# Snap-3D:
# Generative Object Creation with Snap Spectacles

Brian Lee
Stanford University
bjlee25@stanford.edu

Steven Le
Stanford University
stevenle@stanford.edu

Vencent Vang
Stanford University
vencentv@stanford.edu

## Abstract

*We present Snap3D, a real-time AR pipeline for object capture and visualization using Snap Spectacles and generative AI. Here, users define spatial regions via hand gestures, which are captured by front-facing cameras and processed into 3D meshes through open sourced generative AI model. These meshes are rendered in-situ via Snap's Lens Studio, enabling intuitive, spatially-grounded content creation. What Snap3D aims to do is bridge wearable AR technology and generative AI modeling together. Offering an intuitive, lightweight, gesture-based interface for 3D generation. Initial results show sub-second feedback for image encoding and under 20 seconds for mesh rendering, supporting potential applications in prototyping, education, and immersive design.*

## 1. Introduction

Most augmented reality (AR) platforms today prioritize consumption and interaction over real-time creation. While AR has rapidly matured in visualization and multiplayer interaction, real-world gesture-based input remains relatively underutilized outside of gaming or VR environments. However, with the rise of wearable AR devices such as Snap Spectacles and advances in hand tracking, we're seeing a shift toward more natural, embodied forms of spatial input.

In this project, we introduce Snap3D, a gesture-driven pipeline that enables users to generate 3D content in real-time by interacting with their physical environment. Using Snap Spectacles as the input device, users define a spatial region using a pinch gesture. The Spectacles captures the region as a image, which is then processed by a backend server connected to a open source generative AI model that will output a 3D visualization of the corresponding image. The resulting 3D mesh is rendered back into the user's environment, completing the interactive loop.

## 2. AR Wearables - Snap Spectacles

For this project, our team was able to get our hands on the **Snap Spectacles** and gain access to Snap's Len Studio developer program. The Spectacles are lightweight, head-worn augmented reality (AR) glasses developed by Snap Inc. Their capabilities include:

- **Transparent stereoscopic AR displays** using waveguide optics.
- **Dual front-facing stereo cameras** for depth-aware image capture.
- **6-DoF tracking** via integrated IMU (accelerometer, gyroscope, magnetometer).
- **Touch-sensitive input** and onboard audio feedback.
- **Stereo rendering support**, including depth of field and optional anaglyph-style visual effects.

In our workflow, Spectacles are responsible for capturing the target region between the user's hands, stabilizing bounding boxes, and contextualizing 3D mesh placement.

## 3. Rendering Engine - Lens Studio

Lens Studio [4] is Snap Inc.'s AR development platform for creating interactive experiences on Snapchat and Snap Spectacles. It uses a scene-graph architecture like Unity and Unreal Engine–supporting prefabs, scripting, and sensor-based tracking.

We use Lens Studio to create, design, handle the front-end interaction, enabling users to scan regions, trigger an image capture, and later render returned 3D meshes.

## 4. Generative 3D Models

Our pipeline integrates open-source generative AI models for fast, lightweight 2D-to-3D object creation. With the surge of open-source 3D generation tools due to advances

Figure 1. User defining a spatial region with gestures which captures and converts the capture into an image.

in diffusion models, implicit representations, and pretrained vision-language encoders, it has made it increasingly feasible to build high-fidelity, low-latency pipelines without relying on proprietary APIs or closed systems.

By leveraging community-developed frameworks like **TRELLIS** [1], we gain both transparency and adaptability, enabling real-time experimentation in spatial computing environments like Snap Spectacles. We also explored two more advanced generative models for future integration, which could have offer higher realism or scene-level synthesis or at the least more options.

### 4.1. TRELLIS (Implemented)

TRELLIS is a generative model designed for efficient 3D object synthesis from minimal 2D input, such as sketches or cropped images. A brief high-level review on it is that it begins by encoding the input 2D image using a convolutional encoder to extract spatial features. These features are then aligned with a learned 3D latent space using a cross-attention mechanism, enabling the model to infer geometric structure from visual cues. This fused representation defines an implicit Signed Distance Function (SDF), which predicts the signed distance from any 3D point to the object's surface.

We selected TRELLIS due to its relatively fast inference time, compact input requirements, and compatibility with standard mesh formats (e.g., GLB), making it ideal for our asynchronous backend pipeline. It delivers structurally sound and visually plausible outputs even with limited training data, which is crucial in scenarios when users define custom object regions (image captures) on-the-fly via AR input.


Figure 2. Image capture is sent by API calls to Trellis which generates the 3D object

### 4.2. Future Models

**Huangyan3D-2** [2] combines CLIP-guided latent diffusion with a NeRF-based volumetric lifting approach, achieving photorealistic meshes at the cost of higher latency and compute.

**Hi3DGen** [3] supports the generation of more multiple objects or otherwise the entire generation of a 3D environment. Offering promise for complex spatial outputs in future iterations.

## 5. The Pipeline process

The Snap3D pipeline is composed of four primary stages: input capture, image encoding, AI-based 3D generation, and augmented visualization. Each step is distributed

| Model | Output Type | Capabilities |
|---|---|---|
| Trellis | Editable Mesh | Implicit SDF, fast mesh via Marching Cubes |
| Huangyan3D | Photorealistic Mesh | NeRF volume lifting, CLIP-guided latent diffusion |
| Hi3DGen | Scene Meshes | LoRA-tuned NeRF, score distillation, compositional prompts |

Table 1. Output and capabilities of selected 3D generative models.

across two key modules—Lens Studio (on-device) and a Node.js backend (offloaded inference). Below is a breakdown of each stage:

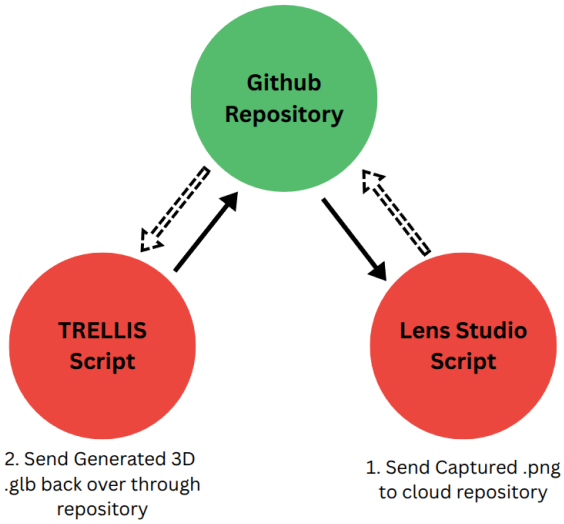


Figure 3. Snap3D pipeline: From AR gesture-based capture to 3D visualization.

### 5.1. Step 1: Capture & Region Selection

The user initiates a capture by performing a pinch gesture with both holds visible to the Snap Spectacles' front-facing cameras. A scanner prefab (bounding box) is then instantiated to define the region of interest. The active region is rendered to a texture object, providing a 2D snapshot of the real-world selection.

### 5.2. Step 2: Image Encoding

Once captured, the region is converted into a base64-encoded JPEG. This encoded string provides a compact format for image transmission and is sent to our GitHub repo through API calls.

### 5.3. Step 3: 3D Generation

Upon receiving the image payload, the backend processes it using the TRELLIS generative model. TRELLIS lifts the 2D image into a latent representation and outputs a 3D mesh in the GLB format. This step is currently asynchronous due to model size and GPU inference requirements.

### 5.4. Step 4: AR Visualization

The returned 3D asset is streamed back to Lens Studio, which dynamically imports and anchors the model at the original region location.

## 6. Conclusion

All in all, the end-to-end latency for image encoding and caption-based interpretation averaged under 10 seconds. And for full 3D mesh generation, inference time ranged from 8 to 20 seconds depending on input complexity.

While this system is not yet suitable for real-time object generation in consumer-facing applications, it demonstrates a functional proof of concept for integrating wearable AR with generative 3D model synthesis.

| Metric | Observed Value |
|---|---|
| Image Encoding Time | 5-10 s |
| Network Latency | 1-3 s |
| 3D Model Generation Time | 14–18 s |
| AR Model Placement Latency | 10-13 s |

Table 2. Averaged System performance metrics across pipeline trials.

## References

[1] Chen, Y., Gojcic, Z., Park, J., Zhang, Y., Wang, W., Pollefeys, M. (2023). *Structured 3D Latents for Scalable and Versatile 3D Generation*. https://arxiv.org/pdf/2412.01506

[2] Xu, K., Wang, W., Zhu, Y., Zhang, X., Li, B. (2023). *Hunyuan3D 2.0: Scaling Diffusion Models for High Resolution Textured 3D Assets Generation*. https://arxiv.org/abs/2501.12202

[3] Fang, Y., Qian, H., Zhu, M., Wu, J. (2024). *Hi3DGen: Hierarchical 3D Scene Generation from Multimodal Prompts*. https://arxiv.org/html/2503.22236v1

[4] Snap Inc. (2024). *Lens Studio Documentation*. Retrieved from `https://docs.snap.com/lens-studio`