

SENIOR CS CAPSTONE PRESENTATION · SPRING 2026

# HomeViewer: Spatial Paint Visualization on Apple Vision Pro

Hunter  
Thiel

St. Norbert  
College

PART 1 — PROJECT DESCRIPTION

# The Problem



# Spatial Understanding

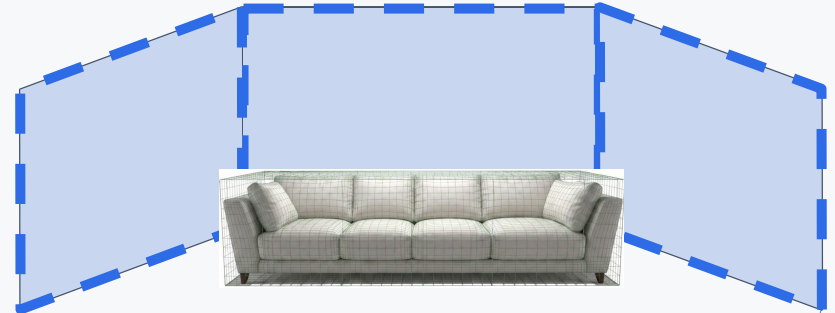
Two systems run simultaneously:

## 1 — PLANE DETECTION

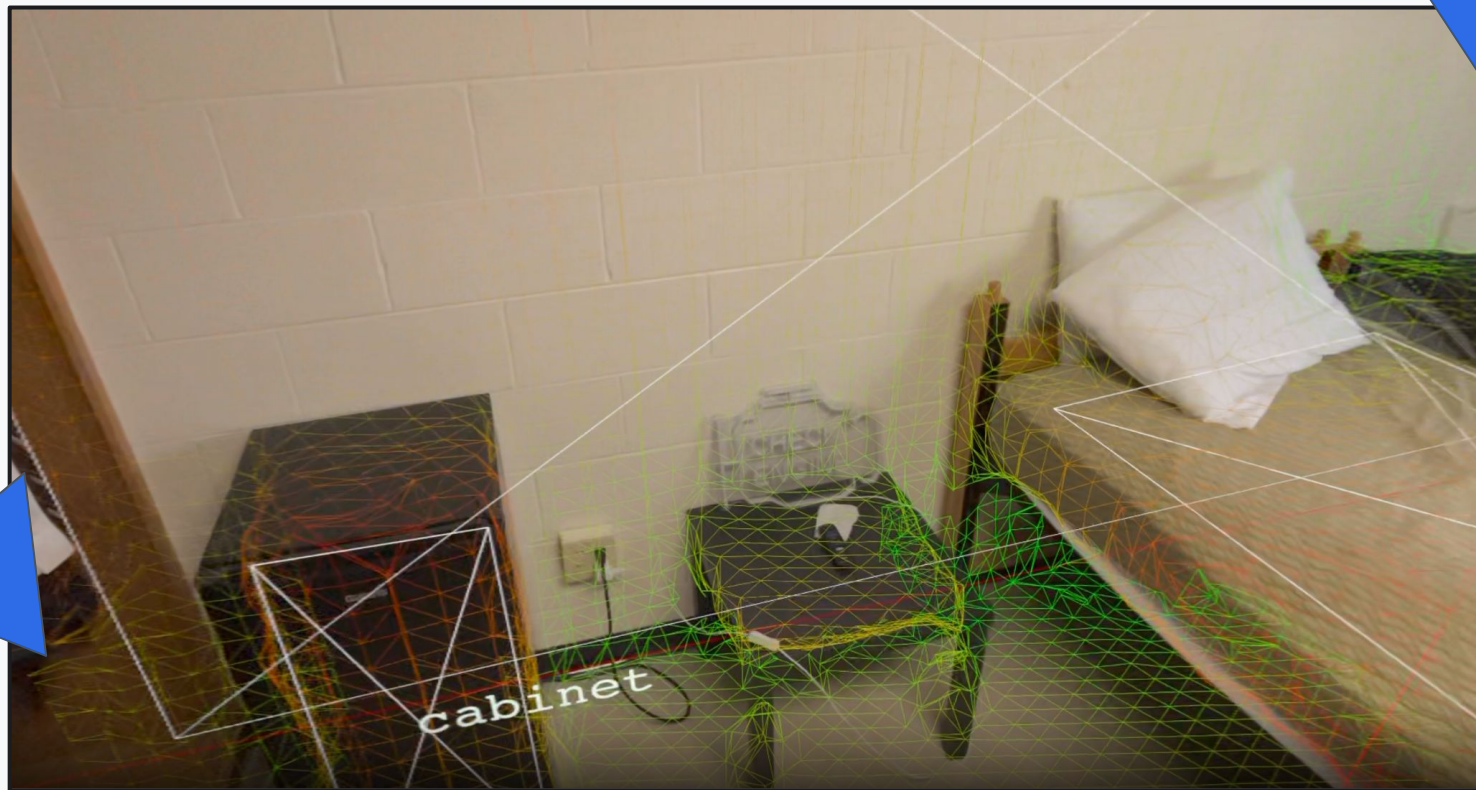
ARKit identifies flat vertical surfaces and anchors an invisible digital rectangle to each one.

## 2 — SCENE RECONSTRUCTION

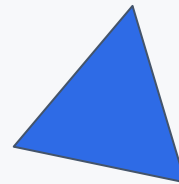
The headset's scanner builds a continuous 3D mesh of the room — furniture, objects, frames, everything.



# Managing the Mesh



# Physically Based Rendering (PBR)



HomeViewer uses PBR materials with three controls:

## Roughness

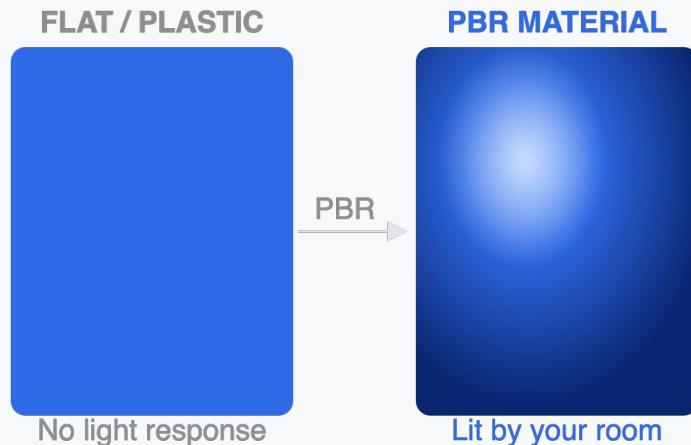
Matte vs. glossy surface — how diffuse or focused reflections are

## Clearcoat

Adds a lacquer-like top layer — the sheen of a fresh paint job

## Specular

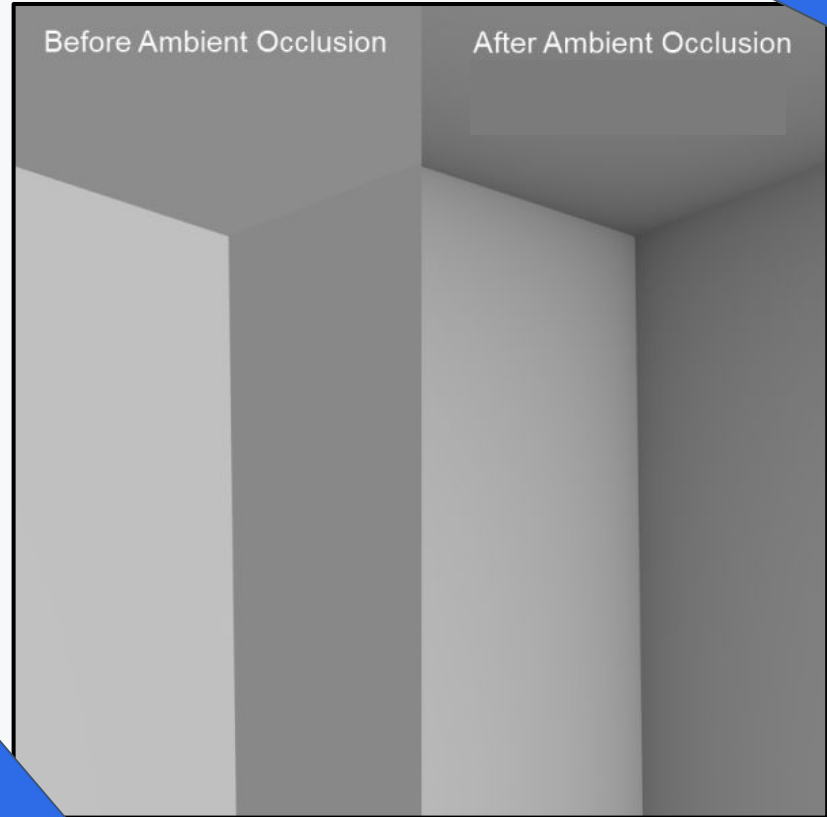
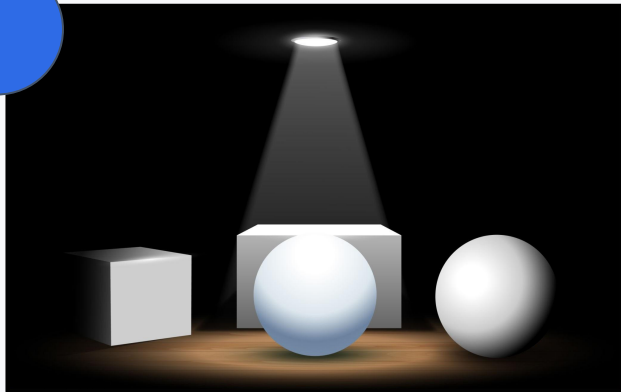
Controls intensity of bright highlight response



# Ambient Occlusion & Lighting

## Virtual Environment Probe Component

This provides a real-time estimated ambient intensity and color temperature that ARKit derives from the camera feed.



# Exceptional

## Door & Window

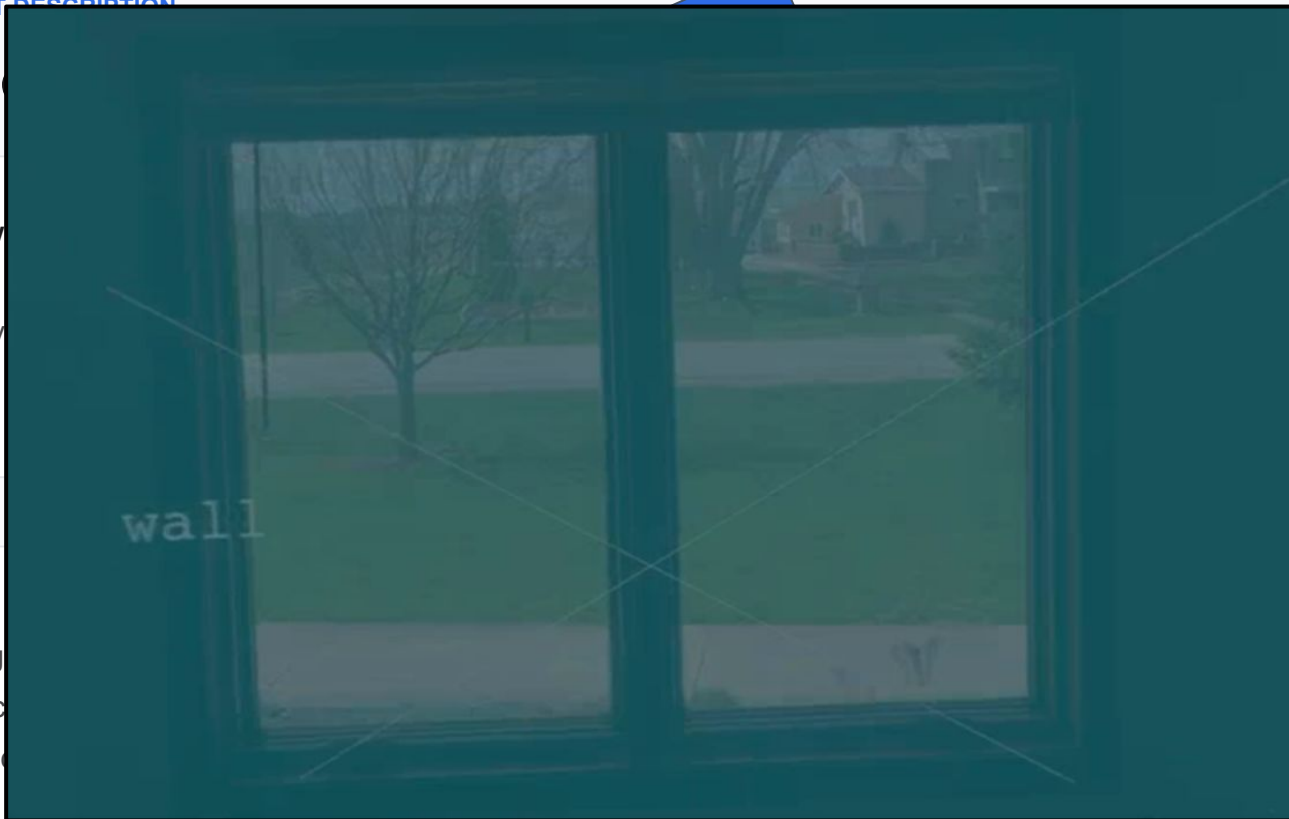
Attempted, but not  
reliably classified

ights

## Texture Import

Color swapping  
user-imported  
stone) was not

interior  
nal



wall



PART 2

# Demonstration

---



# What to Watch For

**01**

## Wall Detection

How quickly the app finds and outlines walls on startup

**02**

## Gestures

Selecting a wall by looking at it and pinching, as well as adjusting the depth with pinch and slide

**03**

## Real-Time Color Swap

Per-wall color changes and the all-walls-at-once mode

**04**

## Manual Adjustments

Dragging corner anchors and stopping updates to handle fine-tune adjustments

**05**

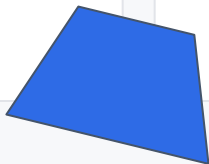
## PBR Controls

Watching the material adjust in real-time

**06**

## Sinking and Offset Sliders

The sinking and depth offset controls and what happens when miscalibrated



PART 3

# Learning & Future Development

---

# Learning & Development

## STARTING OUT

### Learning from Zero

Swift and visionOS simultaneously, with no prior Swift experience.

## DEBUGGER

### Seeing the Mesh and Surfaces

Debugger was a huge tool for seeing both the Plane Detection Provider and the Scene Reconstruction Provider.

## COOLEST SOLUTION

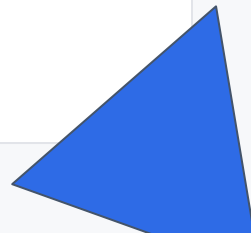
### Z-Fighting and Sinking

The mesh depth issue took more time than anything else. Multiple strategies tried before the layer-separation solution worked reliably.

## STARTER PROJECTS

### Apple Tutorial Projects

Apple's documentation provided some tutorials for learning swift and visionOS development



# Future Development



**01 Better Door & Window Occlusion**

**02 Full Texture Import**

**03 Exterior Wall Testing/Support**

**04 Pre-built Color Palettes**

**05 Mesh Improvements**

**06 Lighting Improvements**



# Questions?

Hunter Thiel

Computer Science & Mathematics · St. Norbert College Capstone 2026

