Room Layout Estimation in AR

Panagiotis Minos[1], Pranjal Mishra[1], Heng Zhao[2], Ziqi Zhang[3]
[1] Department of Mathematics, ETH Zurich
[2] Department of Information Technology and Electrical Engineering, ETH Zurich
[3] Department of Mechanical and Process Engineering, ETH Zurich
pminos@ethz.ch, pmishra@ethz.ch, zhaohe@ethz.ch, ziqzhang@ethz.ch

Abstract

*This work presents an end-to-end pipeline for room layout estimation in augmented reality (AR) environments, specifically leveraging the Magic Leap 2 platform. By integrating lightweight, cuboid-based 2D boundary segmentation (**ST-RoomNet**), refined 3D line generation, and a user-centric rendering approach, the proposed method achieves both robust performance and intuitive user interaction. Experimental results demonstrate the system's ability to reduce clutter and enhance structural clarity through distinct 2D and 3D line refinement modules. Additionally, the pipeline incorporates an ergonomic user interface design, ensuring a comfortable and seamless user experience. Ultimately, the findings highlight the potential of combining advanced AR hardware with meticulously crafted algorithms to enrich users' perception of indoor spaces. This work lays the groundwork for future research in areas such as long-range depth sensing, sophisticated room layout models, and the development of more user-friendly AR interfaces.*

1. Introduction

Accurate room layout estimation is pivotal for a wide array of applications, including robotics, architectural design, and, notably, AR. In AR, precise understanding of indoor spatial configurations is essential for seamlessly overlaying virtual objects, enhancing user immersion, and facilitating intuitive interactions. Traditionally, generating three-dimensional layouts of indoor environments has relied heavily on panoramic cameras or single images with complex processes such as image segmentation followed by depth estimation. While these methods can be effective in controlled settings, they are often cumbersome and computationally intensive, limiting their applicability in dynamic environments.

The advent of advanced AR hardware, particularly the Magic Leap 2 platform, marks a significant advancement in room layout estimation. Unlike traditional single-image or panoramic-based approaches, Magic Leap 2 is equipped with sophisticated depth sensing capabilities that provide rich and accurate depth data in real-time. This inherent depth information obviates the need for separate depth estimation processes, thereby streamlining the workflow for room layout estimation and enabling faster and more efficient reconstructions.

In this work, we leverage the depth data provided by Magic Leap 2 to develop an efficient pipeline for room layout estimation in AR environments. Our approach integrates lightweight, cuboid-based 2D boundary segmentation using **ST-RoomNet**, refined 3D line generation, and a user-centric rendering methodology. By utilizing the inherent depth information from Magic Leap 2, we bypass the computationally intensive steps associated with traditional depth estimation, resulting in faster and more accurate room layout reconstructions.

The key contributions of this paper are as follows:

- **Utilization of Magic Leap 2's Depth Data**: We directly employ the depth sensing capabilities of Magic Leap 2 to simplify the room layout estimation pipeline, reducing dependency on separate depth estimation models and enhancing processing speed.

- **Efficient 2D Boundary Segmentation with ST-RoomNet**: We implement ST-RoomNet for segmenting room boundaries from single images, leveraging its alignment with cuboid models to achieve high segmentation accuracy in standard indoor environments.

- **Dual-Stage Line Refinement Modules**: Our method incorporates separate refinement stages for both 2D and 3D line generation. In the 2D domain, morphological operations and orientation-based grouping streamline the detected lines. In the 3D domain, robust line fitting and spatial clustering techniques further refine the layout representation, enhancing structural clarity and reducing clutter.

- **Customizable and User-Friendly Interface**: To enhance user experience in AR applications, we provide a customizable interface that allows users to define the format of generated lines according to their needs. Additionally, a built-in user help menu offers guidance, ensuring smooth and intuitive interaction with the estimated room layouts.

Our experimental evaluations demonstrate that the proposed pipeline effectively reduces noise and accelerates processing, achieving a balance between accuracy and performance. However, the Magic Leap 2's depth camera range of approximately seven meters and ST-RoomNet's reliance on cuboid assumptions present certain limitations, such as challenges in corridor-like or irregularly shaped environments. Nonetheless, the overall system showcases strong potential for standard room layouts.

This paper is organized as follows: Section 3 details our methodological framework, encompassing 2D segmentation, 3D line generation, and rendering techniques. Section 4 describes our experimental setup and presents the results, including comparative analyses and user study feedback. Finally, Section 5 concludes the paper by discussing the implications of our findings and outlining directions for future research.

By harnessing the advanced depth sensing capabilities of Magic Leap 2 and integrating them with efficient segmentation and rendering algorithms, our work advances the field of room layout estimation. This synergy not only enhances technical performance but also delivers a user-centric AR experience, paving the way for more immersive and intelligent indoor mapping applications.

2. Related Work

2.1. Single Image Room Layout Estimation

Single image room layout estimation is a critical task for indoor scene understanding, enabling applications in 3D reconstruction, augmented reality, and navigation. Traditional methods, such as the seminal work by Hedau et al. [2], utilized vanishing points and geometric cues to generate layout hypotheses. However, these approaches struggled with occlusion and clutter, and their reliance on handcrafted features limited their adaptability.

Recent advances in deep learning have transformed this field. RoomNet [5] reformulated layout estimation as a keypoint regression task, leveraging an encoder-decoder architecture to predict room layouts directly. Similarly, LayoutNet [9] integrated multitask learning to jointly estimate room corners and boundaries, optimizing for Manhattan layout constraints. ST-RoomNet [3] introduced an unsupervised spatial transformation network to map a reference cuboid layout to the target layout, achieving high efficiency without requiring segmentation or keypoint estimation.

Other notable contributions include Zhang et al. [8], who proposed a deconvolution network for high-quality edge map prediction, facilitating layout generation through adaptive sampling. Pano2CAD [7] extended layout estimation to panoramic images, relaxing the box-shaped room assumption and integrating 3D object pose estimation for enhanced scene understanding. Lin et al. [6] further improved real-time layout estimation with a single-stage, end-to-end pipeline, introducing novel augmentation and training strategies to enhance model generalization.

Together, these methods highlight a paradigm shift from heuristic-driven pipelines to data-driven, neural network-based approaches. They address challenges such as occlusions, clutter, and diverse room topologies, demonstrating significant advancements in both accuracy and computational efficiency.

2.2. Magic Leap 2 and its Features

Magic Leap 2 represents a contemporary AR device that integrates high-resolution RGB imaging with precise depth sensing, thus offering a streamlined alternative to traditional single-image or panoramic-based pipelines. Its lightweight form factor, larger field of view (up to 70° diagonal), and Time-of-Flight (ToF) sensor make it well-suited for spatial tasks. Unlike external depth estimation algorithms that introduce additional overhead, Magic Leap 2's dedicated hardware directly provides dense 3D information of the environment in real time.

The device's RGB camera captures up to 4K video at 30 frames per second, essential for segmentation-based tasks, while the ToF sensor generates over 300,000 3D points with millimeter-level accuracy, covering a range up to approximately 7.5 meters. Additionally, Magic Leap's Android-based operating system, SDKs, and support for platforms like Unity facilitate rapid prototyping of AR applications, allowing developers to leverage the depth data without cumbersome preprocessing steps.

In the context of room layout estimation, Magic Leap 2's integrated depth sensing obviates the need for separate monocular or panoramic inference pipelines. This not only expedites the layout reconstruction process but also increases robustness when dealing with occlusions or partial views. Its WebSocket support further enables real-time communication between the headset and a remote server, ensuring continuous data flow for immediate visualization and refinement of the estimated layouts.

3. Method

This section details the methodology for room layout estimation in an augmented reality environment, structured around three primary components: **2D Line Detection with Segmentation**, **3D Line Generation**, and **Rendering**, which are illustrated in Figure 1. Each component employs

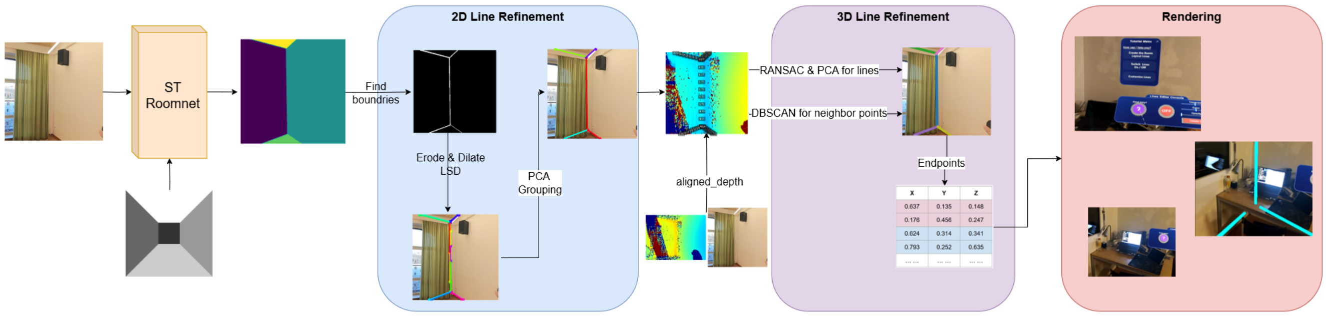

Figure 1. Pipeline for room layout estimation in Augmented Reality with Magic Leap 2: segmentation via ST-RoomNet, 2D and 3D line refinement, depth Integration, and rendering.

specific processes and techniques that collaboratively enable the accurate reconstruction and visualization of indoor layouts. The subsequent subsections provide an in-depth description of these components.

3.1. 2D Line Detection with Segmentation

Detecting room boundaries in two-dimensional images is a critical step for establishing a robust structural framework before depth alignment and three-dimensional rendering. In this study, we adopt **ST-RoomNet** for segmentation and boundary extraction.

ST-RoomNet employs a spatial transformation network to align detected layouts with a reference cuboid model. This approach delivers high segmentation accuracy in environments with cuboid-like structures (e.g., the top two rows of Figure 2), where room geometries closely adhere to the reference model. However, in non-cuboid settings (e.g., corridors) or partial scenes (e.g., isolated corners), the model faces challenges in capturing layouts accurately, often resulting in imprecise or incomplete segmentations (see the bottom two rows of Figure 2). Given our focus on predicting room layouts that primarily conform to cuboid references, we selected ST-RoomNet for its lightweight design and rapid processing capability.

Following the initial room layout segmentation, extracting and refining structural boundaries is essential for generating a three-dimensional geometric representation. This process begins with boundary extraction from the segmentation output, utilizing connectivity-based methods to delineate boundaries between segmented regions. Morphological operations, specifically dilation and erosion, are applied to enhance the boundaries and reduce noise.

Subsequently, the Line Segment Detector (LSD) is employed to extract two-dimensional line segments directly from the refined boundary image. Each detected line segment is defined by its endpoints, offering a detailed geometric representation of the room's layout. However, the algorithm often identifies numerous small line segments (see the first and second columns of Figure 3), which can lead to cluttered results during subsequent rendering stages.

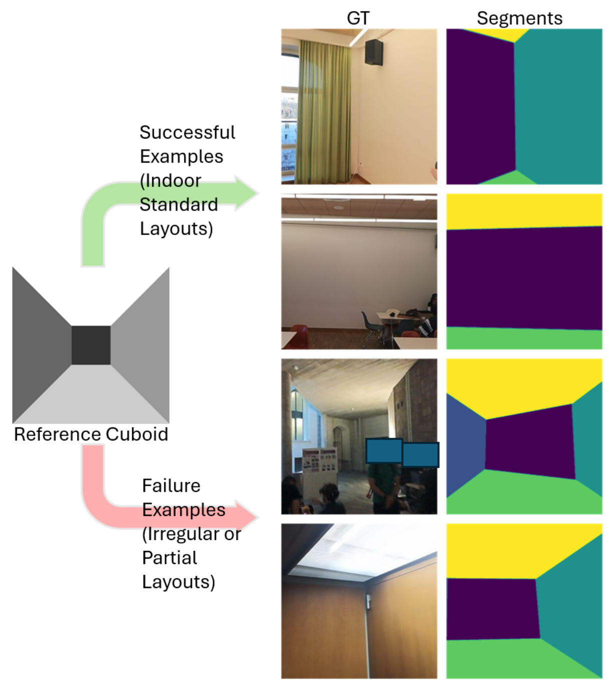

Figure 2. Segmentation results of ST-RoomNet in cuboid and irregular indoor environments. (Top two rows) Accurate segmentation in standard cuboid settings. (Bottom two rows) Irregular scenarios, including corridors and partial views, which deviate from the reference cuboid assumption.

To refine the detected line segments, we implement a two-stage grouping method based on orientation and spatial proximity. First, the orientation of each line segment is calculated as the angle between the segment and the horizontal axis, normalized to [0, 180) degrees. Line segments with angular differences within a predefined threshold are grouped together, ensuring that lines with similar orientations are clustered accurately. Angular differences are computed by accounting for wrap-around at 180°, maintaining precise grouping.

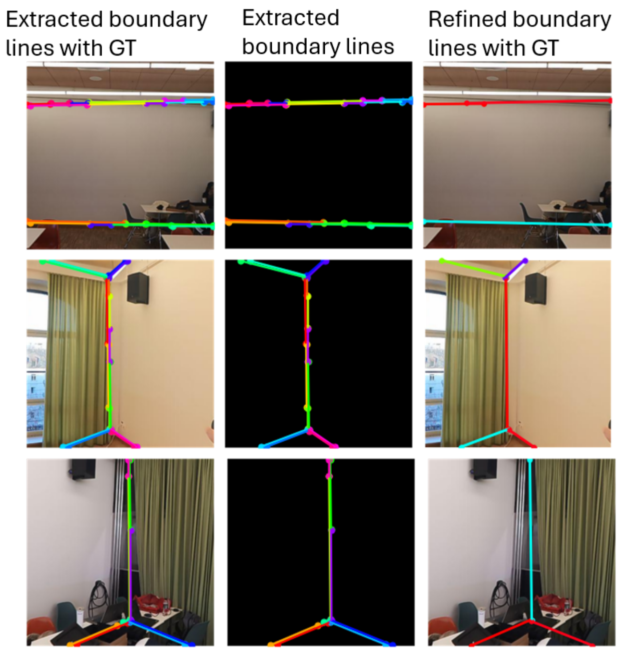

Figure 3. Extracted boundary lines from segmentation output (first and second columns) and consolidated representative lines after refinement (third column).

Next, within each orientation group, line segments are further subdivided into subgroups based on spatial proximity. Proximity is determined by calculating the shortest distance between two line segments, considering both endpoints and projections to handle non-overlapping cases. Line segments within a predefined distance threshold are grouped together.

Representative lines are then computed for each group by aggregating all line endpoints and determining the principal axis using Principal Component Analysis (PCA). The centroid of the aggregated points is calculated, and projections onto the principal axis define the start and end points of the representative line, capturing the group's primary geometric orientation. This consolidation process reduces noise and provides a concise depiction of the layout geometry (see the third column of Figure 3).

3.2. 3D Line Generation

This subsection outlines the process for aligning depth images from a depth camera with color images captured by a co-located color camera, both of which have similar orientations and positions. The method involves transforming depth data into a three-dimensional point cloud, projecting the points onto the color camera's image plane, and associating the corresponding depth values. Finally, the room layout lines with depth values will be refined and generated.

On the Magic Leap 2 platform, aligning the color and depth cameras poses challenges due to differences in their undistortion parameters. However, analysis across multiple frames reveals that the physical separation between the cameras is approximately 2 centimeters, with orientation differences of about 2-5 degrees along each axis. Given a depth range of up to 7 meters and a field of view (FOV) of the depth camera of 89 degrees, these discrepancies are negligible. Consequently, the alignment process assumes that the color and depth cameras are effectively co-located and share identical positions and orientations.

The alignment of depth images from a depth camera with color images from a co-located color camera is achieved through the following mathematical framework:

Camera Intrinsic Parameters Both cameras are characterized by their intrinsic matrices:

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{1}$$

where f_x and f_y are the focal lengths, and (c_x, c_y) denote the principal point coordinates.

Depth to Point Cloud Given a depth map $D \in \mathbb{R}^{H\times W}$, each pixel (i, j) with depth $z_{i,j} = D(i, j)$ is back-projected to three-dimensional coordinates:

$$x_{i,j} = \frac{(j - c_x)\cdot z_{i,j}}{f_x},\ y_{i,j} \quad = \frac{(i - c_y)\cdot z_{i,j}}{f_y},\ z_{i,j} = D(i, j). \tag{2}$$

This transformation generates a point cloud $\mathbf{P} \in \mathbb{R}^{H\times W\times 3}$ representing the scene's 3D structure.

Point Cloud Projection To align with the color image, each 3D point is projected onto the color camera's image plane:

$$u_{i,j} = \frac{x_{i,j}\cdot f_x^{\text{color}}}{z_{i,j}} + c_x^{\text{color}},\ v_{i,j} \quad = \frac{y_{i,j}\cdot f_y^{\text{color}}}{z_{i,j}} + c_y^{\text{color}}, \tag{3}$$

Finally, depth values are mapped onto the color image based on corresponding pixel coordinates. The captured depth images and the aligned depth images are presented in the first and second columns of Figure 4. The second column additionally displays the room layout lines with sampled points and their associated depth values. However, the aligned depth images exhibit quality issues, such as occlusions and noise artifacts. To address these challenges, we implement a refinement method that enhances depth accuracy by fitting robust line models and clustering spatially proximate points.

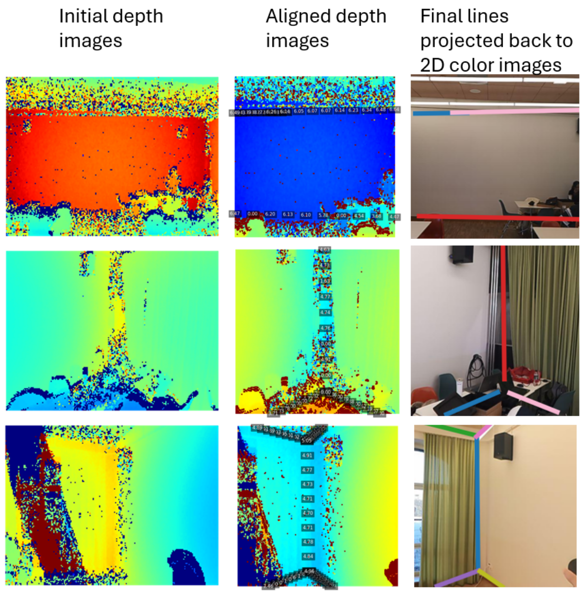

Figure 4. Depth alignment and final representative lines. (First and second columns) Captured and aligned depth images; (Third column) Generated 3D lins projected back to 2D color images for verification.

Specifically, the refinement process begins by selecting the top N points with the highest depth values and applying the RANSAC algorithm to fit a linear model, thereby mitigating the influence of outliers. Subsequently, DBSCAN clustering is performed in the x-y plane to group nearby points, ensuring consistent depth values within neighbor points.

To derive representative lines from the refined 3D points, PCA is employed for each 3D room layout line containing a group of points, summarizing the points into a single, well-defined line segment. This approach reduces noise and enhances the clarity of the reconstructed room geometry. Finally, the generated 3D lines are projected back onto the color images using the cameras' extrinsic and intrinsic parameters to verify the accuracy of the alignment (see the third column of Figure 4).

3.3. Rendering and User Experience

The endpoints of the generated 3D lines are transmitted to the Magic Leap 2 for final rendering. In addition to establishing a robust back-end and seamless communication between the headset and server, a primary objective of our project was to deliver a functional and enjoyable user experience. To achieve this, we evaluated and tested various approaches to ensure the user interface (UI) was intuitive, visually appealing, and user-friendly. Our goal was to create an interface that felt natural and effortless to navigate, enhancing both the usability and overall satisfaction of the AR experience.

3.3.1 UI Design and Layout

For the UI design of our AR application, we wanted to leverage an interaction model that is both familiar and comfortable for users, especially given the novelty of AR technology. A common interaction paradigm in computer applications is the use of "windows," which users typically interact with via mouse, touchpad, or touchscreen interfaces. We decided to maintain this window-based approach in our AR environment, as it is a familiar concept that users can easily adapt to.

3.3.2 Overlay or 3D Windows?

Since the application is designed for AR and to be used with a headset, we were able to go beyond the traditional flat windows seen on computers and smartphones by transforming them into more dynamic and engaging 3D objects. When deciding how to handle the presentation of windows in AR, we considered two primary approaches:

- **Overlay Windows:** Similar to how windows are used on traditional computers or smartphones, where they remain in a fixed position on the screen, even as the user moves their head or interacts with the environment.
- **3D Object Windows:** The second option, where windows are instantiated as objects within the 3D AR space, allowing them to be manipulated and positioned freely in the environment.

After conducting several early tests, we found that overlay windows, which would remain fixed in front of the user's view (essentially always "in their face"), could often be obtrusive and lead to motion sickness. This effect occurs when static UI elements are positioned too close to the user, moving along with head movements and causing discomfort. This issue has been observed in other AR applications as well [1]. As a result, we opted for the second approach: implementing windows as 3D objects in the AR space, which could be placed at a comfortable distance from the user, preventing discomfort while maintaining usability.

3.3.3 Ergonomic Window Design

Designing the window's shape and positioning was not an easy task. We went through multiple iterations to strike the right balance between accessibility, and comfort. Our approach was inspired by real-world usage objects, such as architects' tables and DJ consoles. These are typically positioned below eye level and are comfortably viewed with a slight downward head rotation. We adapted this concept for

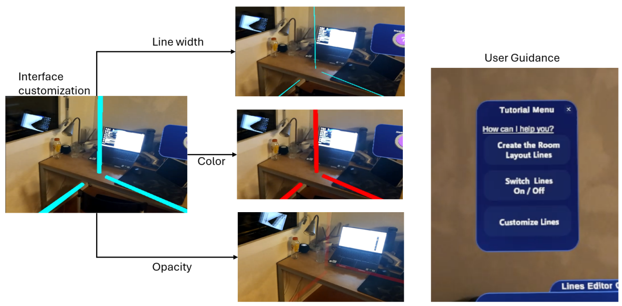

Figure 5. Console layout for the designed Magic Leap 2 app; (Left) Ergonomic window design; (Right) User onboarding and guidance.

our AR interface, placing the windows at a similar angle to ensure the user could interact with them comfortably (see the left image of Figure 5).

3.3.4 Interaction Modes

To make the interaction as intuitive as possible, we chose to use MRTK (Mixed Reality Toolkit)-based tools. This decision was made to ensure that anyone familiar with AR headsets and their control schemes would immediately feel at home in our application, while new users would be introduced to the expertly crafted, engaging features of AR applications. By integrating these tools, we were able to deliver both direct and indirect interaction modes:

- **Close-proximity Interaction:** Users can interact with the window by using their hands or controllers when they are close to the window, providing a tactile, direct mode of interaction.

- **Far-field Interaction:** For users who prefer a more comfortable, indirect interaction, we provided the ability to use controllers' pointers to select and manipulate the window, allowing them to operate from a greater distance.

This dual mode of interaction gives the user the flexibility to choose their preferred way to engage with the application, adapting to different contexts or personal preferences.

3.3.5 Comfort Considerations

In terms of comfort, we followed the official Magicleap Guidelines [4] for placing objects at comfortable distances in AR. Based on these guidelines, we ensured that windows were positioned at a distance greater than 0.40 meters from the user to avoid the discomfort that can arise from objects being too close to the face . By keeping the windows within arm's reach for most adults, we ensured that users could interact with them comfortably without experiencing visual or physical strain.

3.3.6 User Onboarding and Guidance

In addition to the UI design, we were also focused on creating an intuitive way to introduce new users to the application's functionalities. Initially, we considered implementing a traditional control guide, commonly seen with real-world machinery and tools. However, during early testing, we quickly realized that this approach was too cumbersome and felt tedious for new users. It became clear that a more engaging and less intrusive method was needed.

Drawing inspiration from video games, we decided to implement a voice assistant that guides users through a small tutorial. This assistant provides real-time audio feedback and instructions based on the user's actions and needs. Users can easily access the assistant through the help menu, allowing them to request assistance whenever necessary. Unlike traditional guides that overwhelm users with information, the voice assistant only provides feedback on the specific parts of the app that the user is interested in or requires help with. These areas of assistance can be selected by the user through their respective buttons (see the right image of Figure 5).

This method of interactive guidance was very well received during user testing. Given its positive reception, we decided to keep this feature in the final release, allowing new users to become familiar with the application in an enjoyable and low-pressure manner.

4. Experiments

4.1. Data Acquisition

We developed a WebSocket-based server to enable real-time transmission and storage of sensor data essential for room layout estimation in AR environments. The server listens on port 8765 and utilizes a thread pool executor with four worker threads to manage concurrent data streams efficiently. Data is received in chunks, each containing metadata such as *message_id*, *chunk_index*, and *total_chunks*. Upon receiving all chunks for a specific *message_id*, the server reconstructs the complete message, ensuring data integrity.

Each message comprises JSON metadata, a JPEG-encoded color image, and a flattened depth map. Metadata is stored as `metadata_{message_id}.json`, color images are converted to PNG format (`color_{message_id}.png`), and depth maps are reshaped and saved as NumPy arrays (`flattened_{message_id}.npy`). Processed messages are indexed in `index.json` for organized retrieval.

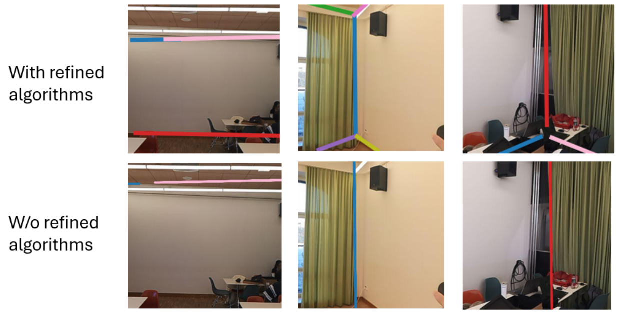

Figure 6. Generated 3D lines reprojected back to 2D images; (The top row) With updated refined algorithms; (The bottom row) Without refined algorithms (with OneFormer instead).

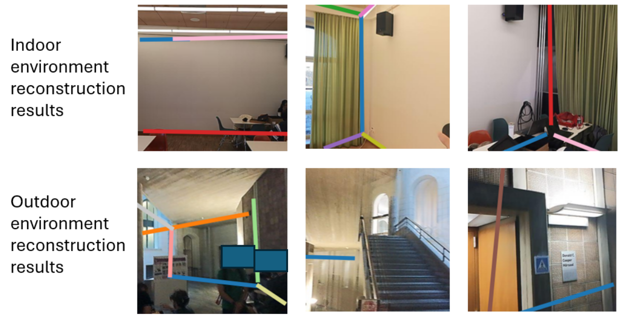

Figure 7. Generated 3D lines reprojected back to 2D images; (The top row) Indoor cases; (The bottom row) Outdoor cases.

The server employs asynchronous operations using the `asyncio` library to handle tasks such as directory monitoring, WebSocket communication, and data processing concurrently. A cleanup routine removes residual `.txt` files before data acquisition begins, maintaining an uncluttered environment. Additionally, a manual termination command ensures graceful shutdown, completing all ongoing tasks and releasing resources appropriately.

This data acquisition framework ensures efficient, reliable, and organized collection of sensor data, providing a robust foundation for subsequent stages of room layout estimation and visualization.

4.2. Results and Comparisons

We conducted a preliminary user study to evaluate both the application's interface and its underlying layout estimation performance. Participants reported that the user interface was intuitive and straightforward to navigate, which aligns with our goal of creating a functional and enjoyable user experience. However, two key concerns emerged: **(1) the accuracy of line generation remained limited, and (2) the overall processing time was relatively high.**

Refinement-Based Improvements. To address these concerns, we opted not to integrate computationally heavy segmentation models such as OneFormer. Instead, we improved our refinement pipeline to reduce the abundance of small line segments and better consolidate extracted boundaries as well as accurate depth acquision. Figure 6 illustrates the outcome of our enhanced refinement strategy, demonstrating clearer room boundaries and a reduction in extraneous lines compared to the initial results.

ST-RoomNet and Depth Range Limitations. Two critical factors affecting layout accuracy are the limitations of ST-RoomNet, as discussed in Subsection 3.1, and the depth camera's maximum range of approximately 7 meters. While this range is sufficient for many indoor settings, it poses challenges in corridor-like environments where walls and objects extend beyond 7 meters. As illustrated in Figure 7, our method accurately reconstructs confined indoor areas but struggles to detect distant boundaries in longer corridors. This limitation contributed to inaccuracies observed during the demo day, particularly when mapping larger or deeper sections of the environment. Overall, our refinement approach significantly reduced noise and accelerated processing without relying on computationally intensive segmentation networks. However, the depth camera's limited range remains a bottleneck in expansive environments, highlighting the need for alternative sensing strategies or cameras with extended depth coverage to achieve robust large-scale AR applications.

5. Conclusion

In this work, we presented an end-to-end pipeline for room layout estimation in augmented reality environments using the Magic Leap 2 platform. By leveraging the device's advanced depth sensing capabilities, we bypassed traditional, computationally intensive depth estimation methods. Our approach integrated lightweight, cuboid-based 2D boundary segmentation through ST-RoomNet, refined 3D line generation, and a user-centric rendering methodology.

The dual-stage line refinement modules effectively reduced noise and clutter, enhancing the clarity and accuracy of the reconstructed room layouts. Our customizable and user-friendly interface contributed to an intuitive AR experience, as confirmed by preliminary user studies. Participants found the interface straightforward to navigate, although they noted limitations in line generation accuracy and processing speed.

To address these concerns, we implemented refinement-based improvements that reduced the abundance of small line segments and consolidated extracted boundaries without relying on heavy segmentation models. While these enhancements improved performance, limitations remained due to ST-RoomNet's reliance on cuboid assumptions and the Magic Leap 2's depth camera range of approximately seven meters. These factors presented challenges in environments with irregular shapes or extended distances, such as long corridors, affecting the system's ability to detect distant boundaries accurately.

Despite these challenges, our findings highlight the potential of combining advanced AR hardware with efficient algorithms to enhance indoor spatial understanding. The synergy between Magic Leap 2's depth sensing and our optimized computational methods not only improves technical performance but also enriches the user experience in AR applications.

Future work will focus on addressing the identified limitations by exploring alternative depth sensing strategies with extended ranges and adopting more flexible room layout models that can handle diverse environments. Additionally, further optimization of the user interface will aim to enhance usability and processing speed, paving the way for more immersive and intelligent indoor mapping applications.

Our research lays the groundwork for advancing room layout estimation in AR, demonstrating the feasibility and benefits of integrating real-time depth data with efficient computational techniques to create accurate and user-friendly spatial representations.

6. Acknowledgement

We extend our deepest gratitude to our supervisor, Remi, for his unwavering encouragement, patience, and insightful guidance throughout this project. His commitment to regularly meeting with us and his approachable nature have been invaluable to our progress and success.

We also wish to thank the Teaching Assistant group and the Instructor team for their dedicated support. Their efforts in organizing the courses, inviting distinguished lecturers, and providing valuable insights have greatly enriched our learning experience. Additionally, their assistance in facilitating effective teamwork has been crucial in the development and completion of this project.

References

[1] E. Chang, H. T. Kim, and B. Yoo. Virtual reality sickness: A review of causes and measurements. *International Journal of Human-Computer Interaction*, 36(17):1658–1682, 2020. 5

[2] Vibhav Hedau, Arpit Gupta, and Martial Hebert. Recovering the spatial layout of cluttered rooms. In *International Conference on Computer Vision (ICCV)*, pages 1849–1856, 2009. 2

[3] Hatem Ibrahem, Ahmed Salem, and Hyun-Soo Kang. St-roomnet: Learning room layout estimation from single image through unsupervised spatial transformations. In *CVPR Workshop (CVPRW)*, pages 3376–3384, 2023. 2

[4] Magic Leap. Comfort and content placement guidelines. https://developer-docs.magicleap.cloud/docs/guides/best-practices/comfort-content-placement/. 6

[5] Chen-Yu Lee, Vijay Badrinarayanan, Tomasz Malisiewicz, and Andrew Rabinovich. Roomnet: End-to-end room layout estimation. *International Conference on Computer Vision (ICCV)*, pages 4865–4874, 2017. 2

[6] Hung Jin Lin, Sheng-Wei Huang, Shang-Hong Lai, and Chen-Kuo Chiang. Indoor scene layout estimation from a single image. In *International Conference on Pattern Recognition (ICPR)*, pages 842–849, 2018. 2

[7] Jiu Xu, Björn Stenger, Tommi Kerola, and Tony Tung. Pano2cad: Room layout from a single panorama image. In *Winter Conference on Applications of Computer Vision (WACV)*, pages 354–362, 2017. 2

[8] Weidong Zhang, Wei Zhang, Kan Liu, and Jason Gu. Learning to predict high-quality edge maps for room layout estimation. *IEEE Transactions on Multimedia*, 19(5):935–948, 2017. 2

[9] Chuhang Zou, Alex Colburn, Qi Shan, and Derek Hoiem. Layoutnet: Reconstructing the 3d room layout from a single rgb image. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2051–2059, 2018. 2