rulAR: Spatial Measurement in Indoor Scenes via Intelligent Labeling and Immersive Visualization

Zhaohan Pan
Independent Researcher
chester052519@gmail.com

Figure 1: rulAR is an augmented reality experiential prototype for indoor scene spatial measurement through visualizing selected intelligent labels (left) or immersive dimension views (right).

Abstract

Most existing augmented reality (AR) measuring tools enforce manipulation patterns that are operation-heavy and demand motor precision, or they impose a requirement for close-range manual references during measuring workflows. As spatial computing enters everyday practice, we present rulAR, an innovative spatial-measurement interactive prototype designed to overcome this limitation. Developed on the Apple Vision Pro, rulAR highlights low-manual to zero-manual task operation through an intelligent, scene-aware spatial interface: it intelligently labels measurement-critical points in the scene for selection or automatically delivers immersive visualized dimensions. A robust backend pipeline for geometric scene reconstruction and rendering is implemented to support the final user interface. rulAR is expected to provide an experimental venue for addressing scenarios where precise device operations are inaccessible and to serve as a concept for investigating innovative interactivity within spatial computing frameworks.

Conference'17, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

ACM Reference Format:
Zhaohan Pan. 2025. rulAR: Spatial Measurement in Indoor Scenes via Intelligent Labeling and Immersive Visualization. In . ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Measurement and dimensioning in spatial scenes are frequent tasks in various practical contexts. Whether in architectural designs or engineering constructions, they always involve measuring objects: furniture, ceiling beams, girder structures, or free-space gaps are all information of interest in planning and design [5]. Over time, humans have invented a variety of tools for accomplishing measurement tasks—rulers, tape measures, calipers, and total stations—each with different operational difficulties and applicable scenarios, yet sharing a broadly similar logic: measuring the distance between two points using a reference.

With the advances in Augmented Reality and 3D computer vision (CV), spatial measurement tasks can now be performed through new forms of human–computer interaction [5]. This capability is supported by several core techniques: spatial dimensions can be derived from segmentation-based geometry extraction, depth–estimation models [11], and Structure-from-Motion pipelines [17]; camera-equipped or LiDAR-enhanced devices run AR sessions that integrate these CV pipelines to reconstruct spatial layouts [6]; interactive interfaces then allow users to specify points of interest and receive measurements through rendered visualizations [2]. Given the limitations in CV accuracy, depth noise, and feature sparsity, these systems are currently most effective in indoor settings, where precision requirements are moderate, and scenes provide sufficient features for stable reconstruction. Consequently, users benefit from reduced manual effort compared with conventional measurement tools.

However, interaction methods for AR-based spatial measurement remain largely constrained to monotonic and operation-intensive workflows: users are required to manually and precisely target points, a process that becomes challenging for individuals with limited fine-motor capabilities [24]. Mobile AR tools further demand substantial device maneuvering and provide only a narrow field of view for presenting dimensional information of an entire scene [14]. Fundamentally, AR measurement systems aim to help users visualize the dimensions of a scene of interest. Thus, interfaces that enable more intelligent and less burdensome mechanisms for selecting and visualizing dimensional information represent a promising direction for advancement. One such direction is to design interfaces that automatically annotate the environment, allowing users to directly select and view the dimensions they need; another is to provide fully immersive, scene-wide annotations that offer comprehensive dimensional information. Such concepts are particularly well supported by emerging head mounted display spatial computing platforms, which combine new input and interaction modalities, immersive scene level visualization, and integrated CV capabilities that together provide an especially fitting environment for implementing these forms of spatial measurement interaction. Moreover, such platforms provide practical advantages for prototyping, deploying interactions, and conducting user studies to validate these ideas.

Therefore, we introduce rulAR, a spatial measurement experiential prototype developed on Apple Vision Pro, highlighting minimal user manual operation by delivering the measurement experience immersively and intelligently within a spatial computing framework. We envision rulAR as a proof of concept that demonstrates new forms of interactivity enabled by spatial computing for AR measurement workflows and serves as an initial step toward rethinking a broader range of situational task practices on emerging AR platforms.

2 Related Works

AR Interfaces as Extended Tools AR has been widely recognized for its ability to support everyday tasks through in-situ visualization, where scene understanding and virtual guidance help reduce both physical effort and cognitive load compared with traditional methods [3, 4, 23]. Indoor architectural design and construction are prominent examples of such tasks and remain major application areas for AR visualization tools, including object placement, spatial planning, and dimension measurement [5, 8]. Prior studies have further shown that AR interfaces can be used to capture indoor texture information for ex situ scene reconstruction [21], and that AR driven interactions can assist in generating CAD models of indoor environments and objects to accelerate creative workflows [16].

Measurement Pattern and References AR interfaces also support the accurate measurement of indoor environments and objects, a capability that has matured across major mobile platforms [2, 7]. While these systems provide core measurement functions, their workflows typically follow a single interaction pattern, despite the fact that measurement activities span multiple patterns [15], a topic frequently explored in novel human–computer interface design. Recent work has expanded classical measurement tools, such as tape measures [10] and calipers [22], into AR based experiences that directly output spatial planning information or modeling reconstructions. Additionally, holographic interfaces have been used to overlay real-time hand drawn geometric data [18]. Gesture or posture based approaches further improve operational convenience; for example, by estimating dimensions from mid air limb posture amplitudes [9] or fingertip distances for grasped objects [19]. Directly showing the dimensions of the selected objects [12] also provides similar convenience in particularly parametric design workflows. However, these approaches still fundamentally depend on detecting reference object features or actions. In contrast, rulAR seeks to explore new measurement patterns that move beyond reliance on such melee references; instead, it intelligently analyzes the entire indoor scene to support a new mode of spatial measurement interaction.

3 Manual Operation and Intelligent Automation

Designing different AR measurement interfaces can be regarded as a balance between enhancing intelligent automation and giving users more operational control. We believe this balancing process can be illustrated through the following comparison scenarios.

- **Manual Measurement with AR Camera** Mobile AR tools typically require users to select measurement points by precisely aiming the device camera and dragging toward new points. Changing the projection plane requires additional interface controls. This interaction format offers the greatest degree of operational control but falls back to relatively complex and precision-demanding actions.
- **Intelligently Labeled Scene Measurement** Indoor measurement tasks primarily target scene objects and enclosed environmental structures, such as wall widths, cabinet heights, and display sizes, whose dimensions are usually obtained by measuring distances between critical points (typically convex or concave corners). If the interface directly provides these critical points, users can quickly obtain the dimensions they need through simple selection. This format represents a balance between intelligent automation and operational control.
- **Fully Immersive Dimension Display** The interface can also directly visualize all dimensions in the scene, eliminating the need for manual interaction. This format removes operational errors and learning requirements but correspondingly reduces the degree of user control.

Within these scenarios, the first approach represents an already established baseline, whereas our interest lies in exploring the latter two. Together, they illustrate how AR measurement can extend beyond precise manual manipulation toward modes that emphasize system intelligence and immersive presentation. In this sense, *rulAR* is positioned as an attempt to interrogate and validate these alternative points along the continuous spectrum between manual operational control and intelligent automation.

4 Experiential Prototype Development

rulAR is developed on Vision Pro (visionOS 2.0), with all designs created through Apple's developer resources. We primarily employed ARKit for essential scene reconstruction, with PlaneAnchor [1] serving as a key API. In addition, RealityKit and UIKit were used alongside ARKit for entity management, rendering, animation design, and constructing utility functions for AR user input. All interactions and functionalities rely entirely on the hardware and software within the Vision Pro ecosystem, making the experiential prototype suitable for future distributed field studies or replication.

5 Scene Information Reconstruction

Reconstructing an accurate and useful 3D scene is the critical backbone of an AR measurement interface. For rulAR, we applied reasonable simplifications and robust algorithms to transform ARKit outputs into a form suitable for the interface.

5.1 Plane Detection by ARKit

PlaneAnchor serves as the core component for obtaining the initial raw scene data by detecting horizontal, vertical, and slanted planes in space. It provides two types of detected outputs, as shown in **Figure 2**: (1) a low fidelity rectangular overlay that approximates plane orientation, size, and position; and (2) a high fidelity mesh overlay that preserves plane orientation while triangular-meshing the main visible region of the plane at accurate in-situ positions. We based our subsequent scene reconstruction on the second type of output. Experimental visualization demonstrates that PlaneAnchor mesh geometry can synthesize the primary plane region even in the presence of occluding objects (for example, the vase in the figure). The overall plane geometry and stable pose it provides are reliable and suitable for delivering meaningful spatial information.

Figure 2: Visualization of two types of plane anchors acquired from Plane Detection Data Provider.

5.2 Reconstruction from Detection Mesh

The drawback of detection mesh geometry lies in its inability to effectively capture the complete geometry of a plane. In particular, corner positions are often rounded into arcs formed by multiple short mesh segments, whereas in our spatial measurement scenario, plane corners are key features for identifying critical points. To support the required usability and robustness of the interface, we applied geometric reconstruction and refinement algorithms in our pipeline.

At this stage, we introduced a simplification by assuming that all detected results originate from independent rectangular planes, and we therefore attempted to reconstruct each polygon into a rectangle most closely aligned with its physical plane. This assumption reflects the observation that many real world indoor planes are composed of rectangular regions. More precise clustering and refinement of detection results into complex planar structures remain a valuable direction for future improvement.

Process Raw Mesh Data The raw mesh data, represented as unsorted 3D vertices and edges, must first undergo preprocessing. The guiding principle is that interior edges are shared by exactly two triangles, whereas boundary edges correspond to only one. For each mesh dataset, we traverse all edges and identify those that appear only once. These unique edges are assembled into a continuous loop, yielding a cleaned polygonal representation of the detected plane in the form of its vertex loop. The vertex loop is then projected into two dimensions using the Newell method [13] to support faster computation in the subsequent reconstruction process.

Rotating Calipers MinArea Search We used the rotating calipers algorithm [20] to search for the minimum-area rectangle on the convex hull vertices, as shown in **Equation 1**.

$$\arg \min_{\theta, u_{\min}, u_{\max}, v_{\min}, v_{\max}} (u_{\max} - u_{\min})(v_{\max} - v_{\min}) \tag{1}$$

where $\{u_{min}, u_{max}, v_{min}, v_{max}\}$ are the extrema of the convex hull vertices projected onto the axes $\{u_\theta, v_\theta\}$. As discussed before, detected mesh geometries usually underestimate rectangular planes, especially at the corners. This method provides a reasonable estimation of the true area and center position (even in cases where the orientation may deviate).

Quantile-Trimming Huber Search Reconstructing the orientation of plane-rectangles is formulated as another optimization problem, where we minimize the Huber loss of the total distance between a quartile-trimmed rectangle and its mesh geometry. The above process undergoes two rounds of search in a coarse orientation set and then in a fine orientation set, as given by the **Equation 2**.

$$\arg \min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \rho_{\delta(\theta)} \left(d_i(\theta) \right) \tag{2}$$

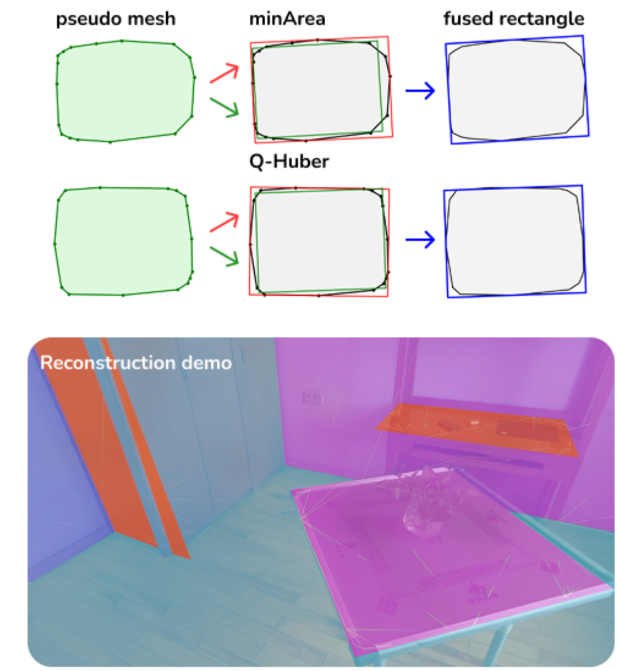

Figure 3: Schematics of plane-rectangle reconstruction algorithm on synthesized data and visualized reconstruction performance on the plane anchor data.

Its corresponding Huber loss function and distance metric are defined below.

$$\rho_\delta(r) = \begin{cases} \frac{1}{2}r^2, & r \le \delta \\ \delta\,(r - \frac{1}{2}\delta), & r > \delta \end{cases}$$

$$d_i(\theta) = \sqrt{\left[\,\max(|u_i - m_u| - \tfrac{w}{2}, 0)\right]^2 + \left[\,\max(|v_i - m_v| - \tfrac{h}{2}, 0)\right]^2}$$

where $u_{min}, u_{max}, v_{min}, v_{max}$ that defines the rectangle is obtained through quantile trims of mesh geometry vertex coordinates $\{p_{i,2D}\}$, and w, h, m_u, m_v are the corresponding width, height, and center coordinates of this rectangle. λ serves as a scaling parameter that adapts the Huber threshold $\delta(\theta)$ to the size of the rectangle, ensuring consistent residual evaluation across scales.

This method suppresses the influence of extreme points on the rectangle's scale and orientation through quartile trimming, while the Huber loss smooths the discontinuous distance metric and outliers, thereby achieving a robust reconstruction of rectangle orientation. Note that the orientation estimated by the Q-Huber search is ambiguous up to a 90 degree rotation, because the loss is invariant to swapping the rectangle's axes and its width–height pair. To resolve this, we use the orientation returned by the MinArea Search enclosing rectangle step to detect the ambiguity and correct it by swapping the width and height when necessary.

The reconstruction pipeline visualization is available in **Figure 3 (b)**. We provide two examples of mesh geometry through synthesized data and show that the two algorithms contribute separately to (1) position & size and (2) orientation attributes. The fused rectangle result of examples and the visualized reconstruction demo shows good quality in achieving our described goals. We also included a completed reconstruction pipeline in **Appendix 1** for reference.

6 Scene Interactivity Design

The described interactivity of rulAR is achieved through visualization and UI design. Reconstructed scene data are applied to rendering branches for different interaction formats.

6.1 Intelligently-Labeled Scene Measurement

Locating Critical Points Using the previously reconstructed plane-rectangles, the spatial positions of critical points in the scene can then be located and visualized for further interaction.

The four corner positions of each plane-rectangle serve as candidates for critical points in the real scene. Because critical points arise from the junctions of adjacent planes, multiple corner candidates may correspond to the same physical point. To address this, we apply a protection radius to eliminate redundancy among candidates. When additional candidates fall within a defined radius of an existing one, their positions are averaged to form a single merged critical point; if no candidates overlap, the point is finalized at its original position. This procedure implicitly leverages information from multiple planes, thereby further mitigating inaccuracies introduced during plane detection and reconstruction.

The finalized critical point positions are then used for visualization. At each position, rulAR renders a sphere entity as a critical point label, as shown in **Figure 4 (a)**. Each entity uses reflective material handled by RealityKit, allowing lighting and shading to adapt to the current environment so that users can perceive spatial positions more intuitively and accurately. To avoid overwhelming the interface, we render only entities within a defined forward viewing range, determined by the camera's look-at vector and a visual cone controlled by specific parameters.

Measure by Selection Along with the visualization of critical points, we added interaction methods to enable the full select and measure functionality.

Since SwiftUI does not currently support direct hover or gaze based interaction with three dimensional entities, we introduced a smaller gaze cone inside the existing visual cone and applied animation to simulate a hover effect. As shown in **Figure 4 (b)**, this gaze cone identifies the first critical point entity it intersects, and the selected entities are given an enlarging animation.

When a critical point is gazed at, the user can perform a spatial tap to create a ruler endpoint entity at its position. The same operation is then used to select a second endpoint. The two endpoints are subsequently connected in rendering and annotated with their corresponding spatial dimensions, completing the visualization, as illustrated in **Figure 4 (c)**. The created endpoints and displayed dimensions remain persistent across viewpoint changes, allowing

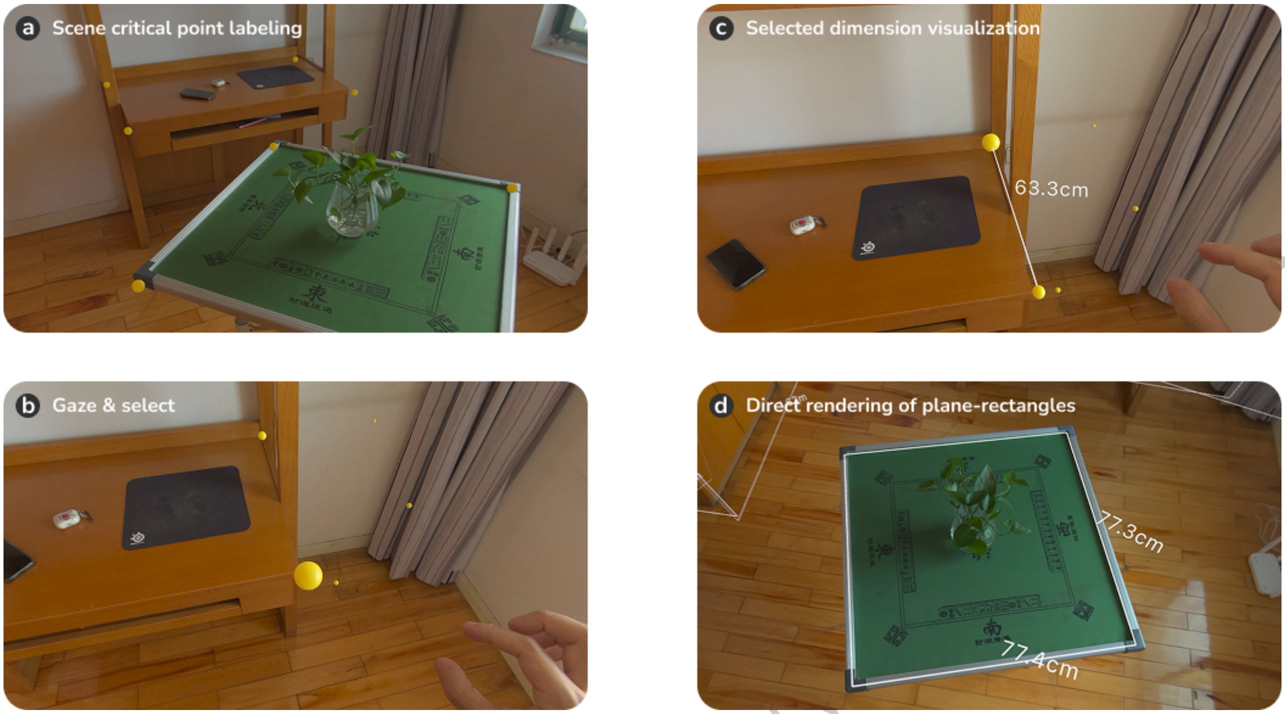

Figure 4: Workflow of conducting a measurement with intelligent labels: (a) viewing the critical points in the current scene, (b) finding the point of interest and gaze & spatial tap it to select, (c) repeat the previous step for another point to complete one measurement. (d) Conducting measurement by directly looking at the object of interest in the immersive format.

measurements to extend beyond the visual cone and to be retained for comparison or for continuous measurement.

6.2 Fully Immersive Dimension Display

A fully hands free measurement experience directly utilizes all plane-rectangle information from the scene, enabling interaction to occur purely through visual communication.

In this branch of the rendering pipeline, we render the boundary vectors of the reconstructed rectangles as line segments and label the length of one selected pair of adjacent edges, as shown in **Figure 4 (d)**. To avoid visual overload, only planes within a defined radius of the device's field of view are rendered. Compared with the other interaction formats, this approach presents a substantially more comprehensive amount of spatial information.

7 Interactive Demo and Discussion

A comprehensive demo of the two designed interaction scenes is shown in **Figure 5**. In **Intelligently Labeled Scene Measurement**, critical points can be used not only to measure plane edges but also to perform seamless measurements on non planar transitions, free-space gaps, diagonals, and essentially any dimension users wish to obtain. In **Fully Immersive Dimension Display**, users can perceive the scene comprehensively by simply walking and looking around their indoor environments.

rulAR provides capabilities expected of AR spatial measurement tools while addressing key limitations of traditional methods. Unlike conventional tools that require physical contact and stable positioning against surfaces, rulAR enables the measurement of hard to reach areas such as high ceilings, distant corners, and mid air gaps through simple gaze and tap interactions. The system also eliminates awkward single person scenarios, such as using a traditional tape measure, where one must hold both ends while simultaneously recording measurements. Furthermore, continuous dimension visualization and fully immersive presentation offer spatial understanding that would otherwise require multiple tedious manual measurements, allowing users to instantly perceive all relevant dimensions within their field of view as they navigate the space.

More importantly, rulAR demonstrates interaction paradigms that extend beyond existing AR measurement workflows. In pursuing our primary objective of exploring novel interactions for established tasks, rulAR introduces a shift from manual point targeting toward intelligent scene understanding. This advancement is particularly valuable across diverse user scenarios: individuals with limited fine motor control can operate through simple gaze selection; professionals whose hands are occupied with parallel workflows can obtain measurements without interruption; and casual users or visitors seeking quick spatial understanding can walk through spaces and glance at automatically generated measurements, bypassing the learning curve of traditional AR interface operations. By combining the natural look and tap interaction paradigm of spatial computing with intelligent automation, rulAR transforms spatial measurement from a focused operational task

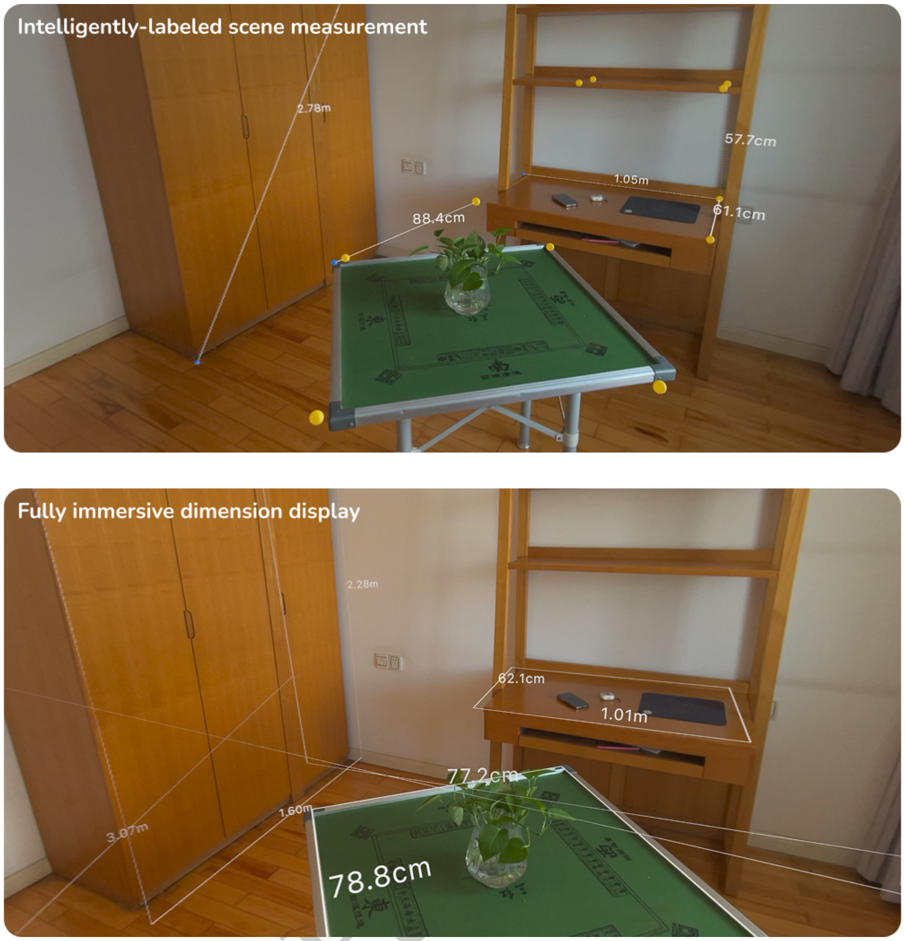

Figure 5: Large field of view clips in a multiple feature-objects scene on the two interfaces introduced by rulAR, Intelligently-Labeled Scene Measurement (top) and Fully Immersive Dimension Display (bottom).

into an ambient and effortless experience that adapts to users' varying needs and capabilities.

8 Limitation and Future Works

rulAR demonstrates a practical path beyond existing AR measurement workflow interactivities. Leveraging the spatial computing framework, it demonstrates that an intelligent, scene aware interface can either surface measurement critical points for quick selection or render comprehensive dimensions directly in situ, providing insight into new directions for AR task human–computer interfaces. Together, these directions point toward spatial computing measurement experiences that more flexibly balance system initiative with human operational control while broadening who—and which contexts—such tools can serve.

A primary limitation of rulAR lies in the accuracy and semantic meaningfulness of scene reconstruction: because the experience is built atop ARKit's data provider, fidelity is ultimately constrained by upstream algorithmic capabilities. To more directly advance rulAR's objectives, future work can better leverage heterogeneous scene signals. For example, by fusing three dimensional plane anchors and meshes with two dimensional image buffer computer vision, intelligent labeling and visualization can be made more reliable and context aware. Building on iterative reconstruction, interactivity can also be expanded by envisioning richer behaviors for scene elements and enabling multiple task pathways that reuse the same underlying scene information.

References

[1] Inc. Apple. 2025. PlaneAnchor.Geometry `struct` — ARKit Documentation. https://developer.apple.com/documentation/arkit/planeanchor/geometry-swift.struct.

[2] Inc. Apple. 2025. Use the Measure app on your iPhone, iPad, or iPod touch. https://support.apple.com/en-us/102468.

[3] Jonas Blattgerste, Benjamin Strenge, Patrick Renner, Thies Pfeiffer, and Kai Essig. 2017. Comparing Conventional and Augmented Reality Instructions for Manual Assembly Tasks. In *Proceedings of the 10th International Conference on PErvasive Technologies Related to Assistive Environments* (Island of Rhodes, Greece) *(PETRA '17)*. Association for Computing Machinery, New York, NY, USA, 75–82. https://doi.org/10.1145/3056540.3056547

[4] Jacky Cao, Kit-Yung Lam, Lik-Hang Lee, Xiaoli Liu, Pan Hui, and Xiang Su. 2023. Mobile Augmented Reality: User Interfaces, Frameworks, and Intelligence. *ACM Comput. Surv.* 55, 9, Article 189 (Jan. 2023), 36 pages. https://doi.org/10.1145/3557999

[5] Juan Manuel Davila Delgado, Lukumon Oyedele, Peter Demian, and Thomas Beach. 2020. A research agenda for augmented and virtual reality in architecture, engineering and construction. *Advanced Engineering Informatics* 45 (2020), 101122. https://doi.org/10.1016/j.aei.2020.101122

[6] Ruofei Du, Eric Turner, Maksym Dzitsiuk, Luca Prasso, Ivo Duarte, Jason Dourgarian, Joao Afonso, Jose Pascoal, Josh Gladstone, Nuno Cruces, Shahram Izadi, Adarsh Kowdle, Konstantine Tsotsos, and David Kim. 2020. DepthLab: Real-time 3D Interaction with Depth Maps for Mobile Augmented Reality. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '20)*. Association for Computing Machinery, New York, NY, USA, 829–843. https://doi.org/10.1145/3379337.3415881

[7] Tasmanic Editions. 2025. CamToPlan - AR tape measure [App]. https://apps.apple.com/us/app/camtoplan-ar-tape-measure/id1292176208.

[8] Aso Hajirasouli and Saeed Banihashemi. 2022. Augmented reality in architecture and construction education: state of the field and opportunities. *International Journal of Educational Technology in Higher Education* 19, 1 (2022), 39.

[9] Bokyung Lee, Minjoo Cho, Joonhee Min, and Daniel Saakes. 2016. Posing and Acting as Input for Personalizing Furniture. In *Proceedings of the 9th Nordic Conference on Human-Computer Interaction* (Gothenburg, Sweden) *(NordiCHI '16)*. Association for Computing Machinery, New York, NY, USA, Article 44, 10 pages. https://doi.org/10.1145/2971485.2971487

[10] Jay Lee, Victor Su, Sandia Ren, and Hiroshi Ishii. 2000. HandSCAPE: a vectorizing tape measure for on-site measuring applications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (The Hague, The Netherlands) *(CHI '00)*. Association for Computing Machinery, New York, NY, USA, 137–144. https://doi.org/10.1145/332040.332417

[11] Zhengqi Li and Noah Snavely. 2018. MegaDepth: Learning Single-View Depth Prediction From Internet Photos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[12] Chen Liang, Anhong Guo, and Jeeeun Kim. 2022. CustomizAR: Facilitating Interactive Exploration and Measurement of Adaptive 3D Designs. In *Proceedings of the 2022 ACM Designing Interactive Systems Conference* (Virtual Event, Australia) *(DIS '22)*. Association for Computing Machinery, New York, NY, USA, 898–912. https://doi.org/10.1145/3532106.3533561

[13] M. E. Newell, R. G. Newell, and T. L. Sancha. 1972. A Solution to the Hidden Surface Problem. https://ohiostate.pressbooks.pub/app/uploads/sites/45/2017/09/newell-newell-sancha.pdf Reprinted PDF available.

[14] Zhaohan Pan, Gaolin Ge, Haoran Lu, Zhihao Meng, Hongrui Wu, Qifeng Yang, and Jingyang Liang. 2025. *karP: an experiential prototype for Kinesthetic Augmented Reality on mobile — Playful, Portable, Producible.* Association for Computing Machinery, New York, NY, USA, 246–250. https://doi.org/10.1145/3715668.3735607

[15] Raf Ramakers, Danny Leen, Jeeeun Kim, Kris Luyten, Steven Houben, and Tom Veuskens. 2023. Measurement Patterns: User-Oriented Strategies for Dealing with Measurements and Dimensions in Making Processes. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 214, 17 pages. https://doi.org/10.1145/3544548.3581157

[16] Aditya Sankar and Steve M. Seitz. 2017. Interactive Room Capture on 3D-Aware Mobile Devices. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology* (Québec City, QC, Canada) *(UIST '17)*. Association for Computing Machinery, New York, NY, USA, 415–426. https://doi.org/10.1145/3126594.3126629

[17] Johannes L. Schonberger and Jan-Michael Frahm. 2016. Structure-From-Motion Revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[18] Anirudh Sharma, Lirong Liu, and Pattie Maes. 2013. Glassified: an augmented ruler based on a transparent display for real-time interactions with paper. In *Adjunct Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology* (St. Andrews, Scotland, United Kingdom) *(UIST '13 Adjunct)*. Association for Computing Machinery, New York, NY, USA, 21–22. https://doi.org/10.1145/2508468.2514937

[19] Evgeny Stemasov, Simon Demharter, Max Rädler, Jan Gugenheimer, and Enrico Rukzio. 2024. pARam: Leveraging Parametric Design in Extended Reality to Support the Personalization of Artifacts for Personal Fabrication. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 337, 22 pages. https://doi.org/10.1145/3613904.3642083

[20] Godfried T. Toussaint. 1983. Solving Geometric Problems with the Rotating Calipers. In *Proc. IEEE MELECON*. Athens, Greece, 1–8. Classic paper introducing the rotating calipers technique.

[21] Zeyu Wang, Cuong Nguyen, Paul Asente, and Julie Dorsey. 2021. DistanciAR: Authoring Site-Specific Augmented Reality Experiences for Remote Environments. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 411, 12 pages. https://doi.org/10.1145/3411764.3445552

[22] Christian Weichel, Jason Alexander, Abhijit Karnik, and Hans Gellersen. 2015. SPATA: Spatio-Tangible Tools for Fabrication-Aware Design. In *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction* (Stanford, California, USA) *(TEI '15)*. Association for Computing Machinery, New York, NY, USA, 189–196. https://doi.org/10.1145/2677199.2680576

[23] Zhen Yang, Jinlei Shi, Wenjun Jiang, Yuexin Sui, Yimin Wu, Shu Ma, Chunyan Kang, and Hongting Li. 2019. Influences of Augmented Reality Assistance on Performance and Cognitive Loads in Different Stages of Assembly Task. *Frontiers in Psychology* Volume 10 - 2019 (2019). https://doi.org/10.3389/fpsyg.2019.01703

[24] Matea Žilak, Željka Car, and Ivana Čuljak. 2022. A Systematic Literature Review of Handheld Augmented Reality Solutions for People with Disabilities. *Sensors* 22, 20 (2022). https://doi.org/10.3390/s22207719

A SceneReconPipeline

Algorithm 1 SceneReconPipeline

procedure MeshGeoRecon($\{p_{\mathrm{i,3D}}\}$, {hyperParam})
  (Stp. 1) 3D → 2D
  $\{p_{i,\mathrm{2D}}\}$, pass ← ProjectByNewell($\{p_{i,\mathrm{3D}}\}$)
  if not pass **then**
    $\{p_{i,\mathrm{2D}}\}$ ← ProjectByPCA($\{p_{i,\mathrm{3D}}\}$)
  end if

  (Stp. 2) MinArea rectangle search
  $H \leftarrow$ ConvexHull($\{p_{i,\mathrm{2D}}\}$)
  $\Theta \leftarrow$ {edge directions of H}
  $(\mathrm{A}^*, \{u_i, v_i\}^*, \theta_{\mathrm{temp}}) \leftarrow (+\infty,\ \mathrm{nil}, \mathrm{nil})$
  for $\theta \in \Theta$ **do**
    $(u_{min}, u_{max}), (v_{min}, v_{max}) \leftarrow$ ExtremOnAxes(H, θ)
    area ← $(u_{max} - u_{min}) \cdot (v_{max} - v_{min})$
    if $\mathrm{A} < \mathrm{A}^*$ **then**
      $(\mathrm{A}^*, \{u_i, v_i\}^*, \theta_{\mathrm{temp}}) \leftarrow (\mathrm{A}, \{u_{min,max}, v_{min,max}\}, \theta)$
    end if
  end for

  (Stp. 3) Q-Huber rectangle search
  $\theta_0 \leftarrow$ InitTwoTheta($\{p_{\mathrm{i,2D}}\}$)
  $\Theta_{\mathrm{coarse}} \leftarrow$ Linspace($\theta_0 - \frac{\Delta}{2},\ \theta_0 + \frac{\Delta}{2},\ n$)
  $(J^*, \theta^*) \leftarrow (+\infty,\ \theta_0)$
  for $\theta \in \Theta_{\mathrm{coarse}}$ **do**
    $J \leftarrow$ EvaluateHuberCost($\{p_{\mathrm{i,2D}}\}$, θ; q, λ)
    if $J < J^*$ **then**
      $(J^*, \theta^*) \leftarrow (J, \theta)$
    end if
  end for

  $\Theta_{\mathrm{fine}} \leftarrow \{\theta^* + \varphi \mid \varphi \in [-\delta^\circ,\ \delta^\circ],\ \mathrm{step} = \epsilon^\circ\}$
  for $\theta \in \Theta_{\mathrm{fine}}$ **do**
    $J \leftarrow$ EvaluateHuberCost($\{p_{\mathrm{i,2D}}\}$, θ; q, λ)
    if $J < J^*$ **then**
      $(J^*, \theta^*) \leftarrow (J, \theta)$
    end if
  end for

  (Stp. 4) Angle correction & rectangle fusion
  $\Delta\theta \leftarrow$ ToHalfPi($\theta^* - \theta_{\mathrm{temp}}$)
  if $|\Delta\theta| > \frac{\pi}{4}$ **then**
    width, height ← height, width `// 90° swap`
  end if
  (center, width, height) ← $\{u_i, v_i\}^*$
  $\mathrm{rect}_{\mathrm{2D}}$ ← ConstructRect(center, width, height, θ^*)

  (Stp. 5) 2D → 3D
  $\mathrm{rect}_{\mathrm{3D}}$ ← BackProjectTo3D($\mathrm{rect}_{\mathrm{2D}}$)
  return $\mathrm{rect}_{\mathrm{3D}}$
end procedure