PreviewAR: Sketching Spaces for Scale-Aware Product Discovery in AR

Davin Win Kyi
Paul G. Allen School of Computer Science and Engineering
University of Washington
Seattle, Washington, USA
davin123@cs.washington.edu

Jason Kim
Paul G. Allen School of Computer Science & Engineering
University of Washington
Seattle, Washington, USA
kimjason@cs.washington.edu

Caleb Hu
Paul G. Allen School of Computer Science & Engineering
University of Washington
Seattle, Washington, USA
chu552@cs.washington.edu

Jewoo Park
Paul G. Allen School of Computer Science & Engineering
University of Washington
Seattle, Washington, USA
jewoop@cs.washington.edu

Jaewook Lee
Paul G. Allen School of Computer Science & Engineering
University of Washington
Seattle, Washington, USA
jaewook4@cs.washington.edu

Jon E. Froehlich
Paul G. Allen School of Computer Science & Engineering
University of Washington
Seattle, Washington, USA
jonf@cs.uw.edu

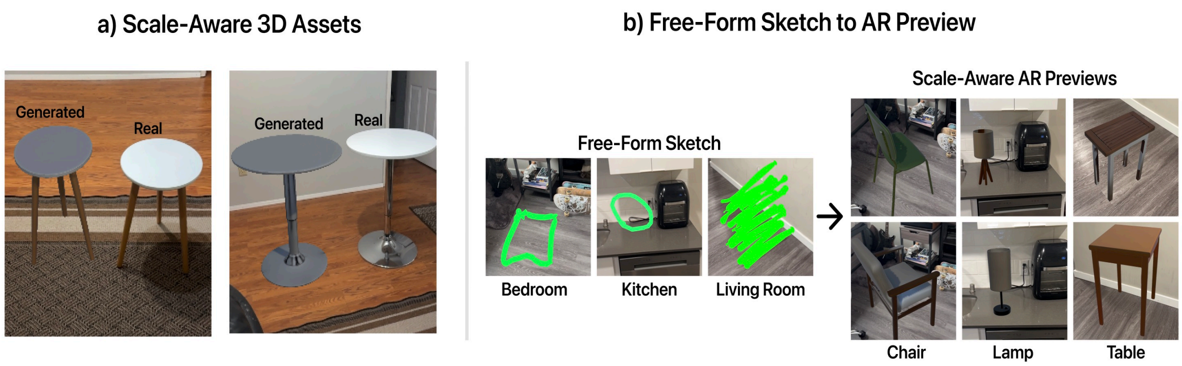

Figure 1: PreviewAR generates scale-aware 3D assets that preserve real-world dimensions for in-situ AR preview (a), and uses free-form sketches as spatial queries to discover and preview compatible products that fit user specified spaces (b)

Abstract

Current AR product-preview tools often lack two capabilities: previewing products without existing 3D assets and supporting space-first product discovery, where users can find items that fit a given space without testing products one by one. Recent 3D asset-generation systems address the first limitation by allowing users to create previews for products with missing 3D assets, but they often fail to preserve real-world scale. To address these limitations, we introduce PreviewAR, an AR application that supports space-first product discovery where users can sketch spatial preferences in their environment, generate scale-aware 3D assets of compatible products, and preview them in their environment.

UIST Adjunct '26, Detroit, MI, USA

ACM ISBN 979-8-4007-2855-6/26/11
https://doi.org/10.1145/3830397.3841872

Keywords

Augmented Reality, Sketch-Based Spatial Interaction, Spatially Aware Product Retrieval, Scale-Aware 3D Asset Generation

ACM Reference Format:
Davin Win Kyi, Jason Kim, Caleb Hu, Jewoo Park, Jaewook Lee, and Jon E. Froehlich. 2026. PreviewAR: Sketching Spaces for Scale-Aware Product Discovery in AR. In *The 39th Annual ACM Symposium on User Interface Software and Technology (UIST Adjunct '26), November 02–05, 2026, Detroit, MI, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3830397.3841872

1 Introduction

AR product preview tools have become increasingly common in online shopping, allowing users to place virtual furniture and other products into their physical environments [2, 3, 13]. Applications such as IKEA Place, Amazon's View in Your Room, and Apple's AR Quick Look demonstrate the value of in-situ previewing: users can inspect how a product might look at real-world scale and evaluate whether it fits their home [2, 3, 13]. However, many products lack readily available AR-ready 3D assets. In addition, these tools also remain largely product-first, requiring users to inspect products one by one before determining whether any fit a desired space.

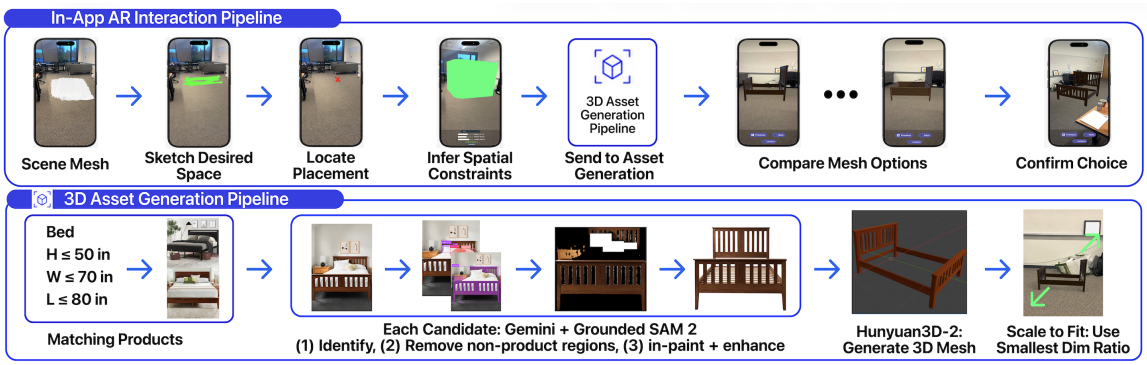

Figure 2: PreviewAR's interaction pipeline turns free-form sketches into scale-aware 3D previews (top), while its asset-generation pipeline produces scale-aware 3D assets (bottom).

To address the limitation of 3D-ready assets, prior work has explored generating 3D assets for AR from images, language, scene graphs, object selections, and product metadata [1, 4, 8, 10]. These systems expand what can be previewed, however, they do not reliably preserve real-world scale. Similarly, sketch-based retrieval systems show that sketches can support asset searches, but treat sketches as shape queries rather than spatial constraints [5, 18].

We present PreviewAR, a space-first AR product exploration system that lets users express spatial intent and preview scale-aware 3D assets for compatible products (Figure 1). Users sketch a desired space in AR, PreviewAR then interprets the sketch as a spatial query, estimates real-world dimensions, combines this constraint with a user's semantic preferences, webscrapes compatible products, generates scale-aware 3D assets, and allows for previews of candidates in situ. PreviewAR has two contributions: (1) scale-aware 3D asset generation for AR product previews and (2) a novel sketch-based AR interaction technique for expressing spatial intent.

2 System Implementation

PreviewAR consists of two connected pipelines (Figure 2): (1) an AR interaction pipeline that enables users to express spatial constraints, and (2) a 3D asset-generation pipeline that receives these constraints, retrieves relevant product metadata, selects and preprocesses the best product image for mesh generation and uniformly scales each candidate to preserve its relative proportions while closely matching the product's real-world dimensions. The resulting assets are streamed to the AR interaction pipeline as candidates.

PreviewAR Interaction Pipeline. PreviewAR is built for iOS using Unity 6000.0.58f2 and Niantic Spatial SDK (NSDK), a Unity plugin for spatial mapping and rendering [11, 17]. NSDK constructs a scene mesh, enabling generated assets to interact with and be placed relative to surrounding surfaces. Users can then interact with the environment by providing a free-form sketch that represents their spatial preferences using Unity's Line Renderer [16].

Using the free-form sketch drawn by the user, PreviewAR samples 3D points from the sketch and uses raycasting to ground those points against detected surfaces to determine the desired center and normal of the anchor for placement. The free-form sketch is also used to estimate the bounding cuboid that best represents the preferred bounding volume that assets should be within. Because free-form sketches can be ambiguous, PreviewAR provides Gemini 3.5 Flash with a screenshot of the sketch in the user's environment to estimate the user's intended spatial preference [6]. Users can later adjust the dimensions of the bounding cuboid if necessary.

Once the desired placement center, placement normal, and bounding cuboid are found, the system sends the user-specified object, which is selected through a dropdown menu, and the spatial preferences of the user, represented by the bounding cuboid, to generate and retrieve valid candidates from the 3D asset generation pipeline. These valid candidates are then displayed to the user, and the user can then place preferred candidates in their environment.

PreviewAR 3D Asset Generation Pipeline. PreviewAR uses Qwen3-4B with the user-specified object, representing semantic preference, and bounding-cuboid dimensions to filter webscraped product pages [14]. Given product dimensions (w_p, d_p, h_p) and a user's preferred dimensions (w_u, d_u, h_u), products are filtered by:

$$\mathcal{P}_{\text{fit}} = \left\{ p \in \mathcal{P} \mid (w_p, d_p, h_p) \leq (w_u, d_u, h_u) \right\},$$

where $\mathcal{P}$ is the retrieved product set and $\mathcal{P}_{\text{fit}}$ its fitting subset.

For each valid candidate, PreviewAR uses GPT-5 to select the best image with minimal occlusion, centering, and, if possible, full product visibility [12]. Once the best image is retrieved, the pipeline then identifies the target object's labels and other object labels using Gemini 3.5 Flash and then uses Grounded SAM 2 to determine the segmentation of the target object, which is kept, and other segmentations to remove [6, 9]. Because removing non-target regions may remove parts of the target, Gemini 3.1 Flash in-paints the missing areas to produce a 3D-generation-ready image [7].

In addition, the 3D-generation-ready image is tuned to provide an image with minimal occlusion, centering, and, if possible, full product visibility, and is then sent to Hunyuan3D-2 on a separate GPU-equipped desktop. These clean inputs support more consistent mesh generation and help preserve the product's relative proportions [19]. The generated mesh is then uploaded to Backblaze and later streamed to an iOS device for previewing.

Because Hunyuan generates models with arbitrary units, PreviewAR rescales each mesh using scraped product dimensions in Unity. Given mesh dimensions (w_m, d_m, h_m) and product dimensions (w_p, d_p, h_p), PreviewAR computes a uniform scale factor that is used to scale the object by computing:

$$s = \min\left(\frac{w_p}{w_m}, \frac{d_p}{d_m}, \frac{h_p}{h_m}\right).$$

The mesh is scaled by a factor of s to preserve relative proportions while matching the product's real-world scale as closely as possible. Across 150 generations spanning 10 product categories, the pipeline required an average of 66.8 seconds per asset and achieved an average dimensional error of 4.88% relative to the product's real-world dimensions using the relative root mean squared error (RRMSE) computed over width, depth, and height.

3 Future Work

Although vision-language models can help interpret spatial intent from sketches, prior work has shown limitations in their spatial reasoning [15]. In PreviewAR, these limitations may affect dimension estimates derived from free-form sketches with ambiguous intent. Future versions could incorporate additional scene context, explicit user input, or lightweight editing controls.

PreviewAR currently focuses on furniture. Future work should evaluate a broader set of objects to uncover limitations that current furniture-focused evaluation may have overlooked. Future work could compare free-form sketching with other methods such as lasso selection or preset spatial templates to assess user preference. Generation latency also remains a practical limitation, with generation runtime averaging 66.8 seconds per asset. Future systems could implement caching, batch model requests, or parallelize pipeline components to provide more reasonable runtimes.

In summary, PreviewAR enables space-first product discovery through scale-aware 3D asset generation and free-form sketching to express spatial intent for in-situ AR product previews. Together, these components shift AR shopping from testing whether a preselected object fits in a space to helping users discover products that fit the constraints of a preferred space.

References

[1] Yashwith D. Alva, Namrata Shrikant Janawade, Avani Jain, Nimish Bhasu, and Sarasvathi V. 2024. Augmented Reality Visualization of Menu Items using Image Reconstruction. In *Proceedings of the 2024 8th International Conference on Virtual and Augmented Reality Simulations*. Association for Computing Machinery, 35–42. doi:10.1145/3657547.3657554

[2] Amazon. undated. See How Products Look in Your Room Using View in Your Room. Accessed June 1, 2026. https://www.amazon.com/visual-search/help/augmented_reality

[3] Apple. undated. Quick Look. Accessed June 1, 2026. https://developer.apple.com/augmented-reality/quick-look/

[4] Majid Behravan, Maryam Haghani, and Denis Gračanin. 2025. Transcending Dimensions using Generative AI: Real-Time 3D Model Generation in Augmented Reality. *arXiv preprint arXiv:2504.21033* (2025). doi:10.48550/arXiv.2504.21033

[5] Mathias Eitz, Ronald Richter, Tamy Boubekeur, Kristian Hildebrand, and Marc Alexa. 2012. Sketch-Based Shape Retrieval. *ACM Transactions on Graphics* 31, 4, Article 31 (2012), 10 pages. doi:10.1145/2185520.2185527

[6] Google AI for Developers. 2026. Gemini 3.5 Flash. Accessed June 2, 2026. https://ai.google.dev/gemini-api/docs/models/gemini-3.5-flash

[7] Google Cloud Documentation. 2026. Gemini 3.1 Flash Image (Nano Banana 2). Accessed June 2, 2026. https://docs.cloud.google.com/gemini-enterprise-agent-platform/models/gemini/3-1-flash-image

[8] Erzhen Hu, Mingyi Li, Jungtaek Hong, Xun Qian, Alex Olwal, David Kim, Seongkook Heo, and Ruofei Du. 2024. Thing2Reality: Transforming 2D Content into Conditioned Multiviews and 3D Gaussian Objects for XR Communication. *arXiv preprint arXiv:2410.07119* (2024). doi:10.48550/arXiv.2410.07119

[9] IDEA Research. 2024. Grounded SAM 2: Ground and Track Anything in Videos. Accessed June 2, 2026. https://github.com/IDEA-Research/Grounded-SAM-2

[10] Jaewook Lee, Filippo Aleotti, Diego Mazala, Guillermo Garcia-Hernando, Sara Vicente, Oliver James Johnston, Isabel Kraus-Liang, Jakub Powierza, Donghoon Shin, Jon E. Froehlich, Gabriel Brostow, and Jessica Van Brummelen. 2025. ImaginateAR: AI-Assisted In-Situ Authoring in Augmented Reality. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology*. Association for Computing Machinery, Article 52, 21 pages. doi:10.1145/3746059.3747635

[11] Niantic Spatial. undated. Setting Up the Niantic SDK for Unity. Accessed June 2, 2026. https://nianticspatial.com/docs/nsdk/3.17.0/setup/

[12] OpenAI. 2025. GPT-5 System Card. Accessed June 2, 2026. https://cdn.openai.com/gpt-5-system-card.pdf

[13] Selcen Ozturkcan. 2021. Service Innovation: Using Augmented Reality in the IKEA Place App. *Journal of Information Technology Teaching Cases* 11, 1 (2021), 8–13. doi:10.1177/2043886920947110

[14] Qwen Team. 2025. Qwen3-4B. Accessed June 2, 2026. https://huggingface.co/Qwen/Qwen3-4B

[15] Ilias Stogiannidis, Steven McDonagh, and Sotirios A. Tsaftaris. 2025. Mind the Gap: Benchmarking Spatial Reasoning in Vision-Language Models. *arXiv preprint arXiv:2503.19707* (2025). doi:10.48550/arXiv.2503.19707

[16] Unity Technologies. undated. Line Renderer Component Reference. Accessed June 3, 2026. https://docs.unity3d.com/Manual/class-LineRenderer.html

[17] Unity Technologies. undated. Unity 6.0 User Manual. Accessed June 2, 2026. https://docs.unity3d.com/6000.0/Documentation/Manual/UnityManual.html

[18] Fang Wang, Le Kang, and Yi Li. 2015. Sketch-Based 3D Shape Retrieval Using Convolutional Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1875–1883. doi:10.1109/CVPR.2015.7298797

[19] Zibo Zhao et al. 2025. Hunyuan3D 2.0: Scaling Diffusion Models for High Resolution Textured 3D Assets Generation. *arXiv preprint arXiv:2501.12202* (2025). doi:10.48550/arXiv.2501.12202