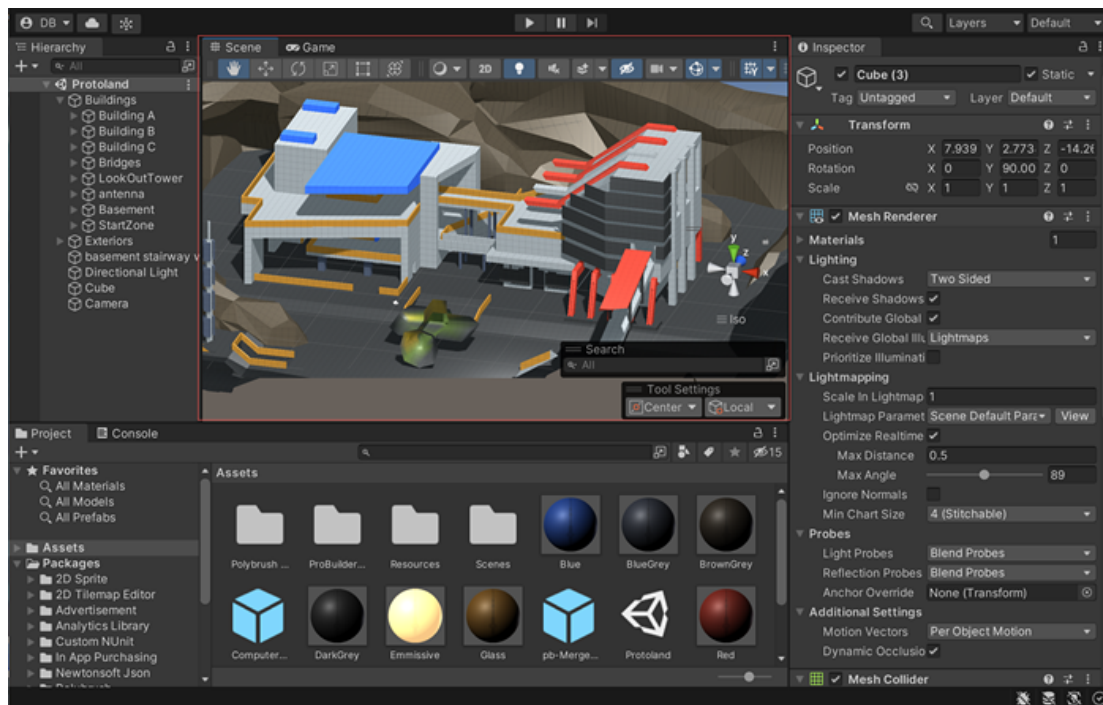


Figure 3. Unity Technologies. Unity interface. The Scene View. Updated July 27, 2023. Accessed August 1, 2023.

<https://docs.unity3d.com/Manual/UsingTheSceneView.html>

Unity3D is our selected development platform for AR application development for this study. Unity3D is a cross-platform development tool offering a friendly interface to developers. With a graphical interface that allows developers to create interactions with its native feature set, Unity is easy to use for all kinds of developers. Its cross-platform capability and well-adapted AR SDK(Software Development Kit) make it an ideal platform for AR application development. Also, the strong community of developers around Unity and the large number of easily accessible documents and tutorials available makes a good foundation of developer experience.

2.2. Relevant Projects

2.2.1 Applications with 3D Segmentation in AR

(1) Simple feature detection in indoor geometry scanned with the Microsoft Hololens
(Björk, 2020)

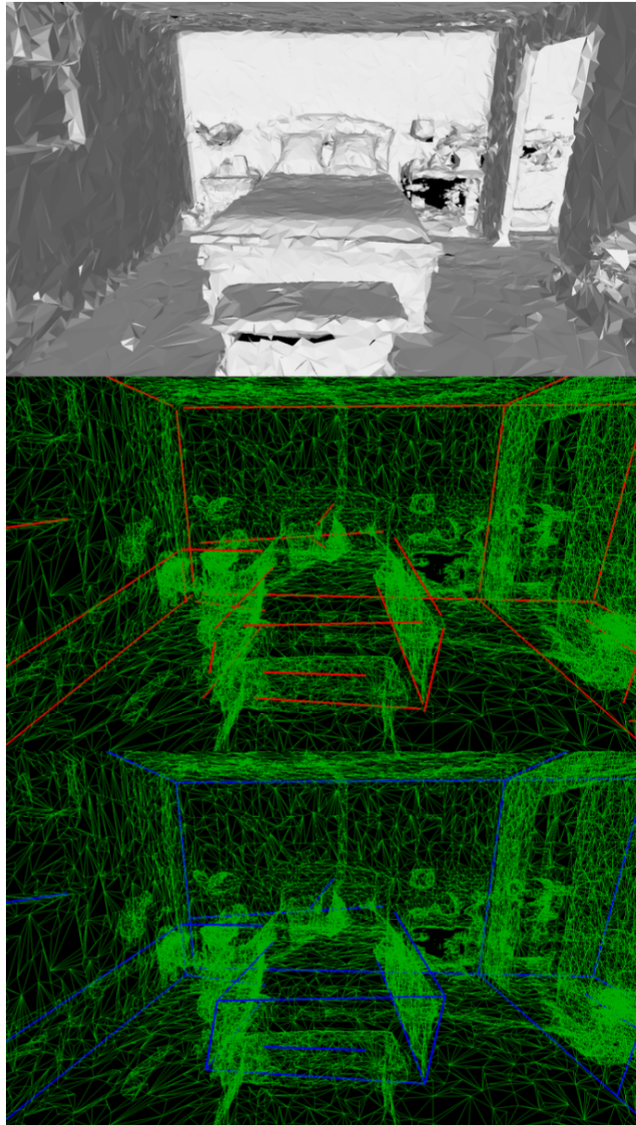


Figure 4. *Examples of features found in the data set (Björk, 2020)*

This study takes in raw spatial map data collected by Hololens, aimed to detect line-type features in the 3D mesh geometry. It applied a combination of LSQ-plane fitting algorithm and relevant mesh cutoff variables as the detection method, but also suggested further improvements are needed. The study mentioned that the choice of method depends on the application's requirements for accuracy and performance.

The work first identified the plan surfaces with LSQ fitting algorithm, and then defined the straight line features by computing on the plane intersections. Thus, it is able to get the geometric boundaries of rooms and layouts in an indoor setting. Although this is not applied to a working Hololens application, the method has well adapted the raw 3D mesh data from the AR headset and analyzed the essential data out of it.

(2) HoloLabel: Augmented Reality User-In-The-Loop Online Annotation Tool for As-Is Building Information (Agrawal et al., 2022)



Figure 5. *A view in the scene and the corresponding ground truth as obtained from the 3D mesh. Left: RDB image of the room. Right: Ground truth mesh annotation with labels. (Agrawal et al., 2022)*

The HoloLabel project is an augmented reality application on HoloLens with the functionality for users to directly annotate a scene in 3D. The application enables users to

annotate rich semantic information while capturing the spatial mesh in real time. With a backend that runs a geometric segmentation algorithm on the remote server, the spatial mesh is transmitted and segmented. In the backend, a point cloud is generated by sampling the mesh, then a Dense-Based Spatial Clustering algorithm is applied to segment the point cloud. Within certain parameters to limit the maximum vertex amount, the algorithm tends to over-segment for the future annotation task. After point cloud segmentation, the result is projected back to the original spatial mesh to segment the mesh. Then, the segmented mesh is sent back to the user for them to annotate in the immersive environment. Users can paint on the mesh segment or individual face in the AR application, and choose a label from a predefined list to assign semantics.

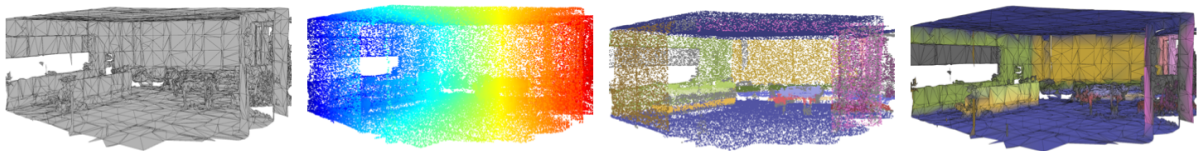


Figure 6. *Segmentation Pipeline: input a mesh, sample a point cloud, segment the point cloud, consolidate segmentation results back on mesh and return to the HoloLens. (Agrawal et al., 2022)*

This tool proposed a human-computer collaborative decision-making protocol to perform the task of 3D semantic segmentation, leveraging the HoloLens's spatial mapping feature and an automatic geometry-based segmentation algorithm to generate the fundamental stage for the user to perform manual annotation. In this project the computer segmentation is proposed for a preparation for users' segment task, while the flow connecting AR devices to a remote backend to perform complex automatic segmentation tasks is inspiring.

(3) Augmented Reality for Users with Low Vision (Schrempp et al., 2021)

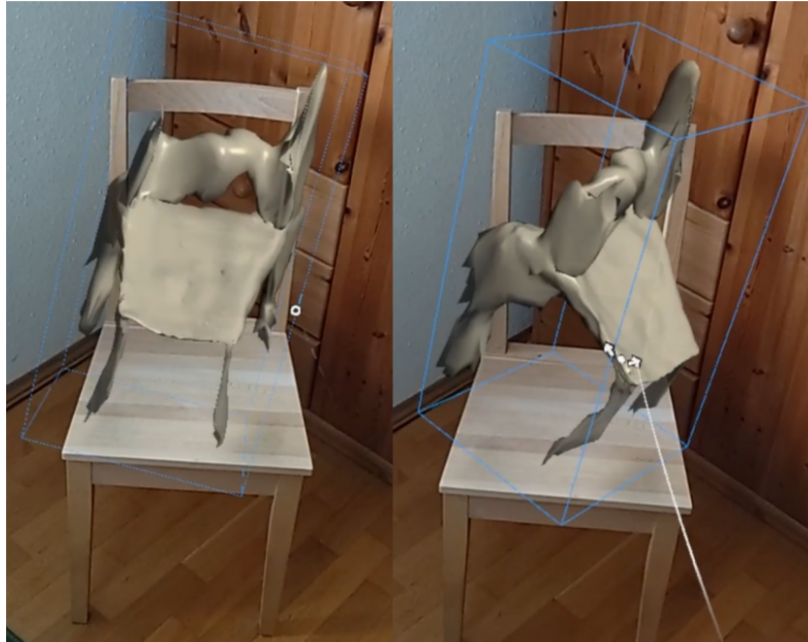


Figure 7. *Interaction with the generated 3D model of a chair (Schrempp et al., 2021)*

This work aims at presenting real-world objects to users with AR headsets as virtual objects (Holograms), as a method for supporting low-vision users' everyday life. With Unity and Mixed Reality Toolkit, the researchers create the project that segments on the 2D RGB image taken by Hololens webcam, and generates corresponding 3D OBJ data with Mask R-CNN model. It also involved a backend server to perform high computation tasks, but in the performance evaluation, it is shown that transforming the result model to the Hololens device is highly time-consuming, which is not friendly to users or developers.

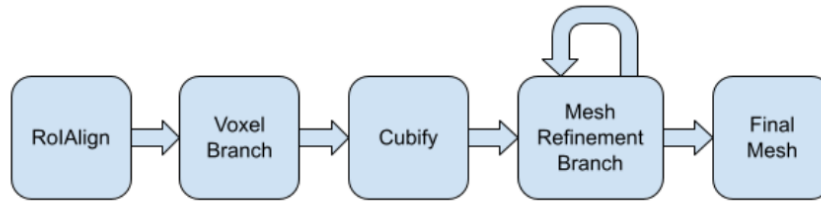


Figure 8. *System overview of Mesh R-CNN (Schrempp et al., 2021)*

In this work, it also documented the selection of semantic segmentation and datasets for their specific requirements. In conclusion, the factors that influenced the technical solution of choice in this project is data adaptability and technical accessibility. As the researchers mentioned, they first decided to utilize the RGBD data to perform segmentation, and then choose an implementation of data processing with this data format. It is also suggested that the well documented and abundantly available information of using and altering the Mask R-CNN architecture to train on RGBD data is the motivation of applying it in this project. The developer report is a helpful resource for developing a well-adapted toolkit for AR researchers and developers.

(4) Semantic Interaction in Augmented Reality Environments for Microsoft HoloLens (Schutt et al., 2019)

This research proposed a vision of interacting directly with the environment in Augmented Reality settings with the semantics classification of HoloLens data. The project utilizes both the spatial mesh and RGB image data collected by the application runtime. In each segment task, the mesh and an image with transformation metrics is sent to the server, where a RefineNet model is applied to classify the pixels of a chair from the image, and the

corresponding vertex of the mesh is filtered to represent the 3D segment. As a result, if the mesh of an object is completely collected in the beginning phase, a well segmented mesh with small misalignment can be captured for further functionalities for the AR application. In this case, it can make the mesh segment an Unity GameObject¹ and assign interaction to it.

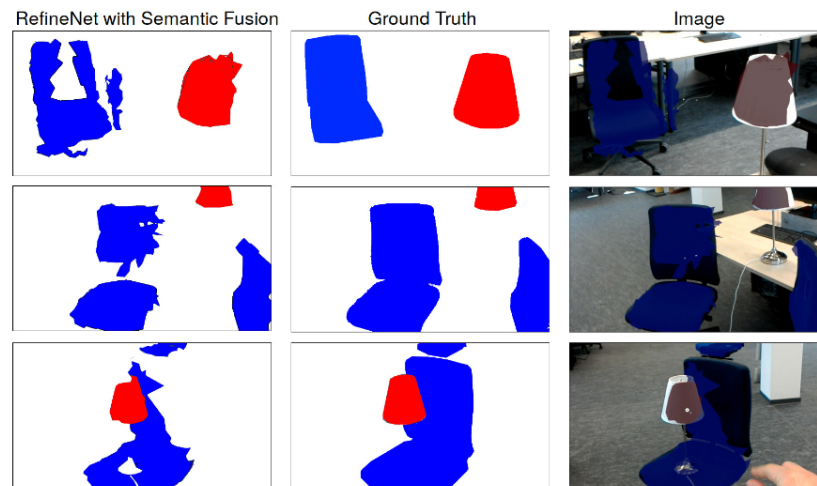


Fig. 4. Examples from the semantic segmentation for each column from the left to right: Segmentation with RefineNet and Semantic Fusion on 3D, Ground Truth, Input image

Figure 9. *Examples of the semantic segmentation (Schutt et al., 2019)*

2.2.2 Open-source Toolkits for AR

In this section, the existing open-source toolkit/project for Hololens and Unity platform will be studied, to discuss how they improve technology accessibility for developers and how they support developer friendly experience.

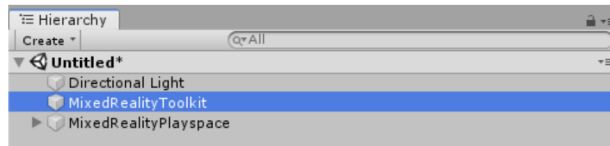
(1) Mixed Reality Toolkit for Unity

¹ GameObject is a fundamental concept in Unity3D. It is the basic unit that developers implement functionalities.

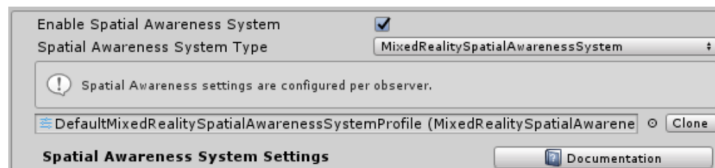
MRTK is an official toolkit for developing mixed reality applications for various types of devices in Unity. It supports developing spatial interactions and UIs with modules of the input system. It consists of extensible UX blocks and example projects for various purposes. Since it is a comprehensive toolkit, here we will take the spatial awareness module as an example, which is also likely to be utilized in the project in this proposed research.

Profile	System Enabled by Default
DefaultHoloLens1ConfigurationProfile (Assets/MRTK/SDK/Profiles/HoloLens1)	False
DefaultHoloLens2ConfigurationProfile (Assets/MRTK/SDK/Profiles/HoloLens2)	False
DefaultMixedRealityToolkitConfigurationProfile (Assets/MRTK/SDK/Profiles)	True

1. Select the MixedRealityToolkit object in the scene hierarchy to open in the Inspector Panel.



2. Navigate to the *Spatial Awareness System* section and check *Enable Spatial Awareness System*



3. Select the desired Spatial Awareness system implementation type. The *MixedRealitySpatialAwarenessSystem* is the default provided.

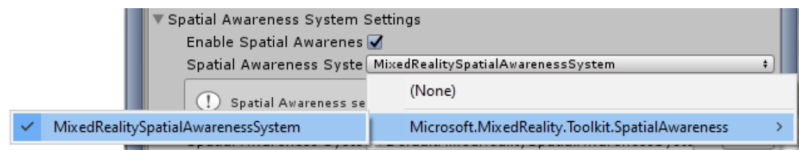


Figure 10. Microsoft. The developer guide of MRTK modules. Spatial awareness getting started. Updated August 1, 2022. Accessed August 1, 2023.

<https://learn.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/mrtk2/features/spatial-awareness/spatial-awareness-getting-started>

For developers to get onboard with the spatial awareness module, a “getting started” document is hosted on the toolkit website. This page can be directly accessed from the homepage of MRTK. A step by step guide with screenshots is provided for developers to start a project with basic spatial awareness from scratch. At the end, the link to the document of configuring and extending the module as well as the API document is shown for further inquiries of the readers. Example scene for the Unity platform is also available via the installer or github repository.

Regarding how this work is advantageous for developer experience, its seamless integration with the Unity platform, the low learning curve it offers, and the well-structured resources and APIs it provides are particularly noteworthy. The integration with Unity's edit and run systems makes it easier to operate, and benefits viable development. It reduces development complexity and facilitates an efficient workflow that developers can appreciate. Meanwhile, the integration with Unity platform ensures a smooth learning curve for developers to effectively grasp its functionalities and utilize them. The ease of use is further supplemented by a comprehensive library of examples and tutorial projects, which provide developers with a direct sense of the project's capabilities and how they can adapt these to their specific needs. Moreover, the project's prefabs² and APIs are meticulously encapsulated, enhancing their usability and making the project more developer-friendly. In summary, MRTK-Unity, by design, offers a comprehensive and developer-friendly experience.

(1) HololensForCV-Unity (Doughty, 2020)

² A prefab is a template object in Unity3D. It is associated with its children, components, and property values.

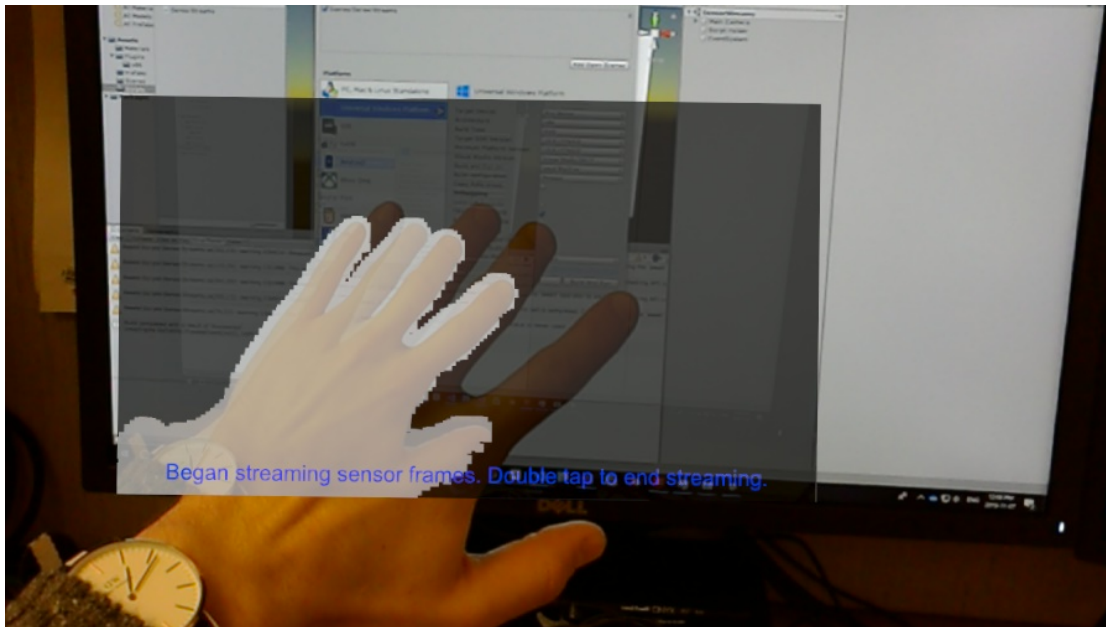


Figure 11. Doughty, M. *HololensForCV-Unity*. Updated August 13, 2020. Accessed August 1, 2023.

<https://github.com/doughtmw/HoloLensForCV-Unity>

HololensForCV-Unity is an open-source project hosted on GitHub. It serves as a great example of making AR technology accessible on the Unity platform. The project offers developers access to sensor data from Hololens, including depth camera and IMU data, which was previously only available through Microsoft's C++ source project. This initiative allows AR developers to access the data directly in the Unity platform, simplifying development. By providing pre-compiled DLL files and clear instructions on application requirements and integration with Unity, the project enables developers to work with raw sensor data streams. Additionally, the project includes two example AR applications to demonstrate the library's usage. Overall, the project successfully provides accessible device data streams for Unity development. However, it doesn't cover the integration of further AR interactions and usage, and it supports only an older version of the Unity platform and Visual Studio.

2.2.3 Developer Experience Examples

Three.js is an open-source JavaScript library and API that enables developers to create 3D graphics in browsers using WebGL. Three.js is hosted on Github and features a dedicated website navigating an example gallery, a comprehensive set of documents and tutorials, and a developer forum. The developer experience with Three.js is good in terms of expectation management and inspiration.

Three.js serves as a compelling testament to the power of managing expectations in the context of developer experience. Upon landing on its homepage, developers are presented with a gallery showcasing a wide array of projects that have been realized using the library. This visual presence not only provides an immediate, tangible sense of what they can achieve using Three.js but also sets realistic expectations from the beginning for developers. The use of real-world applications as a representation of the library's capabilities ensures developers understand the breadth and depth of possibilities open to them. This expectation management is further enhanced by the provision of an interactive user manual and an exhaustive API reference, allowing developers to delve deeper into the functionalities of the library.

In addition, Three.js excels in its ability to provide inspiration for developers. This inspiration primarily stems from the vibrant community that has been cultivated around the library. The developer forum offers a dynamic platform for communication and discussion, fostering an environment of collaborative learning and problem-solving. Developers are given an opportunity to both contribute to and learn from the community, offering them unique insights

and innovative ideas for their own projects. By fostering this sense of community and creating a space for the exchange of ideas, Three.js not only facilitates knowledge sharing but also ignites creativity and encourages exploration.

In drawing lessons from Three.js, good developer experience for accessible open-source projects can prioritize clear expectation management and the provision for sources of inspiration. An open-source project that embodies these principles can foster a thriving community, encourage creative exploration, and drive the development of innovative applications. Therefore, the essence of Three.js's approach may be contemplated when designing for developer experience in the context of similar open-source projects.

2.3 Moving Forward

In our review of precedent projects, we have identified two crucial factors for developing an accessible developer experience – managing expectations and fostering inspiration. These are our prioritized aspects in creating an open-source framework aimed at augmented reality developers.

Effective management of expectations is critical for facilitating the adoption of new technologies. It enables developers to onboard easily, make informed decisions, and work efficiently in the context of their projects. Moreover, fostering inspiration is helpful for the emergence of creative projects, nurturing an active community, and fostering a robust technology ecosystem.

To meet the expectation management goal, we need to provide developers with a clear understanding of what they can achieve with the open-source toolkit. This clarity can be

facilitated through comprehensive documentation accompanied by visual guides, videos or sample projects. Documentation is the starting point for users and it forms their initial impression of the project and significantly impacts users' expectations. Also, we need to make strategic engineering decisions in this project regarding development language and platform choices and the functionalities we choose to incorporate. These choices should aim to enable developers to easily adapt the toolkit to their projects, test and run functions, and reach their expected goals.

To stimulate inspiration, our project should be designed and implemented with a flexible and innovative roadmap. Consider this scenario: a game developer is crafting an AR game that involves interacting with real-world objects in an immersive environment. However, they encounter difficulties in aligning the interactive holograms with corresponding spaces in the real world. Upon discovering our proposed project, they find it not only resolves their challenges but also facilitates a smooth and manageable adoption process due to its structured workflow and comprehensive modules. This experience inspires the game developer to rethink their development process and harness the flexibility of the toolkit in their development workflow. The adaptable and effective toolkit inspires fresh perspectives and drives the evolution of the development process for AR applications.

Moving forward, we will prioritize expectations and inspiration when trying to provide accessible developer experience in creating this open-source project. Guided by the insights gained from our review of precedent projects, we aim to create a toolkit that facilitates and improve the process of AR application development and encourages innovations for developers.

3. Hypothesis and Research Questions

3.1. Hypothesis

This research is built on two key hypotheses. The first hypothesis is that an accessibility gap exists in augmented reality development between the application of deep learning and the actual needs of AR developers. Deep learning and 3D semantic segmentation algorithms are traditionally developed and optimized by domain-specific researchers. However, AR application development draws a diverse group of software engineers, interaction designers and researchers, who may not have in-depth knowledge of deep learning. The combination of deep learning models trained on indoor datasets with an AR system's 3D real-world object reconstruction could provide developers and researchers with the ability not only to interact with virtual objects, but also to layer interactions onto the real world, expanding the scope for AR interactions and applications.

The second hypothesis suggests that one strategy to bridge this accessibility gap is to create a tool that not only manages expectations but also inspires innovation, thereby promoting a positive developer experience. The tool can stimulate the evolution of developers' projects and enhance the overall development experience by presenting developers with clear expectations of their potential achievements and inspiring them to innovate within their application development workflow. By offering encapsulated deep learning libraries and plugins with comprehensive documentation and tutorials, we can lower the barriers of AR application development for diverse groups and enable them to utilize advanced technologies more effectively. This approach will not only help them meet their goals but also foster a more inclusive and innovative development community.

3.2. Research Question

The primary scenario of this research involves researchers who are starting from scratch to implement real-world object interaction through holograms. By utilizing the toolkit proposed in this project, we envision developers, even those without prior expertise in extracting specific geometric data from their surroundings with deep learning 3D segmentation, will be able to competently handle tasks related to environmental mesh. Thus, the focus of this research is to address the following questions:

1. How can an open source toolkit create a positive developer experience for AR developers?
2. How does having clear expectations of project outcomes and inspiring use of AR technology contribute to the experiences of AR developers?

For a positive developer experience, we anticipate that an open-source toolkit will help developers gain a clear understanding of what they can achieve. This clarity will enable them to plan their project development process more informedly, predict potential challenges, and set achievable goals, thereby improving their efficiency and satisfaction. Also, the inspiration derived from novel approaches to achieving developers' goals can push them to expand the boundaries of their work, thus enhancing their motivation for AR development.

3.3. Research Goal

There are two goals in this research project:

The first objective of this research is to design and implement an open-source toolkit that offers accessible 3D semantic segmentation for AR application development. This toolkit will prioritize elements of expectation and inspiration in its developer experience. Through comprehensive documentation, intuitive functions and APIs, and detailed developer guide, it will improve developers' abilities to realize their conceptual visions. The ambition behind the toolkit is to incorporate advanced computer science technologies, such as deep learning and 3D computer graphics, into the field of augmented reality. We aim to empower AR developers with more accessible technologies.

The second goal is to assess the benefits of expectation and inspiration on AR developers' experiences by observing and reflecting upon their use of this toolkit. Our aim is to get insights into how developers manage and realize their expectations through interaction with this open-source toolkit, and to examine how the toolkit's design and functionality foster inspiration for AR developers. These insights will be beneficial in not only shaping the evolution of our toolkit and ensuring that it continues to meet the demands of the AR development community, but also in informing other open-source projects concerning technology accessibility in AR.

4. Methods

In this chapter, we discuss the motivation behind the design of VisionARy, the considerations for its design and implementation with an emphasis on developer experience, and the structure of the toolkit itself. We also delve into the application development process utilizing the toolkit, assessing how it contributes to expectations and inspiration within the developer experience.

In the beginning, we present preliminary experiments that helped identify the limitations and challenges of the existing usage of indoor spatial data in AR. Building on the insights gained from these experiments, we then formulate specific engineering design decisions that were made in VisionARy with an emphasis on expectation and inspiration. Following this, we will describe the design and implementation of the VisionARy toolkit, and its detailed components including geometric segmentation, data adaptation, machine learning segmentation, AR platform integration, and API development. At the end, a case application is discussed to provide an end-to-end explanation of the VisionARy toolkit's usage.

4.1. Preliminary Experiments

(1) Raw Spatial Mesh in AR application

In our initial experiment, we developed and tested an AR application that facilitates basic interactions with virtual objects in an indoor setting, applying the original mapping capabilities of the HoloLens device. This experiment revealed that while the device could construct an approximate mesh corresponding to the real-world environment and update it in real time, there were several significant limitations. The spatial mesh was coarse and not visually engaging for

users. Users cannot directly interact with the mesh. In addition, the periodic updating process disrupted the smoothness of the user experience. These findings underscored the need for a more accessible solution that would enable AR developers to utilize 3D spatial mesh data. The solution should enhance both visualization and interaction, especially for those developers without experience in 3D computing.

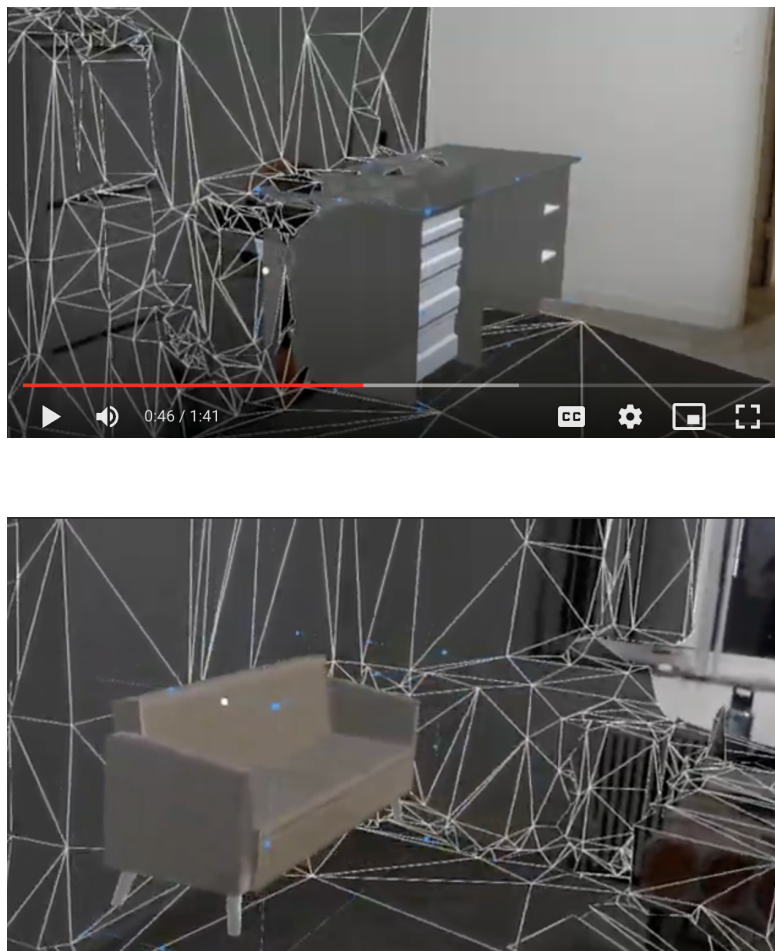


Figure 12. *Raw spatial mesh with virtual furniture objects.*

(2) Pre-processed Geometric Segmentation in AR application

In this experiment, our objective was to implement a path-finding algorithm on spatial meshes and visualize a path guide in an augmented reality view. The first step to complete this task was to perform a complete spatial mesh scan with Hololens headset and download the data using Hololens web portal. Following this, we used a basic geometric segmentation algorithm to identify the floor area within the mesh. The algorithm split the mesh horizontally into square areas of 40cm, then sampled the position data of each vertex within each square area. If vertices were found within one area and the points' height were in reasonable range, the square area was designated as part of the floor. After sampling, the square areas were aggregated to create a floor mesh which was then loaded into the AR application.

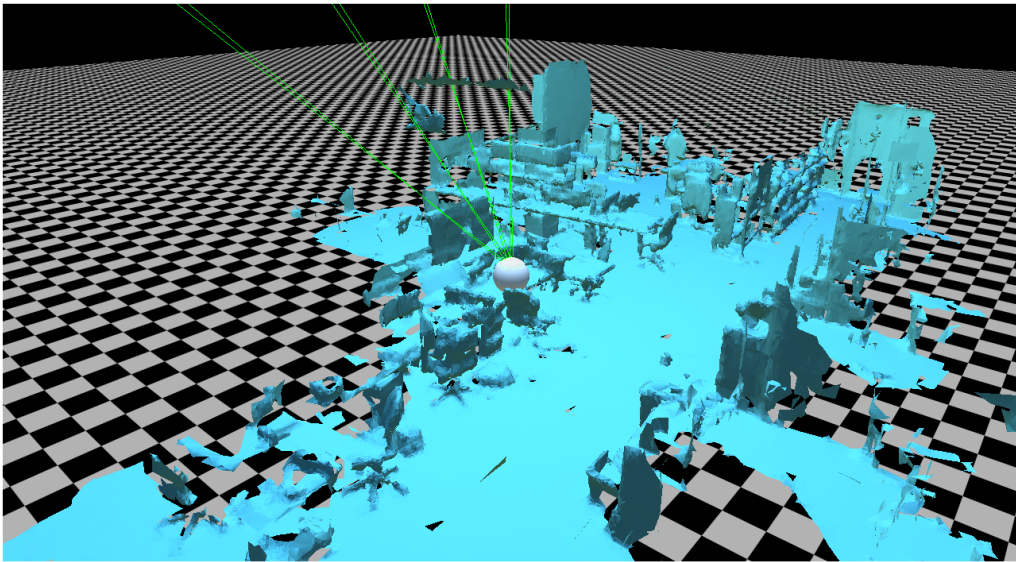


Figure 13. *Spatial mesh in Hololens web portal view.*

To implement a real-time path-finding method on the segmented floor data, we utilized the vertex-link feature of the mesh object. We pinpointed the vertices closest to the target and the user positions as start and ending points, and ran a Dijkstra algorithm on the mesh graph. Upon obtaining a polyline path, we implemented a curve algorithm to generate a smoother path.

Eventually, we created a visualized guide UI to align with the path and aid users. This experiment demonstrated that geometric segmentation can be successfully and effectively applied in AR applications for real-time computing and interaction.

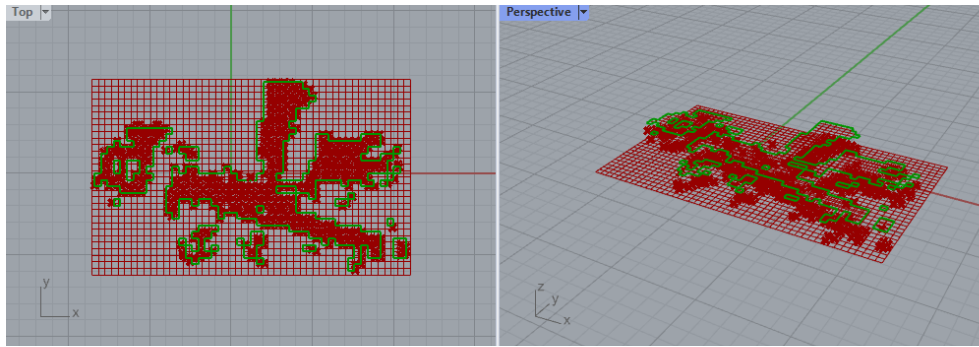


Figure 14. *3D spatial data processing in Rhino.*

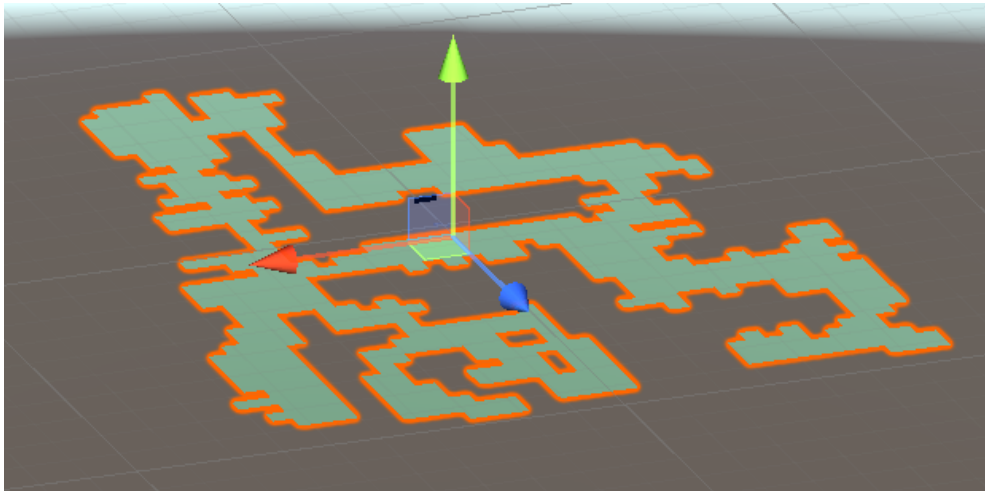


Figure 14. *Apply the processed environment map in Unity application.*

While this technical approach achieves the goal of guiding users in augmented reality through spatial data processing, the development process is complex due to the involvement of various tools and development platforms. It required frequent transitions between the PC and the site to manually update spatial mesh data collection. Additionally, the step of importing mesh

data into the parametric modeling tool, Grasshopper, and running custom 3D segmentation algorithms with it could be a barrier for developers without knowledge of 3D parametric modeling. This reinforces the need for reusable and accessible tools when dealing with 3D segmentation in AR development.

4.1.1 Problem Formulation

Through the preliminary experiments, we identified several limitations in the raw spatial mesh in AR applications. These include a rough and non-user-friendly visualization, a lack of interactivity, and periodic updates that hinder a smooth user experience. In addition, although the second experiment shows the potential of using segmented AR spatial data for new interaction methods and user experiences, it also revealed the challenges faced by developers who may not have experience in 3D modeling software like Grasshopper.

Taking into account the previous observations, accessibility to 3D segmentation techniques is crucial for developers and researchers to maximize the potential of AR devices. The challenges lie in selecting and applying appropriate segmentation algorithms to spatial data based on specific development goals and effectively utilizing the segmentation results.

Therefore, our open-source toolkit with various modules is tailored to address different aspects of using 3D spatial segmentation in Augmented Reality. It simplifies the implementation of geometric and semantic segmentation, as well as offers an approachable entry point for developers and researchers who lack experience in 3D modeling or computer vision. Prioritizing developer experience, the toolkit is designed for flexible and extensible structure, user-friendly APIs and detailed user manual for developers to easily utilize it in their AR projects.

4.2. Developer Roadmap

Providing good developer experience is the primary intention behind the design of VisionARy. In this context, DevEx encapsulates more than just user-friendliness or efficiency. It prioritizes factors that empower developers, such as helping them manage expectations and fuelling their inspiration.

Managing expectations is our prioritized factor to good DevEx. Hence, the VisionARy toolkit is designed with a workflow of two phases that effectively distributes computing tasks between the HoloLens and general computer hardware. This workflow is encapsulated in a modular structure, with each module being an essential unit of the 3D segmentation process for AR. It not only allows developers to understand and experiment with the toolkit with simple modules and their combination outcomes, but also promotes expectation management by providing a clear path towards the creation of AR applications with 3D segmentation. In addition, we provide comprehensive documentation and tutorials to guide developers through the toolkit's integration process. We ensure that the developers are well-equipped to seamlessly combine and coordinate the modules, thereby reinforcing their expectations about the toolkit's functionality and usability.

Also, we prioritize inspiration as an integral part of DevEx. The flexible and extensible structure of VisionARy toolkit encourages developers to approach the AR development process with an explorative mindset. When developers encounter challenges incorporating real-world spatial data into AR applications, the VisionARy toolkit emerges not only as a solution to achieve their goals of spatial data segmentation but also as a source of inspiration. It inspires developers to further advance their projects and their development process, possibly by

extending their project's interactivity or reflecting the structure of their project. The toolkit's roadmap provides developers with a straightforward demonstration of its capabilities, stimulating them to consider the wide-ranging possibilities of AR when combined with 3D geometric computing and the power of deep learning segmentation.

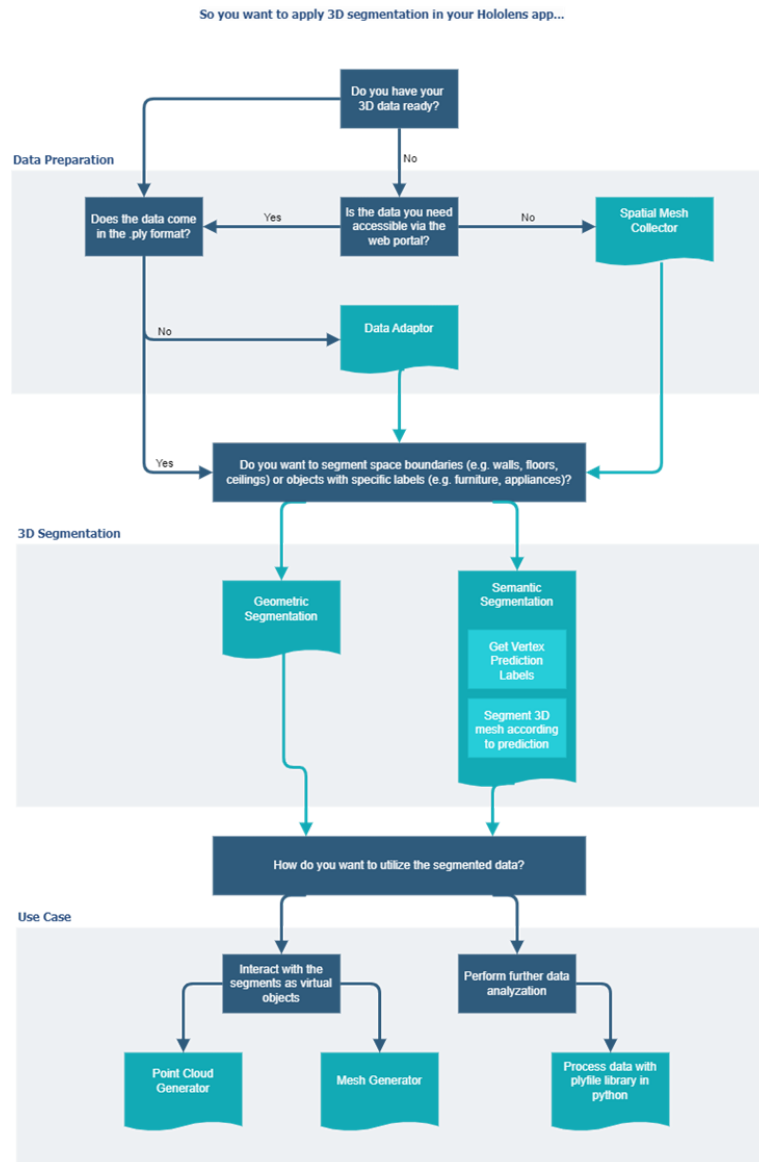


Figure 15. Developer roadmap of VisionARy toolkit.

4.3. VisionARy Toolkit Overview

The primary purpose of VisionARy is to make existing 3D segmentation algorithms and models more accessible to AR developers and researchers, specifically within the context of Hololens development on the Unity3D platform. It aims to bridge the gap between AR applications and existing 3D segmentation methods by dividing potential functionality into several adaptive and easy-to-use modules. This allows developers to flexibly incorporate them into their specific AR projects. The toolkit breaks down the workflow into three phases: retrieving raw spatial data, running segmentation algorithms, and applying the results in AR projects within the Unity platform. The modules of the toolkit include:

1. Real-time raw data retrieving

For certain applications, real-time retrieval of spatial 3D mesh data is required, despite the existing ability to update and download environmental data via the Hololens web portal. To accommodate these needs, our toolkit offers a runtime data retrieval module. This allows developers to deploy the module directly to their device for easy utilization and customization. The module empowers users to scan their surroundings using just the headset and its virtual interface. When the headset is connected to a PC backend, the module can easily transmit the data for further processing.

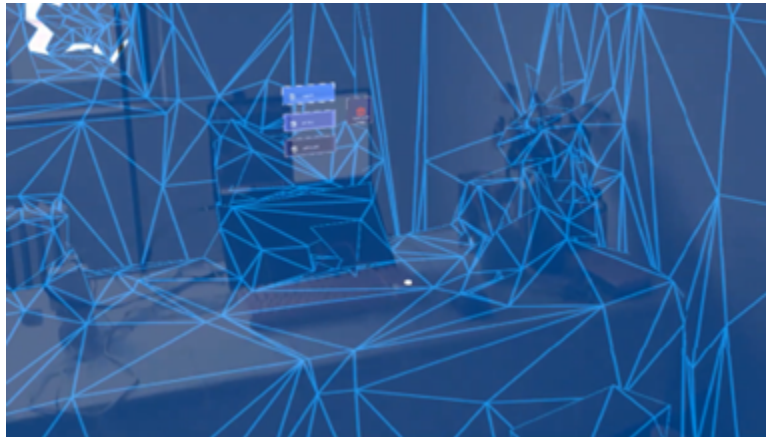


Figure 16. *Augmented interface of the spatial data collector module*

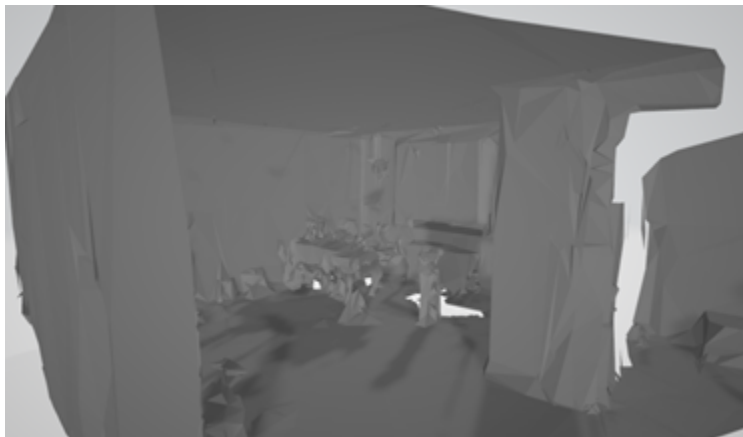


Figure 17. *Real-time data collection results in the local computer.*

2. Geometric segmentation:

This function enables developers to extract simple structures from spatial mesh data. Geometric segmentation is particularly effective for identifying indoor boundaries, such as walls, ceilings, and floors. The mesh-like nature of the HoloLens spatial mesh enables easy utilization of vertices and their connections within spatial data. Furthermore, executing geometric

algorithms for segmentation is more efficient than semantic segmentation, considering the complexity of the algorithm.

3. Data adaptation for machine learning:

The adaptation of raw mesh data collected by HoloLens is vital for running semantic segmentation. This step is necessary because most deep learning datasets and algorithms for semantic segmentation operate on RGB-D video data (e.g., ScanNet) or colored point clouds (e.g., S3DIS). To apply the hologram data to machine learning models, we need to transform the spatial mesh from the HoloLens camera collection and its location data into a format compatible with deep learning semantic prediction. In this toolkit, we leverage the vertex nature of the 3D mesh to convert the raw mesh into a point cloud for semantic segmentation. We then map the predicted labels back to the vertices to segment the 3D mesh, ensuring a smooth process for data adaptation for machine learning.

4. Deep learning segmentation:

The toolkit integrates state-of-the-art pre-trained deep learning based semantic segmentation models into AR applications. The advantage of deep learning segmentation is its ability to recognize and segment objects in indoor environments while providing their semantic labels. For this project, we employ the Mesh3D model, a cutting-edge 3D segmentation model, trained on the ScanNet200 dataset which labels 40 types of indoor elements and supported by comprehensive documentation and user-friendly working examples. This function will enable developers to create 3D virtual representations of real-world objects in the exact same location and scale, assisting in implementing augmented visualization and interaction.

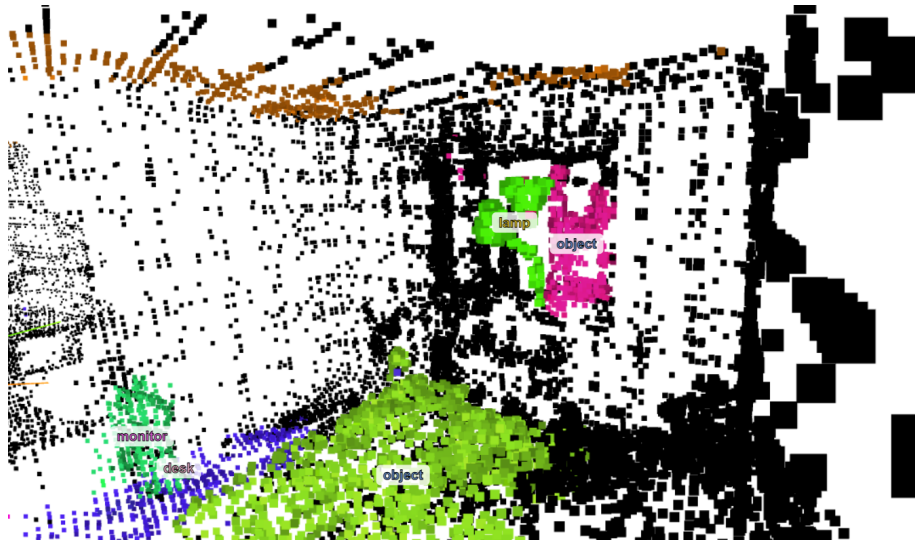


Figure 18. *Vertex label semantic predict result*



Figure 19. *Segmented 3D mesh based on deep learning prediction.*

5. AR platform integration:

We aim to provide seamless integration for applying segmentation results to AR development platforms, specifically for Unity 3D. To accomplish this goal, we offer tools that load the segmented mesh into the development platform in both Unity GameObject and point cloud formats. The objects in the development environment accompanied by a custom shader

script and the augmented reality view can be displayed with personalized materials and colors. The module ensures a consistent and intuitive experience for both developers and users.

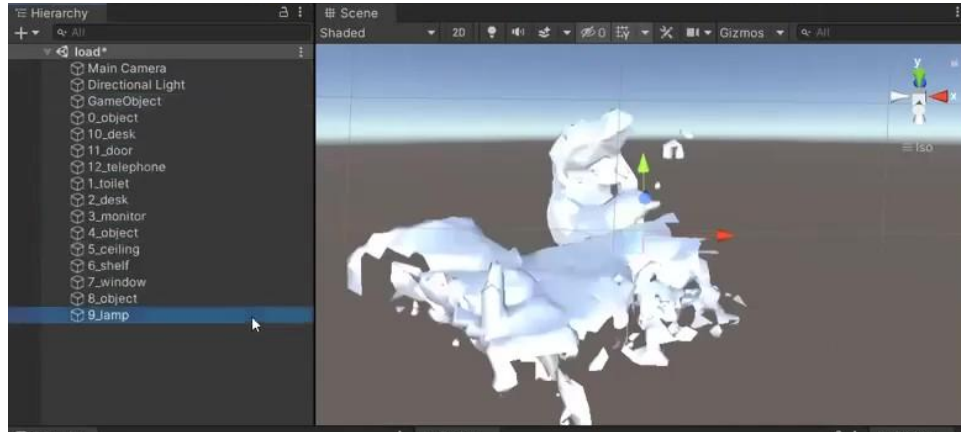


Figure 20. 3D segmentation data in Unity platform.

4.4. Application Case

In this section, we will delve into the development process involved in creating an AR application with real-world hologram interaction. We will follow the software development lifecycle from start to finish, to evaluate how our proposed open-source toolkit influences the developer experience.

Developers begin by reading the documentation to gain a fundamental understanding of the toolkit's capabilities. From there, they choose a path from the roadmap based on their specific development needs and backgrounds. The roadmap outlines the different stages of implementing 3D segmentation in an AR application. It defines specific input and output data formats for each stage and identifies the toolkit modules to carry out the task. Once a roadmap is chosen, developers can follow the steps to segment 3D objects from a scanned real-world environment.

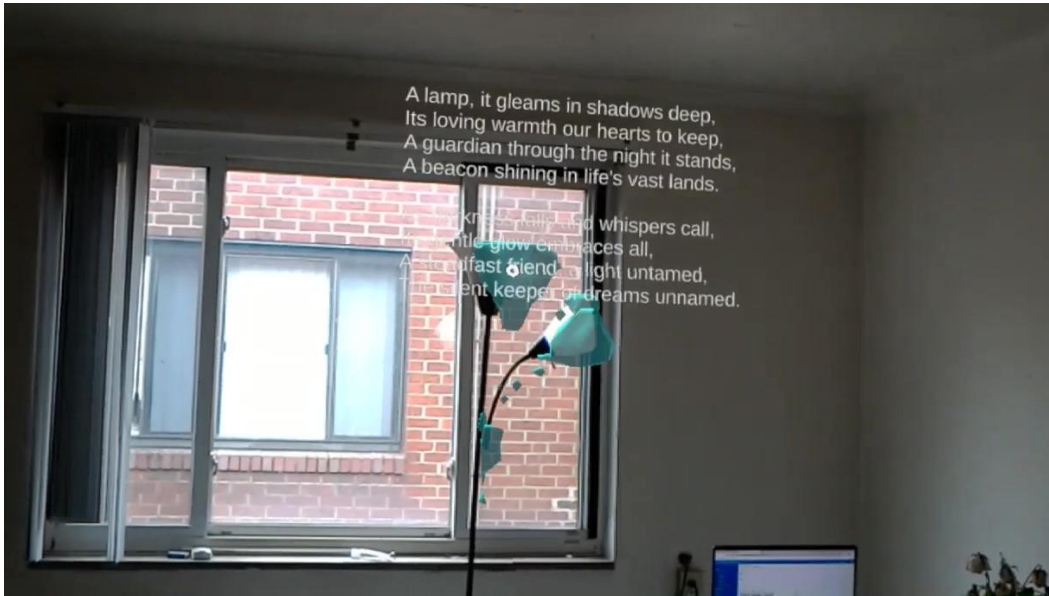


Figure 20. *An AR interaction example of a real-world object.*

In this development process, the developer is aiming at making elements in a living room interactable in augmented reality. To get started, the developer should download the toolkit from GitHub and save it in their working directory. The modules of the toolkit are organized into folders according to their functionalities.

At the beginning of the project, the 3D spatial data of the living room environment is required to be collected. Without the VisionARy toolkit, the developer could potentially use Hololens web portal to gather the surrounding data. This would involve holding the headset to scan the area while clicking the “update” button on the web page to preview the 3D model, until it is able to represent the indoor environment in its entirety. Then, the developer can download a .obj file of the spatial environment from the web portal to the local computer. However, this process is not user-friendly because it requires moving back and forth between the area to be scanned and the computer, and lacks real-time feedback during data collection.

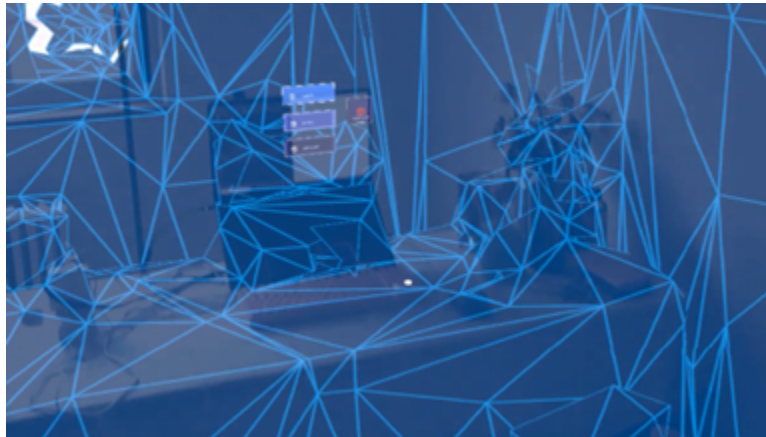


Figure 21. *Augmented interface of the spatial data collector module*

In contrast, the data collection module in the toolkit simplifies this process. Developers can deploy a pre-built AR program to the HoloLens device and set up a local server on a computer using the provided script in the toolkit. The developer can then walk around with the headset and receive real-time, in-place visual feedback on the reconstruction mesh of the surroundings, and transmit the results to the server when the surrounding environment is completely scanned. Although using the toolkit requires some setup, it is manageable for AR developers as the module is pre-built and ready to be deployed to the device. Furthermore, it offers real-time visual feedback and can be seamlessly integrated with other functionalities, such as segmentation and interaction with the segmented results, providing a more friendly and extensible solution for AR application development.



Figure 22. *The entire spatial mesh including the target object (lamp)*

Once the raw spatial mesh is obtained, developers can then use the data adaptation module to convert the mesh into a format that is compatible with the deep learning model. In this prototype, we process the environmental data into a .ply file using a Python script for applying to the Mask3D³ model. This file format contains data of mesh vertices, color, and connective relationships. The toolkit also remains open for future modifications to accommodate various models, hence enabling separate modules to support different 3D segmentation tasks.



Figure 23. *Semantic segmentation result of the environment data.*

³ Mask3D is a transformer-based approach for 3D semantic instance segmentation (Schult et al, 2023).

With the raw spatial mesh now prepared for segmentation, the developer can proceed to execute several Python scripts in the semantic segmentation module for the subsequent tasks. These tasks include optional transformations of the 3D model's y and z axis before segmentation, sending the adapted data to the segmentation server, and retrieving the predicted labels after segmentation completed. The final step is mapping the labels to their corresponding vertices with the sentiments. This allows the reconstruction of separate files for each element as per deep learning predictions, so that separate files for each element can be reconstructed according to the deep learning prediction and be saved to the working directory. In this phase, every task can be accomplished with a single command, and example code snippets are provided for developers' reference to ensure their optimal accessibility .

Without the toolkit, developers might need to search for a dataset and a deep learning solution suitable for their specific needs, learn about open-source models and how to operate them, and resolve data format conflicts by themselves. This process can be time-consuming and demanding for developers without a deep learning background.

The VisionARy toolkit provides a one-stop solution for the developer community. It significantly improves usability and efficiency in the application development lifecycle concentrating on the accessibility of deep learning 3D segmentation for AR developers and researchers. Furthermore, understanding and investigating the toolkit modules can provide developers with no prior knowledge an initial understanding of how deep learning predictions and labels work with 3D object data.

Once the 3D data is segmented for different elements in the environment, developers can import them into their AR projects. For this step, the VisionARy toolkit offers modules to import

the elements into Unity projects as either reconstructed mesh or point cloud objects. By placing the module scripts in the Unity directory's asset path, developers can attach the importing function to any GameObject's inspector. This provides a graphic user interface that allows introducing elements to Unity AR developer platform as needed via their path. The elements are placed in the same coordinates as their original locations. As a result, when developers launch their application, the holograms of the elements will appear in the same place, overlapping with their real-world counterparts. The vertex and connection relationship of the 3D data file are translated into Unity's MeshFilter feature, which allows seamless integration and modification as with any other Unity elements for further interaction or visual effects.

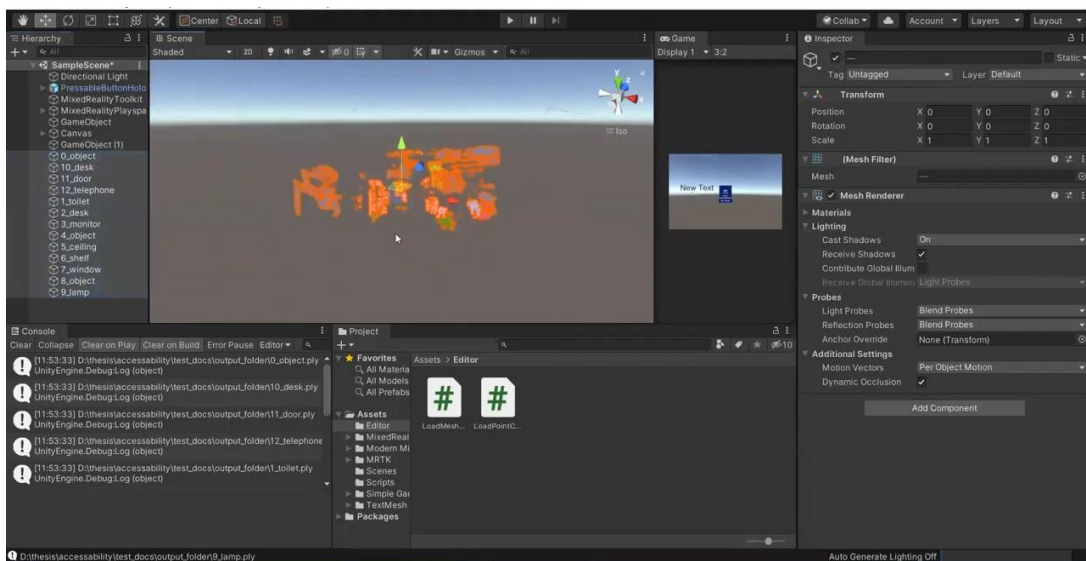


Figure 24. Segmentation result loaded in Unity as point cloud.

In conclusion, the development process with Visionary toolkit is easy to use with simplified steps and expectation management. Even though there are many discussions and proposed solutions about adapting 3D data into Unity projects—despite this not being a native Unity function—the module offered by the VisionARy toolkit stands out for its streamlined

approach and user-friendly design. Accompanied by step-by-step instructions and visual guides, this module provides a seamless transition point to conclude the 3D segmentation aspect in an application development process, making it a robust end-point for the toolkit.

5. Discussions

5.1. Contribution

5.1.1 Easy Onboarding and Adoption

The VisionARy toolkit simplifies the development process for utilizing 3D segmentation in AR applications. Inspired by the MRTK library for Unity3D that provides an easy-to-use workflow to create and deploy AR programs across various platforms and devices, the VisionARy toolkit complements this workflow by integrating seamlessly with Unity3D. It features several Unity Editor scripts that introduce graphical user interface panels in the project inspector section, which allows semantic segmentation results to be loaded into the working scene by inputting the file path with a single click. This process facilitates the integration of segmentation result data into AR application projects without file format conflicts, and these can be applied to interaction events just like any other GameObjects in Unity. Also, the toolkit features a runtime data collector module in the form of a Unity project ready to be deployed to HoloLens for environment data collection and streaming to local computers. This feature allows AR developers, even those with no experience in using the spatial mapping function of HoloLens, to easily retrieve the environment reconstruction data for semantic segmentation. In short, the VisionARy toolkit integrates well into the typical AR application development workflow and is easy to use for developers integrating it into their projects and research.

The data processing and transmission protocols defined in the toolkit modules are easily reusable for developers. For the specific task of running a deep-learning semantic segmentation model on raw 3D meshes, the toolkit includes a module that sends 3D mesh data to a computing

server, retrieving the segmentation results after the task is completed. In the background research, several projects used local computers as servers for computationally heavy tasks, preventing AR devices from becoming overloaded. However, the use of local computers as servers typically needs developers to define their own protocols for data transmission specific to their tasks, which can cause repetitive work across projects. The VisionARy toolkit has pre-defined data transformation protocols for streaming mesh vertex, edge, and label information, inputting mesh data into the Mask3D model for segmentation, and mapping the predicted labels back onto the mesh. This workflow makes it easy for developers to adapt the model to a variety of tasks related to indoor AR. The toolkit is easy to reuse across various versions of the Unity platform or AR devices because only the client-side scripts need modification for adaptation, while the Python script can run on the local server without any changes. The enhancement of code reuse benefits users' adaptation of the project.

The modular structure of the VisionARy toolkit makes it easy for developers to adapt and integrate into their projects. It allows developers to selectively use the modules they need for their specific projects. The developer roadmap provides an intuitive and straightforward guide for developers to learn about and get started with the toolkit. Each module operates independently, allowing developers to modify or even construct additional modules as needed. This modular design makes the toolkit adaptable and customizable to suit different projects. It affords developers the freedom to extend the capabilities of the toolkit, catering to a wide variety of research and application development cases.

Lastly, the detailed documentation and tutorial videos of the toolkit contribute to a positive developer experience by aiding developers in managing their expectations. The

documentation with the roadmap diagram clarifies what can be achieved using the toolkit. It also identifies the best suited modules for constructing a semantic segmentation solution tailored to developers' specific needs. The functionality and operation of each module are clearly explained, complete with sample commands, screenshots, and step-by-step guides. The document helps new users in getting started with the toolkit and setting clear expectations regarding the toolkit capabilities. The tutorial videos for each module guide users from the input stage to the output result, offering developers a reliable reference and assistance in troubleshooting. With this comprehensive documentation, developers can smoothly start using the VisionARy toolkit and fully utilize its functionalities to reach their objectives.

5.1.2 Seamless Integration of Technologies

The VisionARy toolkit offers a comprehensive solution for integrating 3D semantic segmentation into AR application development workflows. Not only does it grant access to cutting-edge deep learning technologies, but it also manages the process of data format transformation for seamless integration. This means developers can conveniently obtain what they need from the toolkit without any additional steps.

Data in this integrated workflow can be transferred and processed seamlessly among the modules. The modular structure ensures that the output of one module can directly serve as the input for the subsequent one. This setup effectively encompasses the entire process from data retrieval and adaptation, through semantic labeling on 3D mesh vertices and 3D model segmentation based on these labels, to the final process of importing 3D models into Unity projects. Furthermore, the Unity editor script⁴ provided by the toolkit enables graphic user

⁴ A Unity editor script is a custom script designed to extend the Unity editor interface for ease of use.

interfaces to import data to project easily, hence ensures smooth integration of the toolkit within the Unity Engine.

The inclusion of a runtime client-server module enhances the integration of various technologies. By delegating compute-intensive deep learning tasks to a local computer server and implementing raw data and segmentation results transfer in a AR device-side client program, the AR device can free up more resources to support smooth graphics and interaction. The core computations of the toolkit take place on the local computer, ensuring its continued relevance and effectiveness despite potential updates to the Unity3D platform, thus preventing it from becoming outdated.

5.1.3 Encouragement of Innovation

Innovation in the context of VisionARy toolkit refers to reflecting and exploring the AR development process and bringing together knowledge from various fields. The toolkit's modular design and roadmap inspire developers to critically assess their development process and experiment with different strategies for various projects. The structure of this toolkit allows developers to contemplate the extension of workflow modules easily and encourages them to create new roadmaps for improving efficiency and productivity. Developers engage with the entire development process from initiation to implementation using the toolkit. The roadmap encourages thoughtful consideration of each step along with module selection throughout the project. This clear pathway supports systematic exploration and optimization of the development process. The flexibility and extensibility of the modular structure promote an environment of continuous learning and evolution for developers. The VisionARy toolkit is regarded not only as

a tool, but also a motivator for innovative thinking in AR development that encourages developers to think beyond traditional development processes.

For interdisciplinary engagement, the toolkit fosters innovation by encouraging collaborations and cross-pollination of ideas in the fields of deep learning, spatial computing, 3D computer graphics and more. The VisionARy toolkit benefits not only those developers who may not frequently use deep learning but also interactive designers who may lack specific coding skills. This broad accessibility opens up more opportunities for creating immersive and interactive experiences in AR applications. By facilitating the segmentation of real-world 3D data and its application in AR experience, developers and researchers can explore interactions with reconstructed real-world objects within the augmented environment, rather than merely interacting with virtual entities. The VisionARy toolkit not only facilitates registering holograms in reality, but also connects the semantics between real-world objects and their virtual representations, hence supporting more immersive augmented reality interactions. Also, the modules within the VisionARy toolkit are designed to be easily customizable and adaptable, inviting developers to contribute their unique modules or feedback. The design has the potential to foster an active and collaborative community, continually enriching the toolkit for broader use cases.

5.2. Limitation

The spatial mesh resolution offered by the current prototype of the toolkit is relatively low, despite its ability to be segmented into game objects that represent real-world entities. While this resolution is sufficient for the deep learning models to identify and label its semantics, the final visualization quality in the AR view can still be improved. This issue comes with the

resolution constraints of the Hololens spatial mapping capabilities. Thus, the current version of the toolkit is more suitable for introducing interactions with real objects rather than offering high-quality visual effects to users.

Another limitation concerns the choice of the deep learning model and dataset. The currently utilized ScanNet200 dataset and the Mask3D model are developer-friendly, with its well-organized documentation and a demonstrative server. However, they might not be the best fit for various types of 3D segmentation tasks. To further enhance the utility of the VisionARy toolkit, we could optimize it to adopt a plug-and-play approach for incorporating different deep learning models in the future.

5.3. Future work

Going forward, there's potential to optimize the VisionARy toolkit's semantic segmentation module. The goal is to make it adaptable to a variety of 3D segmentation models and algorithms, opening up the field for the integration of newer deep learning networks. This adaptability would be beneficial to developers who prefer using specific segmentation algorithms for their projects.

Moreover, it's essential to consider broadening the toolkit's reach by supporting different versions of development platforms and environments. Currently, our research uses Hololens gen 1, compatible with Unity versions up to 2019. But as more developers and researchers are venturing into using Hololens gen 2 and newer development platforms, the toolkit should be adaptable to these environments as well.

Finally, we can provide regular updates, fixes, and improvements for VisionARy to become a community-driven open-source project. By collecting feedback from developers and researchers in the open-source community, we can continue to evolve the project, keep it updated, and encourage more developers to utilize and contribute to VisionARy.

Looking ahead, the evolving trends in Augmented Reality emphasize immersion, user presence, and spatial computing in various application scenarios. We are confident in the continued relevance of the VisionARy project for the following reasons: From the developer's perspective, the toolkit is designed to adapt to future advancements in deep learning and spatial computing, while maintaining its flexibility for developer customization. From the user's perspective, the toolkit feature's ability to update surrounding environment reconstructions in real-time makes it advantageous for responding to changing environments, thus enhancing user presence simultaneously in immersive and real-world environments.

6. Conclusion

In conclusion, the VisionARy toolkit effectively improves the developer experience by prioritizing developers' expectation management and inspiration in the AR development process. It is easy to use with its comprehensive documentation and seamless integration with the existing technology stack of AR applications. The toolkit simplifies the onboarding and adoption processes, streamlines technological integration, and fosters innovation.

From the perspective of developer experience, the VisionARy toolkit benefits developers in managing their expectations with its structured workflow and comprehensive documentation. This setup provides developers with a clear understanding of what they can achieve using the toolkit and how to build their AR applications to suit their specific goals. In addition, the flexible and extensible structure of VisionARy stimulates the creation of new interaction methods in AR and encourages innovation in the development process. The VisionARy toolkit is able to incorporate knowledge from various fields.

The AR application development process with the VisionARy toolkit is designed to be easy. To facilitate onboarding, the toolkit offers a range of sample videos and a clearly outlined roadmap, equipping developers with the resources they need to accomplish their goals. For technology integration, each module can be applied with just a single command or one click. This streamlined process significantly reduces activities that add no value, saving time and effort. Moreover, by providing easy access to advanced technologies and expedited prototyping, the toolkit encourages innovation and paves the way for developers to explore new opportunities.

In closing, the VisionARy toolkit, born from a desire to bridge gaps and empower developers, has demonstrated the potential of an open-source toolkit in enhancing the developer experience. As technology advances at an increasingly rapid pace, and as enthusiasm for these technologies grows among people from diverse backgrounds, we recognize the importance of fostering a more inclusive and dynamic AR community. Our aspiration is to shape a landscape in which everyone has the opportunity to both benefit from and contribute to the wonderful technology of Augmented Reality.

Reference

- Agrawal, D., Lobsiger, J., Bo, Y. F., Kaufmann, V., & Armeni, I. (2022). HoloLabel: Augmented reality user-in-the-loop online annotation tool for as-is building information. In Proceedings of the 2022 European Conference on Computing in Construction (Vol. 3). University of Turin.
- Azuma, R. T. (1997). A survey of augmented reality. Presence: teleoperators & virtual environments, 6(4), 355-385.
- Björk, N. (2020). Simple feature detection in indoor geometry scanned with theMicrosoft Hololens.
- Borouhaki, T., Lam, M., Dodds, L., Eid, A., & Adib, F. (2023). Augmenting Augmented Reality with {Non-Line-of-Sight} Perception. In 20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23) (pp. 1341-1358).
- Doughty, M. HololensForCV-Unity. Updated August 13, 2020. Accessed August 1, 2023. <https://github.com/doughtmw/HoloLensForCV-Unity>.
- Fagerholm, F., & Münch, J. (2012, June). Developer experience: Concept and definition. In 2012 international conference on software and system process (ICSSP) (pp. 73-77). IEEE.
- Hübner, P., Clintworth, K., Liu, Q., Weinmann, M., & Wursthorn, S. (2020). Evaluation of HoloLens tracking and depth sensing for indoor mapping applications. Sensors, 20(4), 1021.

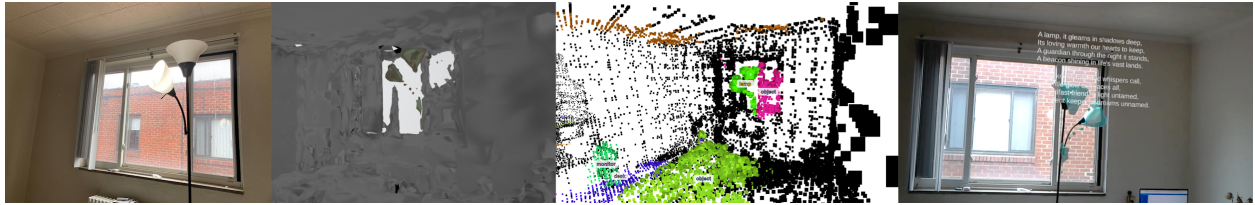
- Mekni, M., & Lemieux, A. (2014). Augmented reality: Applications, challenges and future trends. *Applied computational science*, 20, 205-214.
- Microsoft. Spatial awareness getting started. Updated August 1, 2022. Accessed August 1, 2023. <https://learn.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/mrtk2/features/spatial-awareness/spatial-awareness-getting-started> .
- Microsoft. (accessed Apr. 09, 2023). Developer Experience (DevEx) - Code With Engineering Playbook. <https://microsoft.github.io/code-with-engineering-playbook/developer-experience/> .
- Milgram, P., & Kishino, F. (1994). A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems*, 77(12), 1321-1329.
- Rozenberszki, D., Litany, O., & Dai, A. (2022, October). Language-grounded indoor 3d semantic segmentation in the wild. In *European Conference on Computer Vision* (pp. 125-141). Cham: Springer Nature Switzerland.
- Schrempp, Y., Zhu, T., & Heinz, E. Augmented Reality for Users with Low Vision.
- Schult, J., Engelmann, F., Hermans, A., Litany, O., Tang, S., & Leibe, B. (2023, May). Mask3D: Mask Transformer for 3D Semantic Instance Segmentation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 8216-8223). IEEE.
- Schütt, P., Schwarz, M., & Behnke, S. (2019, September). Semantic interaction in augmented reality environments for microsoft hololens. In *2019 European Conference on Mobile Robots (ECMR)* (pp. 1-6). IEEE.

Ungureanu, D., Bogo, F., Galliani, S., Sama, P., Duan, X., Meekhof, C., ... & Pollefeys, M. (2020). Hololens 2 research mode as a tool for computer vision research. arXiv preprint arXiv:2008.11239.

Appendix

Appendix A. VisionARy User Manual

VisionARy: An Open Source 3D Environment Segmentation Toolkit for Augmented Reality



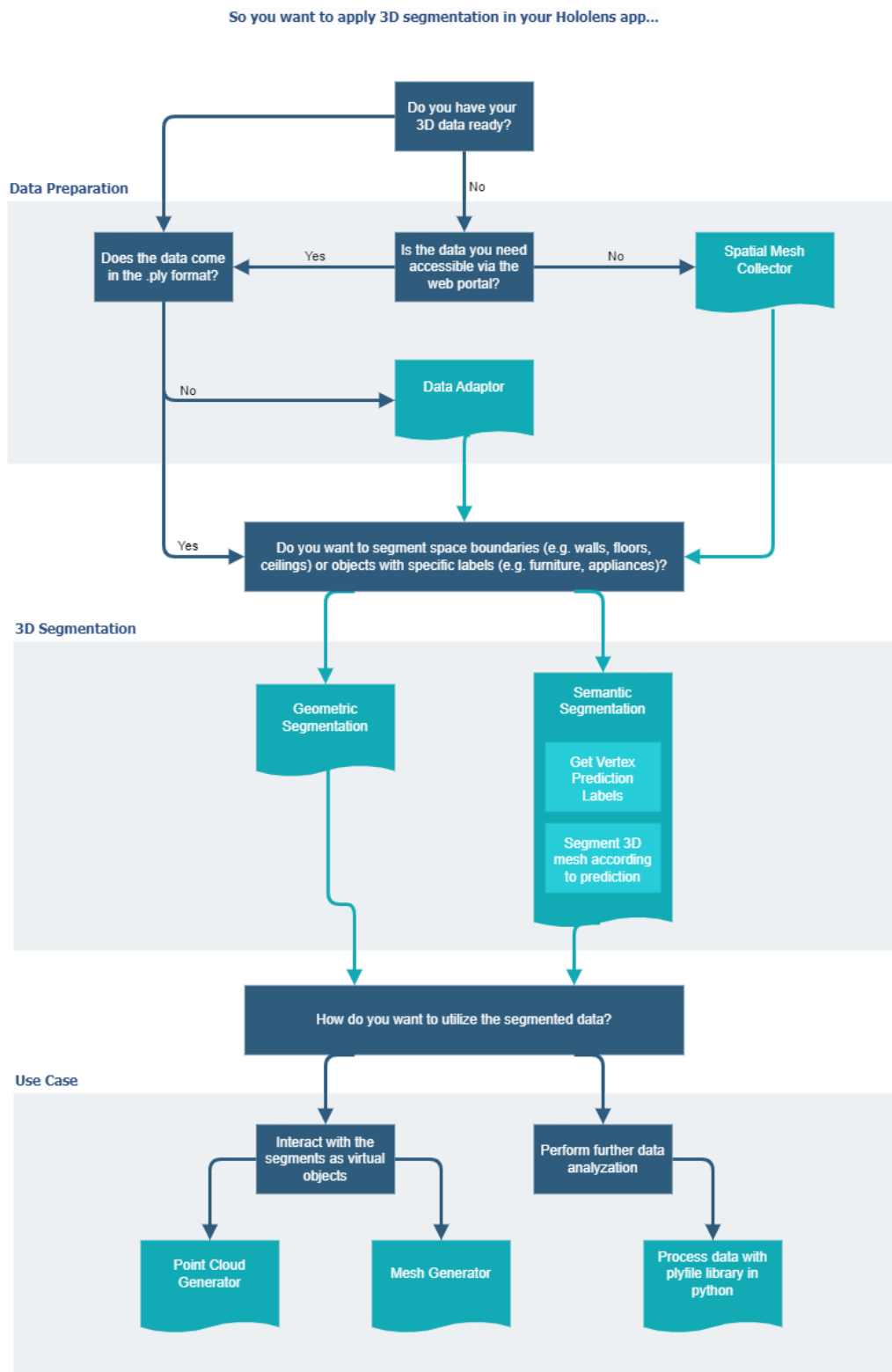
VisionARy is an open-source toolkit designed to make 3D indoor segmentation technologies accessible to Augmented Reality (AR) developers and researchers. With a strong emphasis on developer experience, the toolkit expands the potential of AR spatial mesh in conjunction with 3D segmentation, providing various modules for geometric and semantic segmentation, data adaptation, AR platform integration, and APIs.

Getting Started

Download the Toolkit

```
$ git clone https://github.com/yuechenw213/VisionARy.git
```

Get Your Developer Roadmap



Prepare your data

- If you already have 3D mesh data that you want to segment, make sure it is in .ply file format. If not, you will need to convert it to this format.
- You can also obtain a spatial mesh from the Hololens web portal, which will be in .obj file format. To use this mesh for segmentation, you will need to convert it to .ply format. The Data Adaptor module can assist you in converting the data.
- If the data is not still available, you can use the Spatial Mesh Collector module to collect spatial mesh data in real-time. To do this, deploy a Unity app with the provided package to the Hololens device, and set up the Python server provided in the module. The app will send the captured mesh data to the backend, where it will be saved as a .ply file.

Perform 3D segmentation

- To perform 3D segmentation, first determine what you want to segment from the data.
- Geometric segmentation involves segmenting based on the geometry and topology of the 3D mesh. This approach is useful for segmenting space boundaries and other geometric features.
- Semantic segmentation involves segmenting based on the labels or categories of objects in the 3D mesh. This approach is useful for segmenting specific types of objects (e.g. furniture) or features (e.g. windows, doors).
- Follow the instructions provided in the module to perform the segmentation.

Apply Segmentation Result to Your Application

- The segmented result is provided in the form of multiple .ply files, each representing a segment of the 3D mesh.
- If further data processing is needed, you can use the Python code and the plyfile library to access the features of the segmented data.

- To apply segmentation to your AR application, modules are provided in the toolkit to load the segmented result into the Unity engine. The segmented data can be loaded as a mesh object or point cloud, depending on the specific needs of your application. The module also provides a batch loading function for loading multiple segments at once.

Modules

1. Spatial Mesh Collector

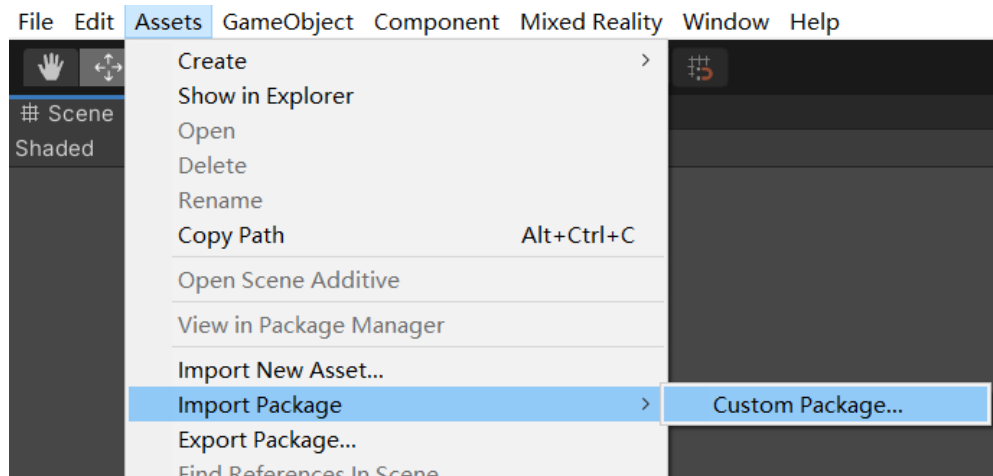
- This module is designed to facilitate the collection of 3D mesh data from real-world environments. The module provides a runtime data collection Unity package and a local TCP server.

Python Server

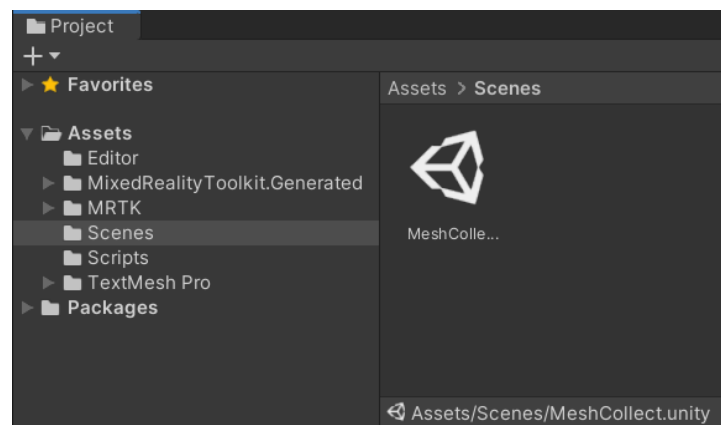
- The Python server receives data from the Hololens client and saves it locally in .ply format. It acts as a bridge between the client and backend processing tools, allowing for real-time data capture and processing.
- Setup:
 - In `./SpatialMeshCollector/server.py`, set the `HOST` constant to your local ip address. Set the output constant to your designated filepath.
 - Run `./SpatialMeshCollector/server.py` to start receiving connections.
 - When the 3D spatial data is received and saved, the server will stop automatically.

Unity Package

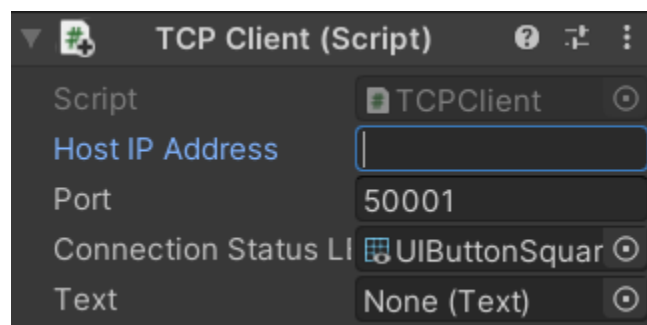
- Setup:
 - In a unity project, use `Assets > Import Package > Custom Package` to import the `VisionARy_MeshCollector.unitypackage` for collecting and sending 3D spatial mesh.



- Then in the Project window, under Asset > Scenes, open the MeshCollect.unity scene.

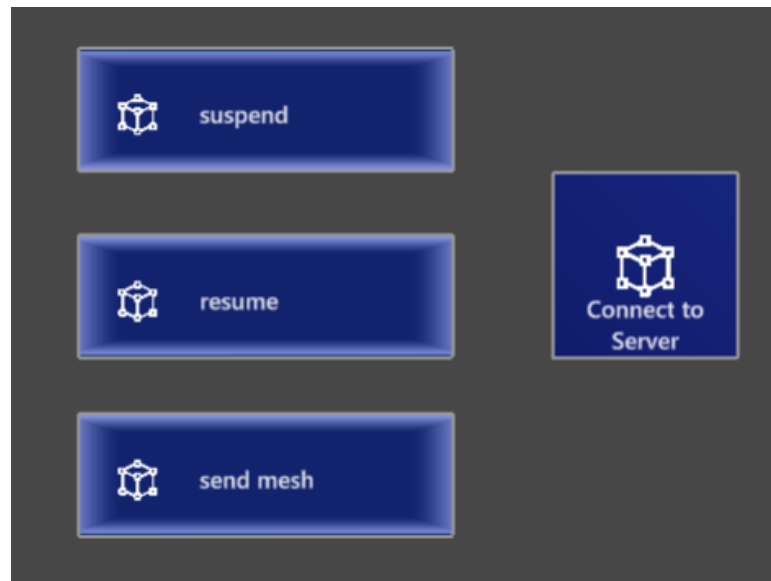


- In the TCP client game object, set the Host IP address to your python server IP.



- Save the change, deploy the scene to your Hololens.

- In the Application:
 - You will see the following AR interface. Use the Connect to Server button to connect to the python server. When success, the button will turn green.



- Use the headset to scan your environment. The suspend and resume button can control the update of spatial mesh.
- When the immersive environment is well covered by mesh, use send mesh to save the mesh data to the server.

2. Data Adaptor

The Data Adaptor converts 3D models in OBJ format to PLY format for segmentation.

- cr: This code is developed based on nabeel3133's file-converter-.obj-to-.ply repository, which can be found at <https://github.com/nabeel3133/file-converter-.obj-to-.ply>

How to use

```
$ python ./DataAdaptor/convert.py --input [objfilename.obj] --output [plyfilename.ply]
```

3. Geometric Segmentation

One sample task that can be performed using the Geometric Segmentation module is the segmentation of the ceiling and floor from 3D mesh data. By using algorithms to identify the horizontal surfaces in the mesh, the module can isolate the ceiling and floor segments and generate separate .ply files.

How to use

```
$ python ./GeometricSegment/Geometric.py --mesh_file [yourmesh.ply] --threshold
[threshold] --output_file [plyfilename.ply]
```

4. Semantic Segmentation

The Semantic Segmentation module is designed to enable segmentation of 3D mesh data based on semantic labels or categories. This module utilizes a Mask3D (<https://github.com/JonasSchult/Mask3D>) model and the ScanNet200 dataset to perform segmentation.

How to use

By making HTTP requests to the Mask3D demo backend(<https://francisengelmann.github.io/mask3d/>), the Semantic Segmentation module obtains predictions of vertex of indoor data from the Mask3D server in single command.

```
$ python .\SemanticSegment\segment_via_HTTP.py --mesh_file [your_mesh.ply]
--output_folder [output_folder]
```

This command will obtain the prediction maps in plain text format. The following command can then be used to map the prediction back to the original mesh.

- for getting all labeled instances from the result
 - \$ python .\SemanticSegment\SegmentationMap\semantics_to_mesh.py

--pred_zip [filename.zip] --mesh_file [your_mesh.ply] --output_folder

[output_folder]
- for getting single instance of your choice

- `python .\SemanticSegment\SegmentationMap\visualize_labels_on_mesh.py --pred_file [predict.txt] --mesh_file .[your_mesh.ply] --output_file [output.ply]`
- or `$ python .\SemanticSegment\SegmentationMap\visualize_labels_on_mesh.py --pred_file [predict.txt] --mesh_file .[your_mesh.ply] --output_file [output.ply]`

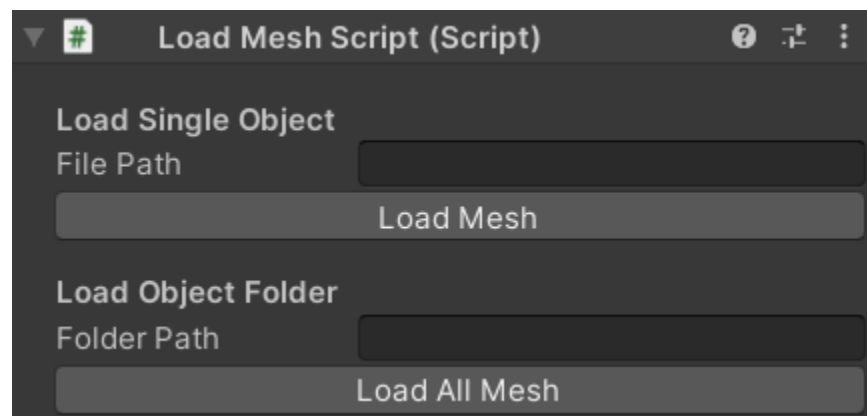
5. Segmented Mesh Generator

This module helps you load segmented data into the Unity development platform with just one click.

How to use:

```
$ cp .\UnityPlugins\editorMeshLoad\. [your_project\Assets\Scripts\]
```

Then add the script to any gameobject.

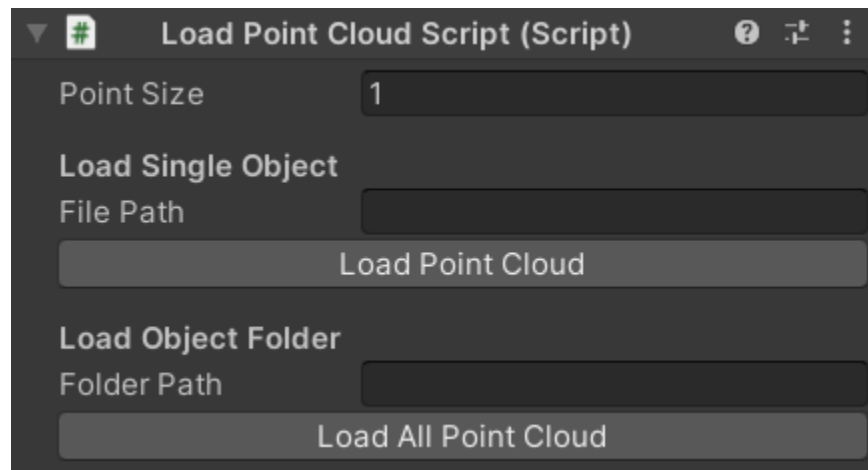


You can put the path to single .ply file or a folder containing multiple files to the corresponding box, and click the button to load it in the scene.

6. Point Cloud Generator

The usage of point cloud generator is similar to mesh generator

```
$ cp .\UnityPlugins\editorPointCloudLoad\. [your_project\Assets\Scripts\]
```



Appendix B. Video

The tutorial and example video of VisionARy is accessible at:

<https://www.youtube.com/playlist?list=PLrYVDrNwUVZA4m67fYLc22FMmgH0v85ib>

Appendix C. Source Code

The source code of VisionARy is accessible at: <https://github.com/yuechenw213/VisionARy>