Spatial LLM

GEOIT1501

Bridging the Gap Between Natural Language and 3D Scans

Under supervision of:

Dr. Liangliang Nan - TU Delft

Dr. Shayan Nikoohemat - ScanPlan

Team members:

Mark van der Meer - Group leader

Hongyu Ye - Technical manager

Segher ter Braak - Technical manager

Julia Pille - Reporting manager

Neelabh Singh - Communication manager

TU Delft

Scanplan

ScanPlan

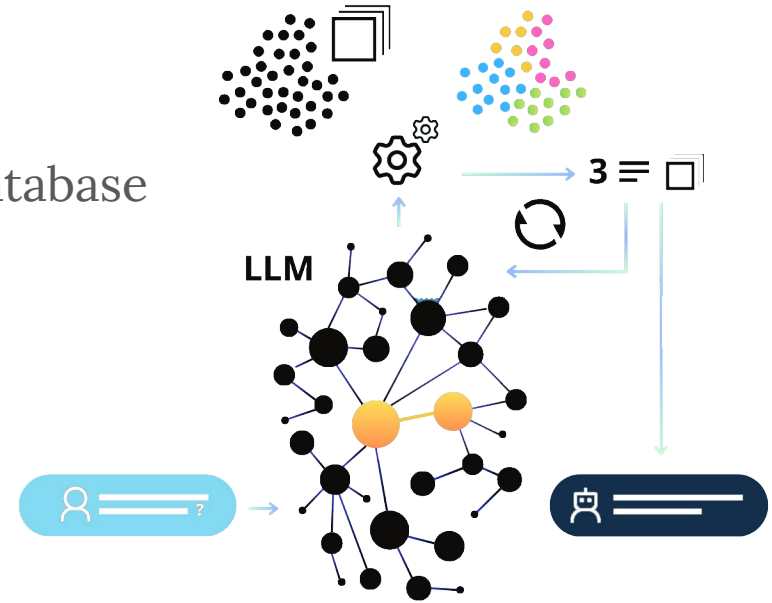
- **Motto:** Make point cloud data accessible, intelligent, and usable anywhere. Their AI-driven platform stores, classifies, and streamlines raw scans into BIM, simplifying complex 3D workflows.
- **Need:** A chatbot interface so non-experts can get quick, intuitive insights.

The Goal

- **Summary:** Develop a chatbot that understands natural-language queries and retrieves relevant spatial and visual evidence.
- **Goal:** Bridge human language with spatial-visual reasoning.

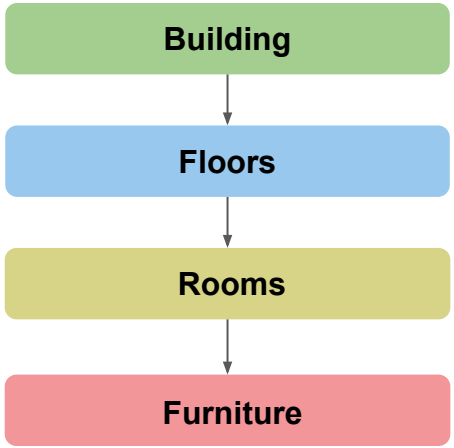
Global Overview

- **Model:** Azure Open AI: gpt-4o-mini
- **Input:** User questions & point cloud
- **Intermediate steps:** Function calling & database
- **Output:** Answer



Hierarchy of Data

- Should be intuitive for the user
- Use natural hierarchy of houses
- Example questions:
  - “How much does it cost to paint the walls in the living room?”
  - “How many chairs can be fitted in the kitchen?”
  - “Can the couch fit through the entrance of the house?”



Input

Raw point cloud

ScanPlan segmentation

Segmented point cloud

Extract panoramas and poses

Separating the Classes

All structural classes combined

- Beam
- Column
- Floor
- Ceiling
- Wall
- Wall exterior

All non structural classes combined

- Chair
- Table
- Window
- Door
- Plant
- etc...

Door

Chair

Table

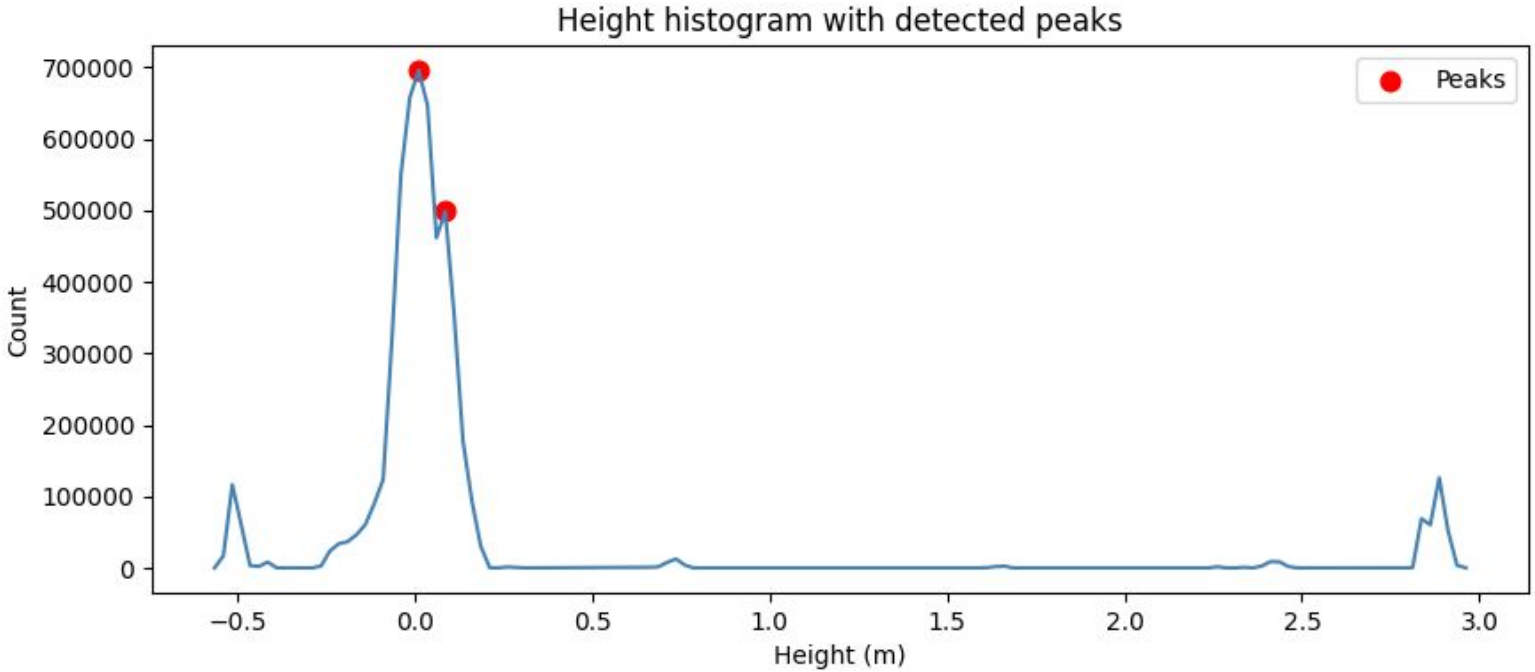
...

Etc.

Room Segmentation

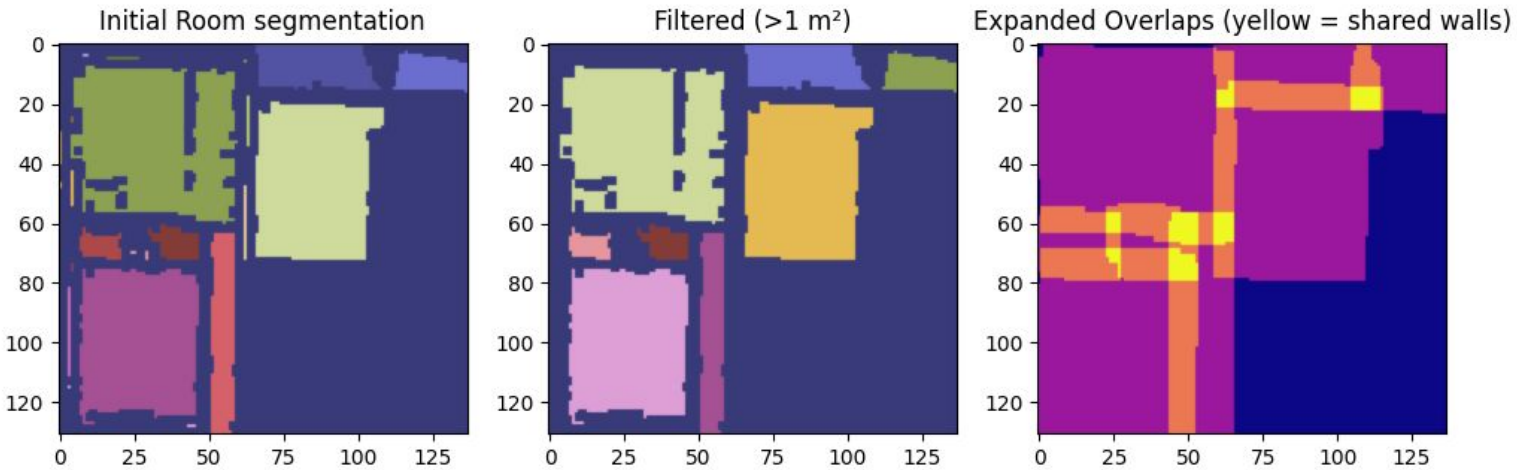
1. Measure heights
2. Slice floors
3. Make 2D floor plan
4. Label 3D points
5. Room reconstruction

Height Measurement



Floor Slicing

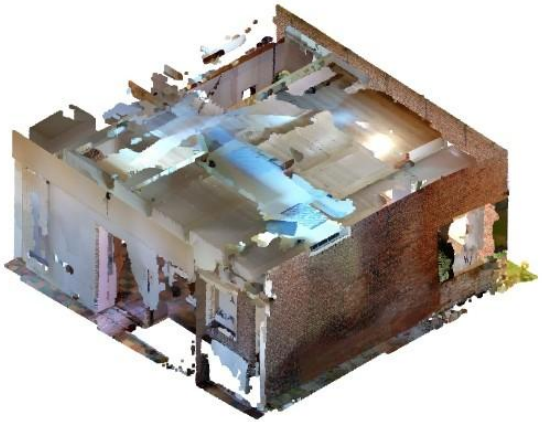
2D Floor Plan

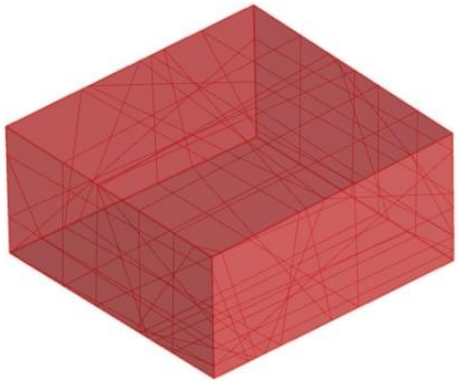


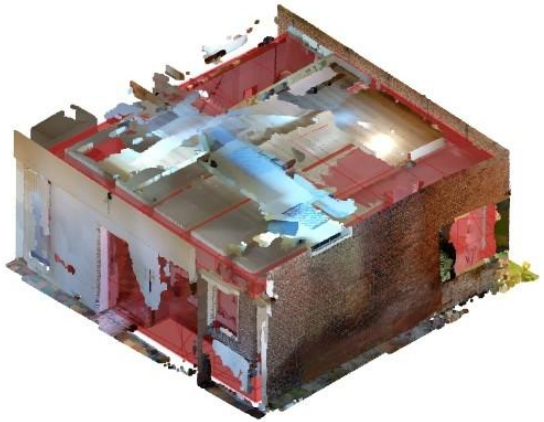
Label 3D Points

Room Reconstruction

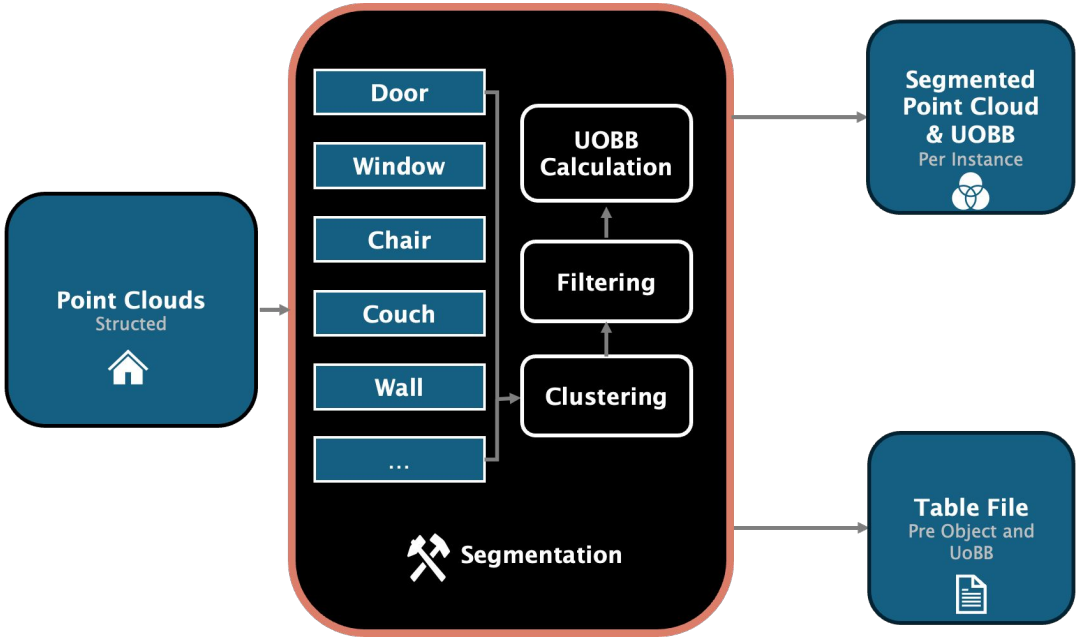
Reconstruction Using Polyfit



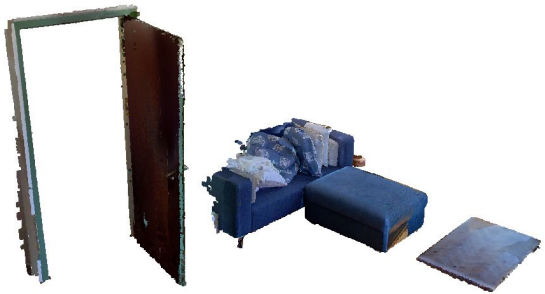


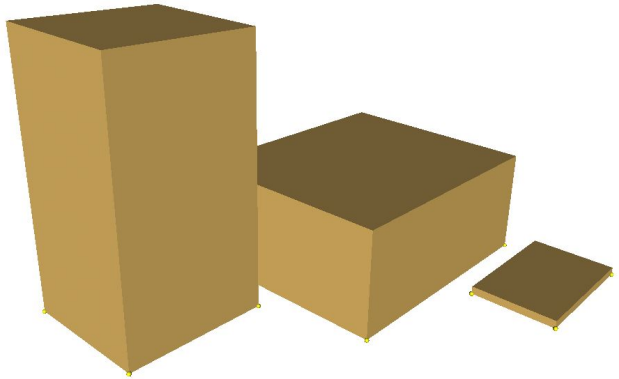


Clustering & Filtration

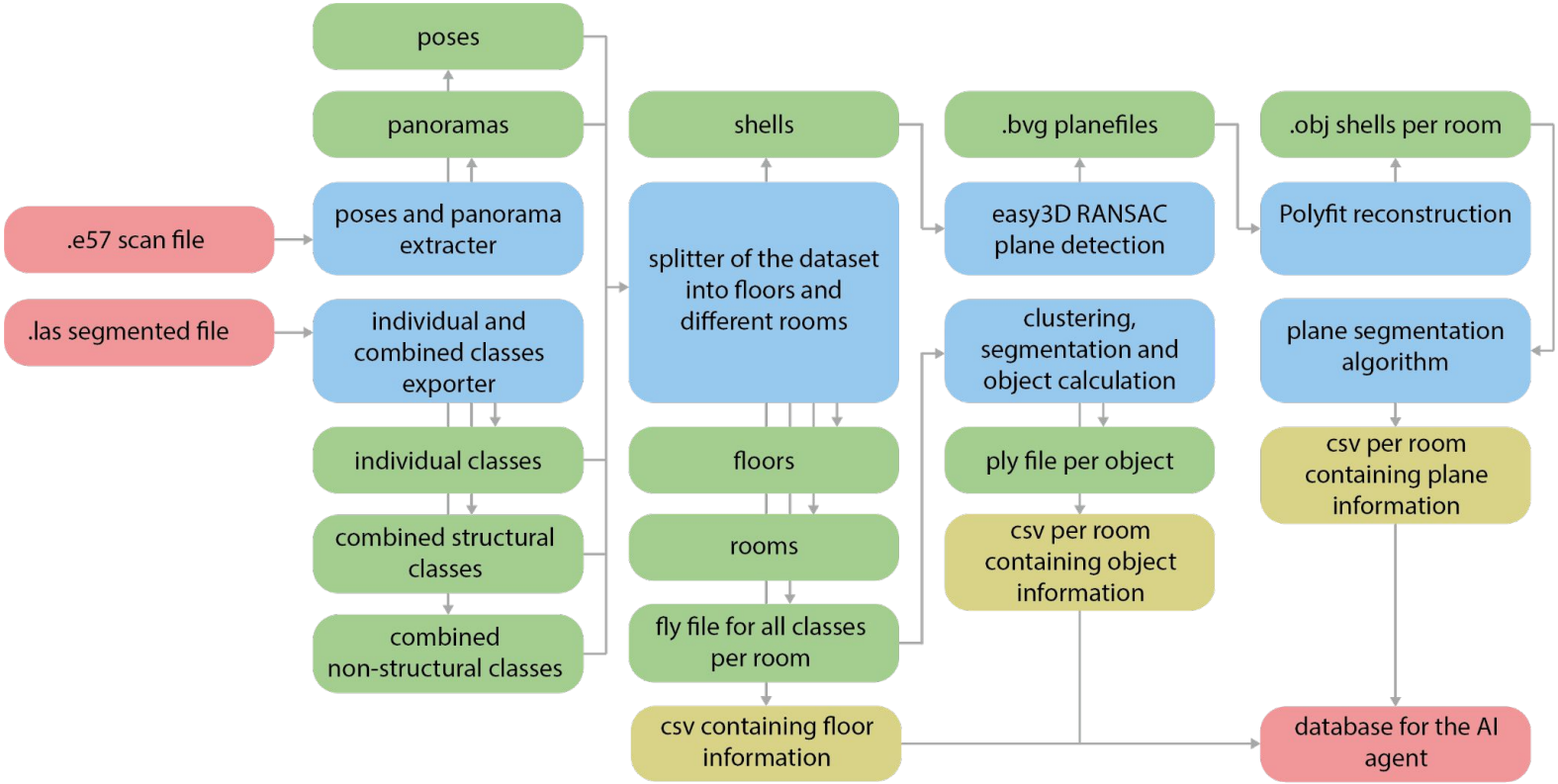


Clustering & Filtration





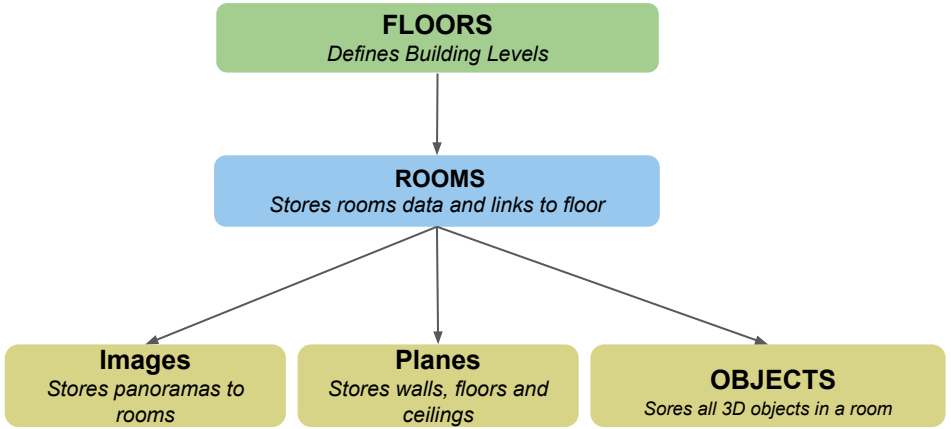
Preprocessing Pipeline



Database: *How the Agent is Built*

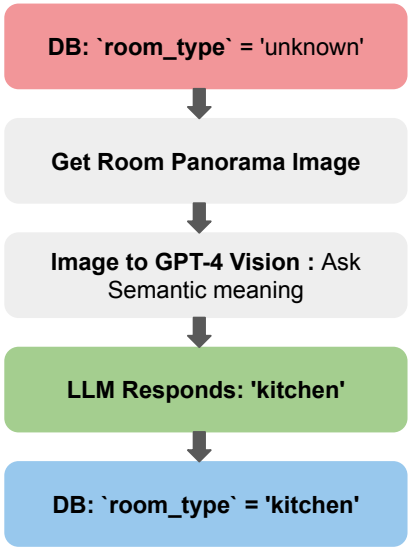
- Reads Manifests
- Populates DB
- Links Data
- Grounds the LLM

Database Schema : *Automated & Hierarchical, 5 types of tables created*



Database : *How the Agent is Taught*

AI-Driven room semantic enrichment



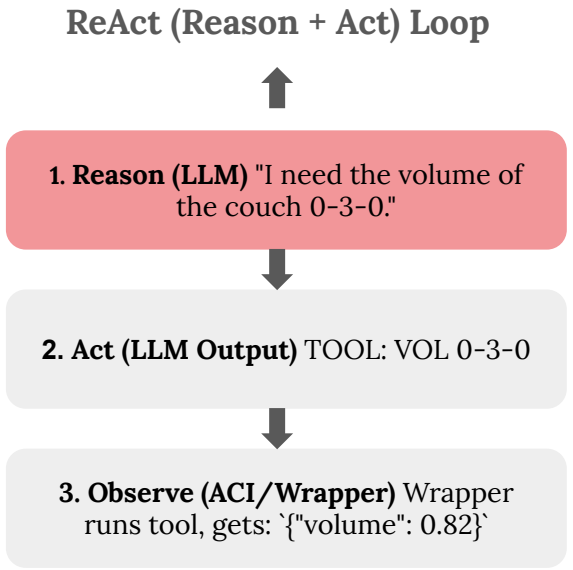
Enables semantic queries *:"How many chairs are in the kitchen?".*

Performance : *Indexed by `room_id` and `object_code` for fast, real-time lookups.*

The 'Bridge': *Giving the Agent 'Tools' to Perform Real Calculations*

- Agent-Computer Interface (ACI): API wrapper
- Based on the ReAct (Reason + Act) framework.

Head Code	Wrapper Function	Purpose
VOL	calculate_volume()	Calculates 3D mesh volume.
CLR	analyze_dominant_color()	Finds dominant RGB colors.
BBD	calculate_bbox_distance()	Measures distance between 2 objects.
VIS	visualize_point_cloud()	Generates a 3D viewer URL.

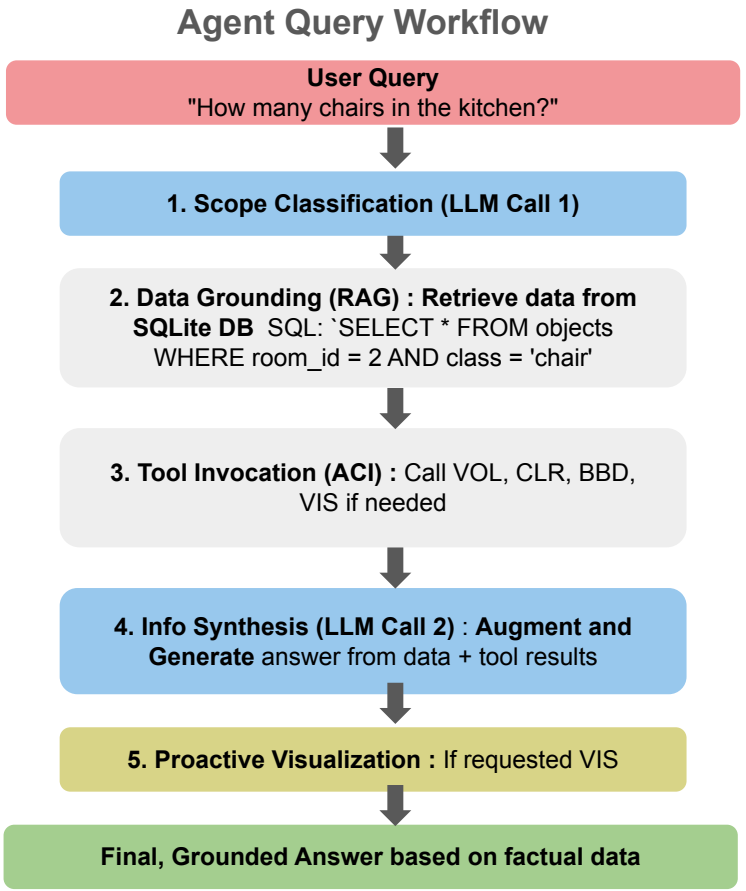


Data returned to agent in a structured ***JSON format*** *for reliable communication*

Agent's Core Architecture

5 Step hybrid RAG + ReAct Design:

- Data-driven, traceable, and accurate
- No LLM hallucination



System Prompt: *Using Declarative Programming to Define Agent Logic*

- 5 Characteristics

System Prompt: *Using Declarative Programming to Define Agent Logic*

- **Role and Persona:** Advanced Spatial AI assistant.

- “You are an Advanced Spatial AI Assistant specializing in architectural space analysis.”

System Prompt: *Using Declarative Programming to Define Agent Logic*

- **Data Grounding:** The prompt is dynamically filled with the data

- “IDENTIFICATION CODES:

  1. Room Code: <floor_id>-<room_id> (Example: "0-7" = Floor 0, Room 7)”

System Prompt: *Using Declarative Programming to Define Agent Logic*

- **Behavioral Guidelines:** Enforces a direct and practical tone.

- “Your primary mission is to provide PRECISE, DATA-DRIVEN spatial analysis.”

System Prompt: *Using Declarative Programming to Define Agent Logic*

- **Reasoning Heuristics:** Rules for multiple sources and spatial intelligence.

- “ADVANCED CAPABILITIES

  - MULTI-MODAL ANALYSIS: Integrate visual data with geometric data.

  - SPATIAl REASONING: "The chair is near the table (0.6m apart)”

System Prompt: *Using Declarative Programming to Define Agent Logic*

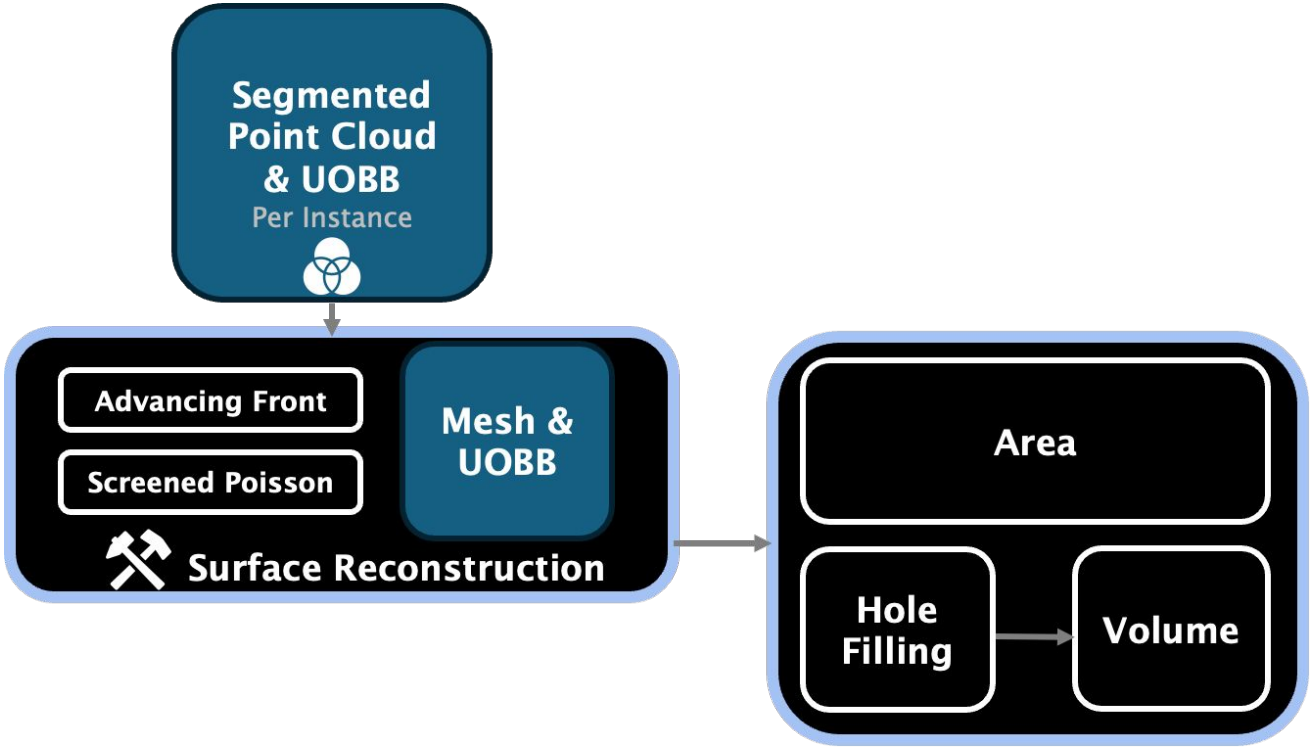
- **Output Formatting:** Mandates a concise summary.

- “RESPONSE GUIDELINES (CRITICAL - READ CAREFULLY)

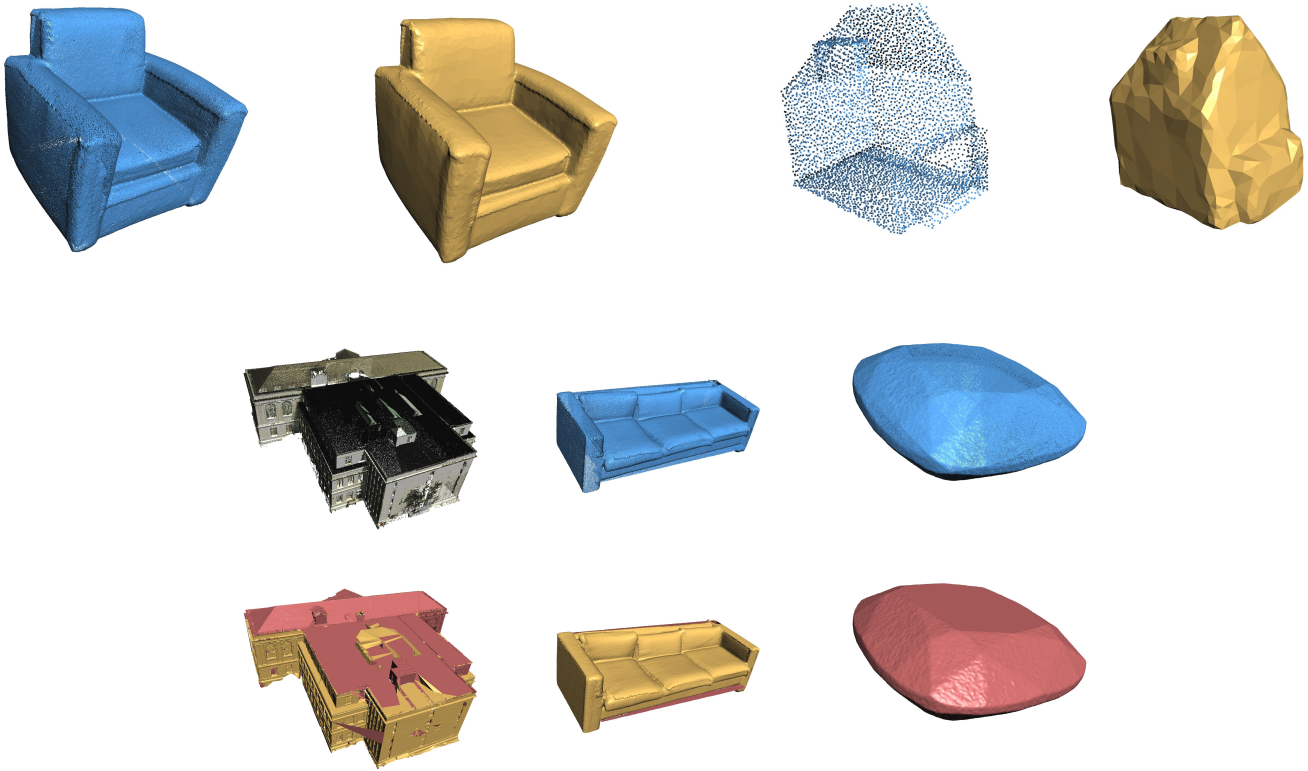
  1. DATA-DRIVEN ACCURACY

     - Base ALL answers on provided room data or API results”

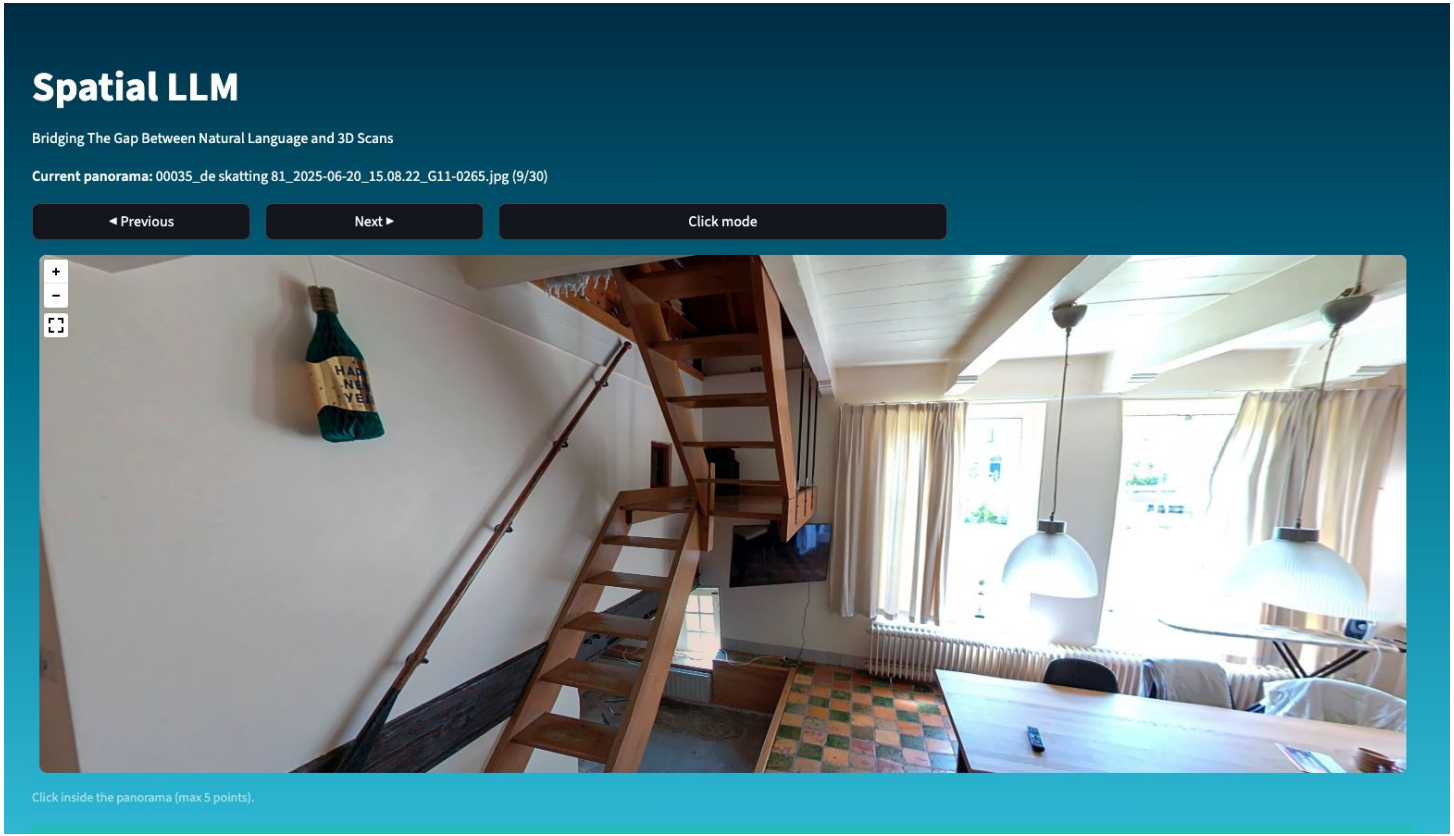
Area & Volume Computation



Surface Reconstruction — Poisson & Advancing Front

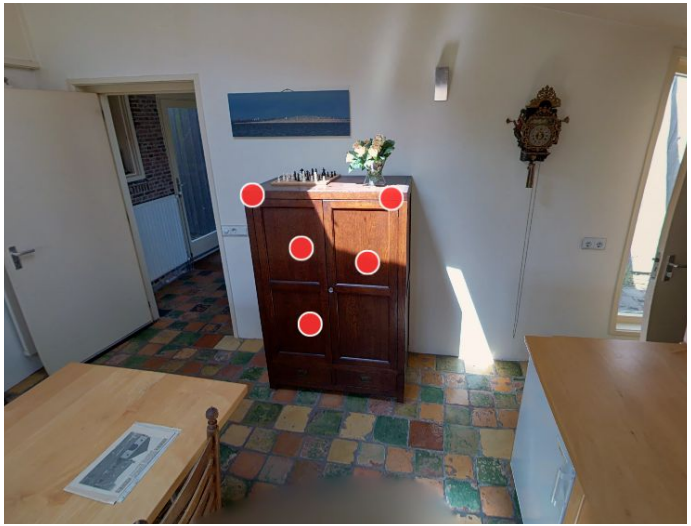


Panorama Viewer



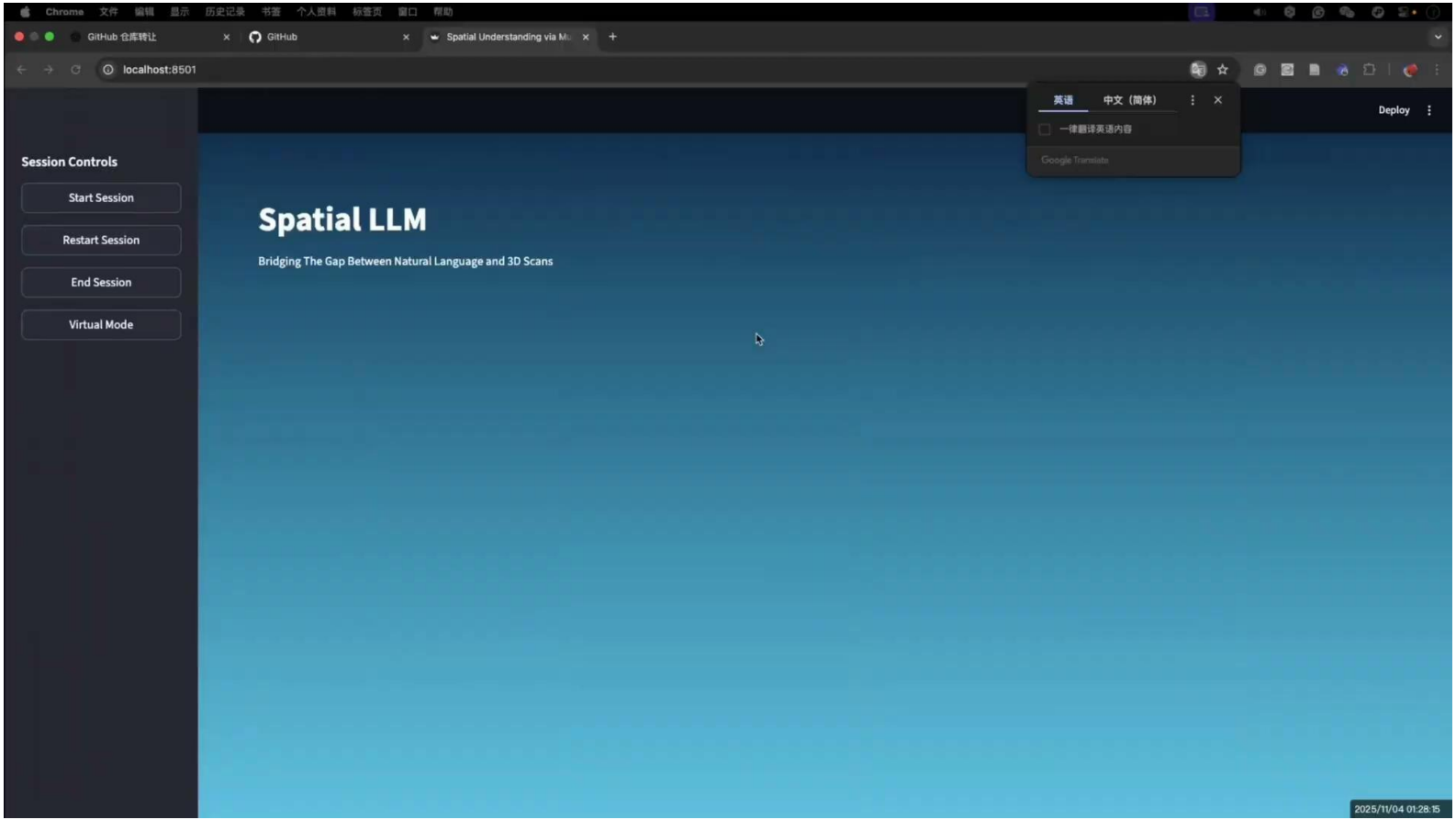
Panorama to Mask

- **Model:** SAM2



3D Clustering

- 3D points projected to image plane.
- 3D points clustered.
- Insert ply file in database


Chrome 文件 编辑 显示 历史记录 书签 个人资料 标签页 窗口 帮助
GitHub 仓库转让
GitHub
Spatial Understanding via Mu
localhost:8501
英语
中文（简体）
一律翻译英语内容
Google Translate
Deploy
Session Controls
Start Session
Restart Session
End Session
Virtual Mode
Spatial LLM
Bridging The Gap Between Natural Language and 3D Scans
2025/11/04 01:28:15

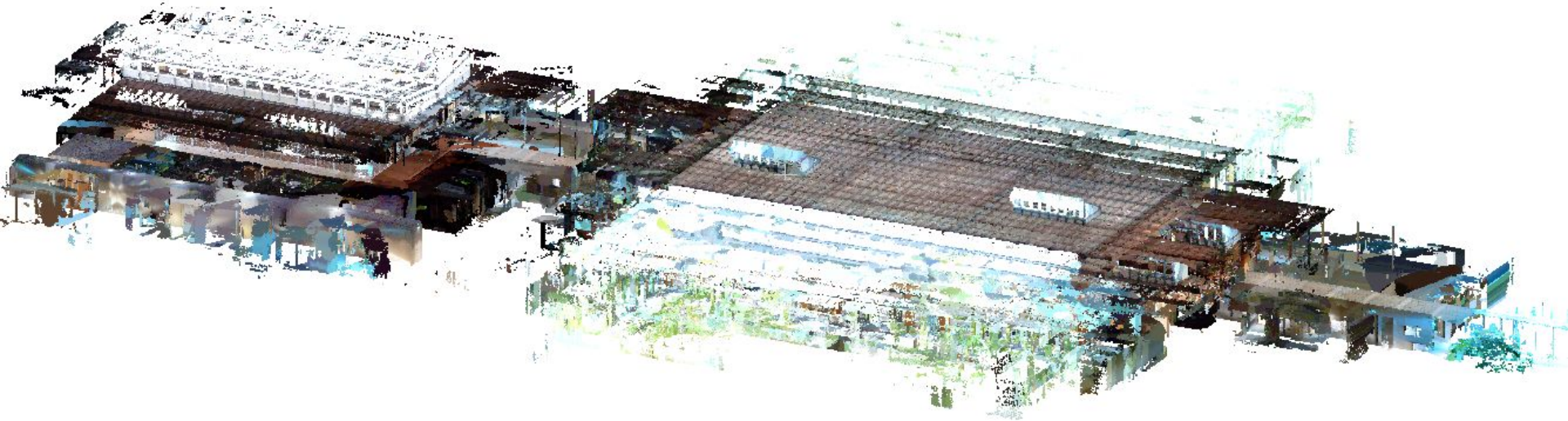
2 Datasets

- House dataset
- TU Twente dataset

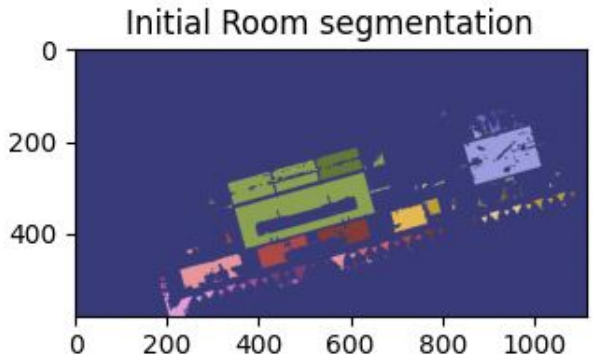
Ground Floor Results

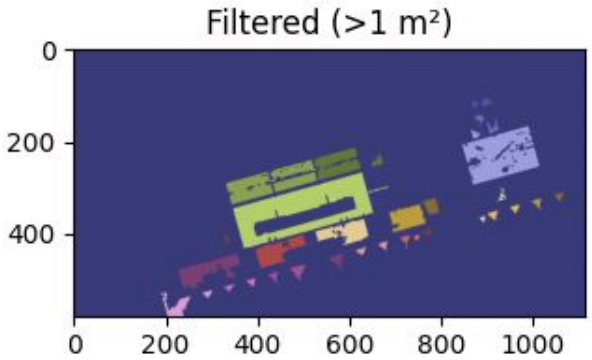
Full House Results

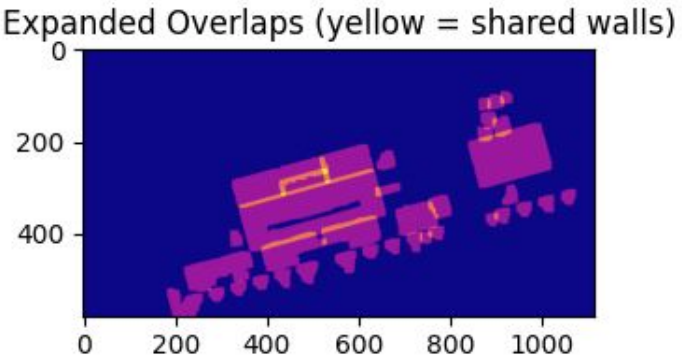
TU Twente Dataset



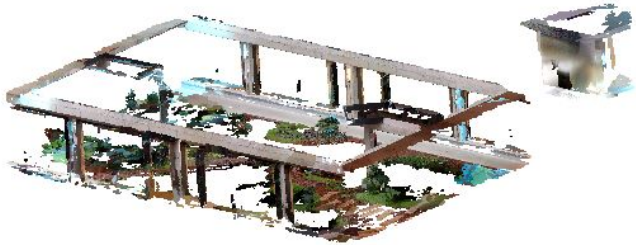
TU Twente Floor Plan

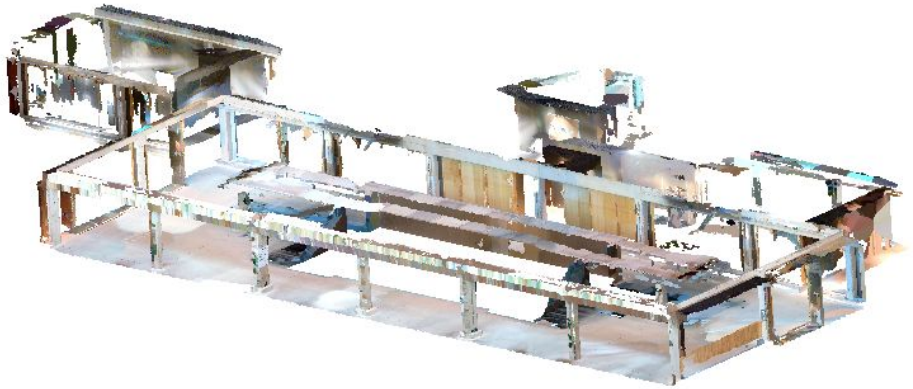




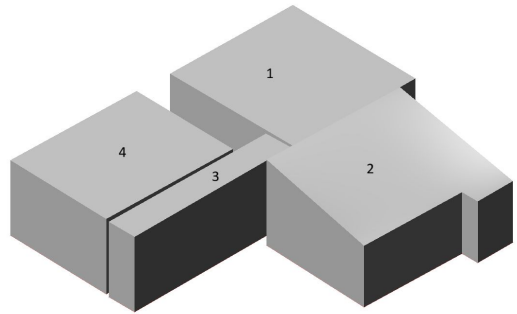


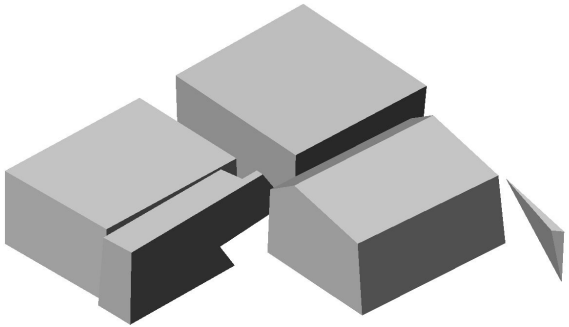
TU Twente Results





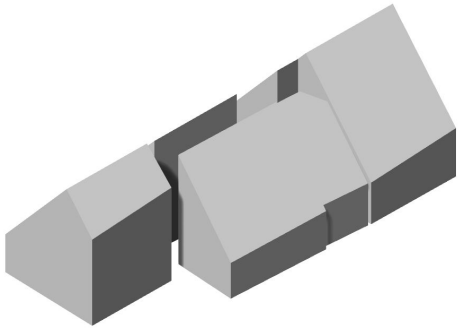
Room Reconstruction

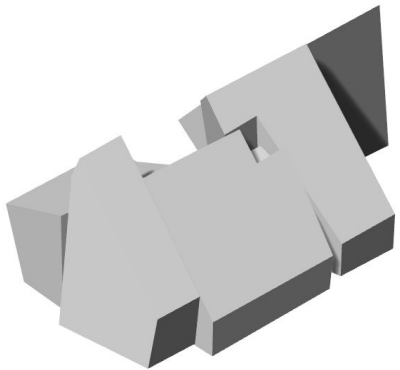


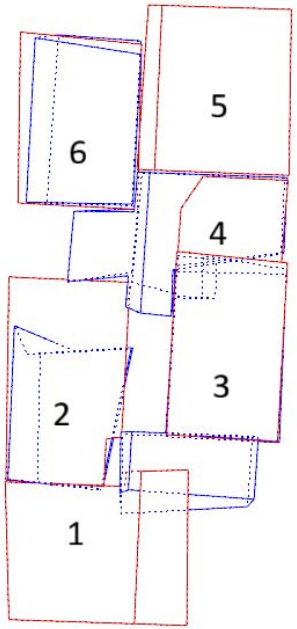


Room	Volume (%diff)	Area (%diff)	Height (%diff)
1. Kitchen	5.63	9.28	5.26
2. Extension	9.79	6.33	3.80
3. Hallway	29.30	15.23	2.62
4. Living room	0.08	0.61	0.08
Average	**11.20**	**7.86**	**2.94**

Room Reconstruction

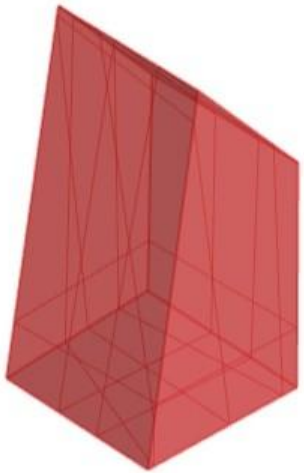




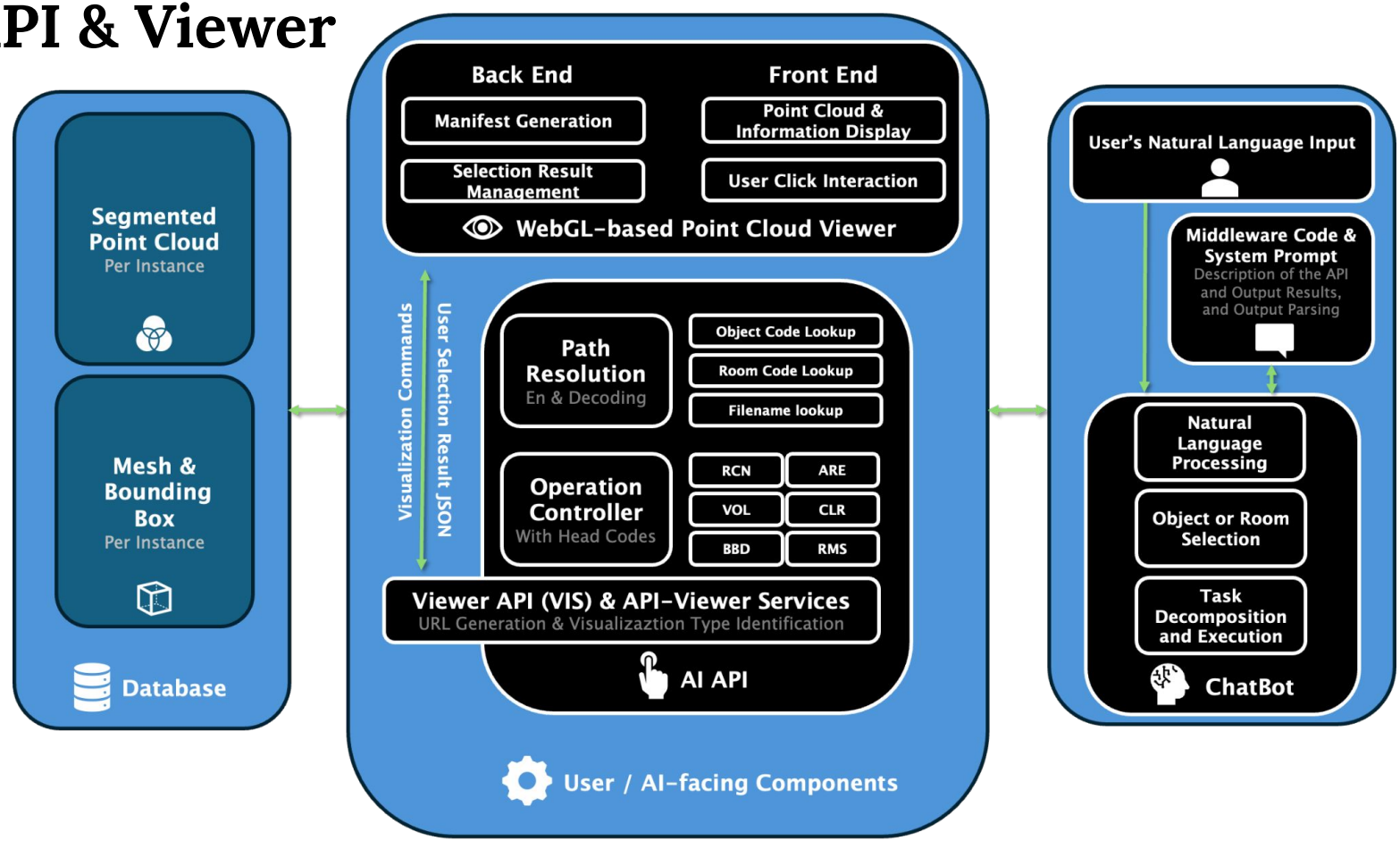


Room	Volume (%diff)	Area (%diff)	Height (%diff)
1.	67.55	49.77	1.44
2.	54.39	45.29	35.34
3.	22.54	13.60	8.78
4.	119.74	80.87	4.20
6.	27.16	19.13	8.43
Average	**58.28**	**41.73**	**11.64**

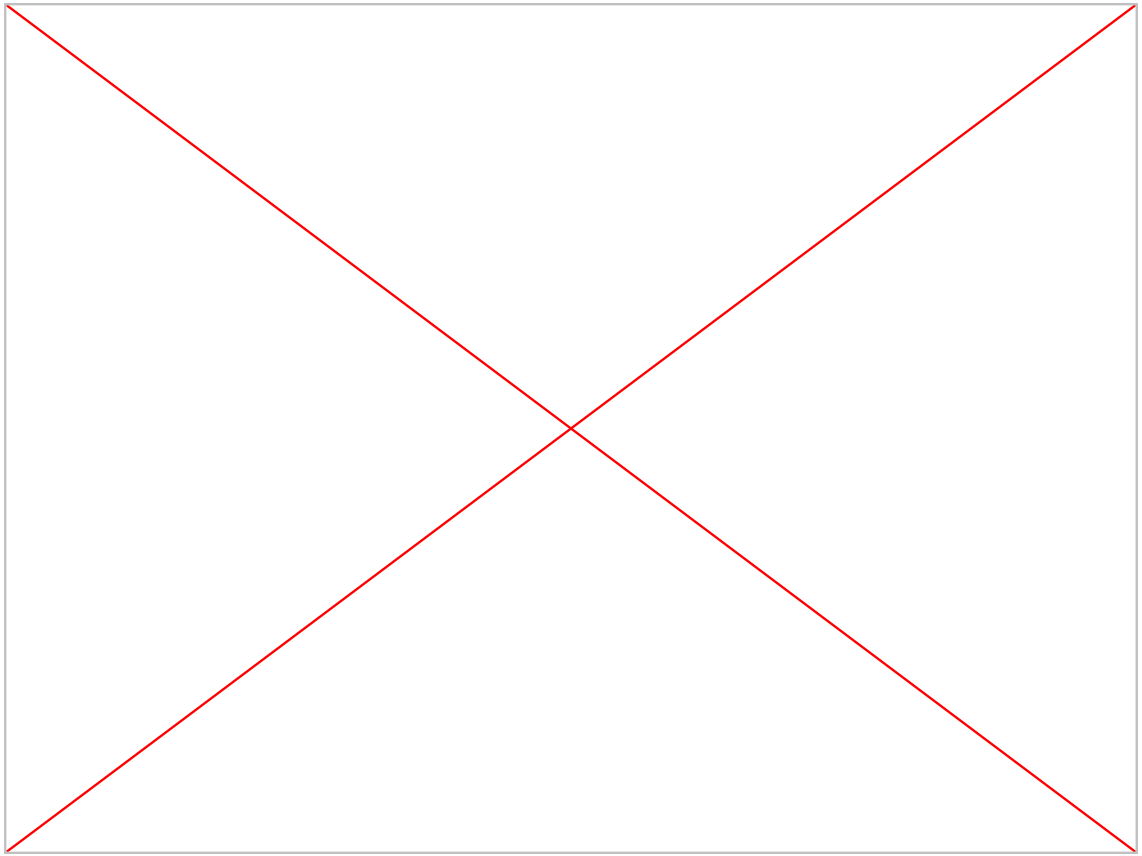
Room Reconstruction



AI API & Viewer



Demo



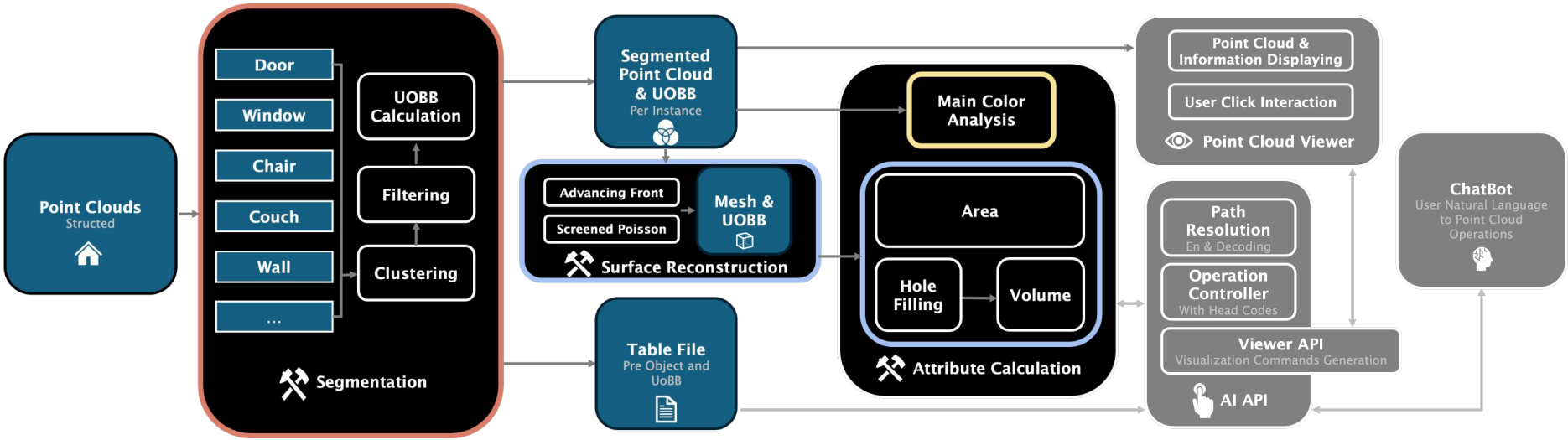
Conclusion

- Overall structure works as intended
- Room segmentation
- Hard to verify accuracy of results from agent
- Improve product based on user feedback
- Improve agent
- Integrate 3D viewer into GUI itself
- Bridged some gaps between natural language and spatial reasoning

Questions?

Slides that follow contain in detail information

Object Clustering and Attributes



The pipeline is **robust and modular**, producing room-level instances, metrics, and (when needed) meshes ready for downstream analysis.

- **Flow:** Input organized room cloud → **FEC clustering** → **adaptive filtration** → **UOBB** → optional **Poisson/AF** → **area/volume** / optional **GMM color**.
- **Assumptions:** Rooms are **moderately cluttered** indoor scenes.

Key Challenges & Future Work

Scaling from a 10-Room Prototype to a 10,000-Room Product

Challenge 1: The 'Context Stuffing' Problem

Problem: MULTI_ROOM queries (e.g., 'find largest room') load *all* 10 room summaries into the prompt. This is not scalable.

Future Solution: Scalable RAG. The agent should generate a *filtering SQL query* first (e.g., SELECT room_name, floor_area...) and only load that small, relevant result into the prompt.

Challenge 2: The 'Sequential Waiting' Problem

Problem: A query for 'volume of 5 tables' calls the VOL tool 5 times, *one after another*. This is very slow.

Future Solution: Parallel Tool Execution. The agent should be upgraded to run all 5 (independent) API calls at the same time. Agent's tool-calling logic should be re-architected

Challenge 3: The 'Semantic Labeling' Problem

Problem: Our enrichment script can identify 'kitchen', but the 3D objects themselves (e.g., 0-7-12) are not semantically labeled.

Future Solution: Advanced 2D-to-3D Mapping. Use models like SAM to project 2D image masks (e.g., of a 'chair') into the 3D cloud, to automatically label 3D objects.

Challenge 4 : Conversational Context (Statelessness)
Problem: The agent is stateless and has no "memory" of the chat history. It fails to understand follow-up questions that use context, such as "what's the color of that chair?", because it doesn't know what "that chair" refers to.

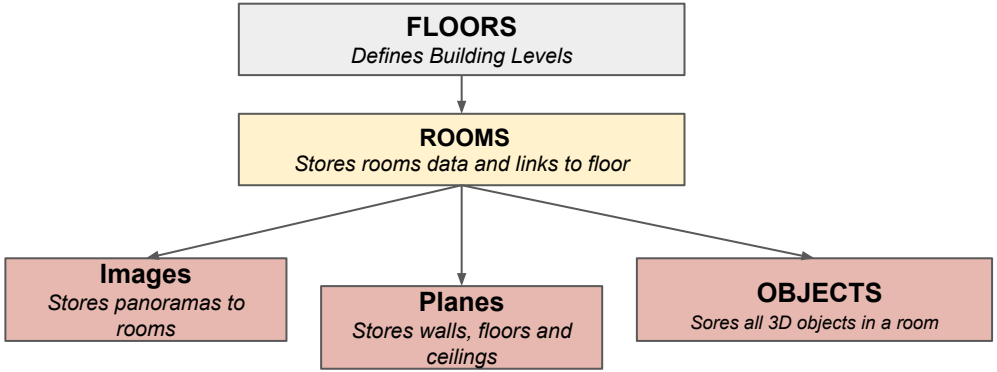
Future Solution: Implement a chat history buffer. By feeding the last 2-3 conversational turns back into the LLM with each new query, the agent can maintain context, correctly resolve pronouns (like "it" or "that room"), and support a more natural, stateful dialogue.

Database : *How the Agent is Built and Taught*

The agent cannot read raw 3D files so a SQLite database is created that it **can** read. Below is the organised folder structure and process.

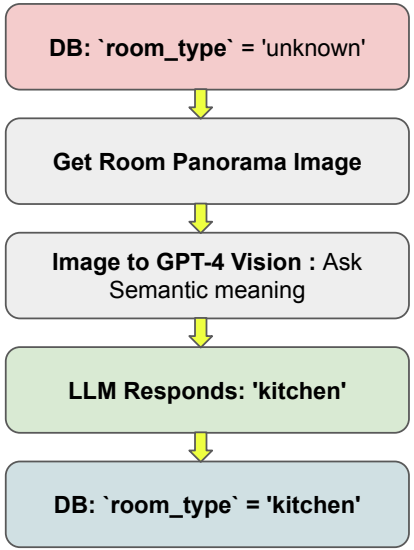
- **Reads Manifests:** The ingestion script reads all rooms_manifest.csv files from each floor_* directory to find all rooms.
- **Populates DB:** It iterates the manifests and populates a central SQLite database, creating entries for each floor and room.
- **Links Data:** It then finds all associated data (object CSVs, plane CSVs, images) and links them to the correct room_id.
- **Grounds the LLM:** This process pre-computes static facts (dimensions, plane areas) to ensure agent answers are based on data, not guesses.

Database Schema : *Automated & Hierarchical, 5 types of tables created*



AI-Driven Room semantic Enrichment

Using GPT-4V to Enable Semantic Queries



Enables semantic queries :*"How many chairs are in the kitchen?".*

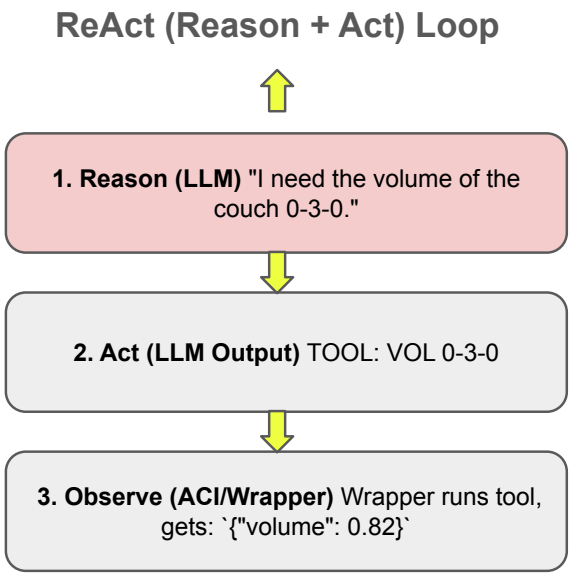
Performance : *Indexed by `room_id` and `object_code` for fast, real-time lookups.*

The 'Bridge' ACI & ReAct Loop : *Giving the Agent 'Tools' to Perform Real Calculations*

Providing agent the ability to call tools : from being a **passive language** model to **an active agent** that can take **actions**.
The **Agent-Computer Interface (ACI)** is a wrapper that connects the LLM's "brain" to external C++ "tools". This is based on the **ReAct (Reason + Act)** framework.

The API exposes a set of high-level functions through concise three-letter Head Codes : the AI agent uses it to interact with the system's data and computational functions.

Head Code	Wrapper Function	Purpose
VOL	calculate_volume()	Calculates 3D mesh volume.
CLR	analyze_dominant_color()	Finds dominant RGB colors.
BBD	calculate_bbox_distance()	Measures distance between 2 objects.
VIS	visualize_point_cloud()	Generates a 3D viewer URL.



*Data returned to agent in a structured **JSON format** for reliable communication*

Agent's Core Architecture : *Agent's 5-Stage Reasoning Cycle (RAG + ReAct)*

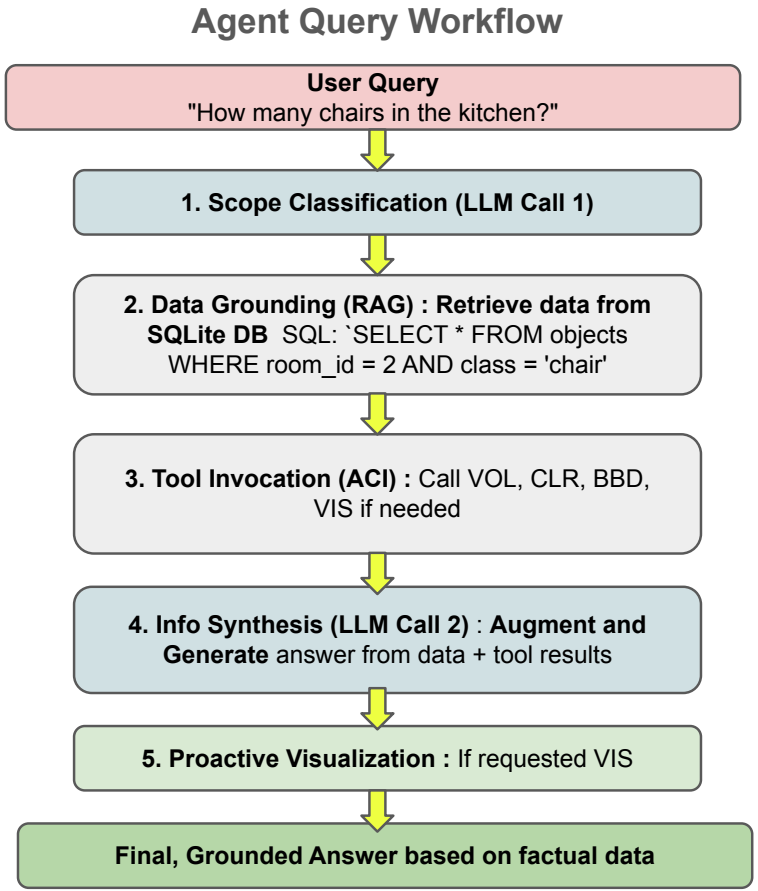
The AI agent's core architecture is based on **Retrieval-Augmented Generation (RAG)**, a framework to make LLMs "smarter" and "safer" by connecting them to a live data source.

WHY? : An LLM alone **will hallucinate. (RAG)** solves this by first **retrieving** factual data from our database and **augmenting** the LLM's system prompt with it, forcing the LLM to generate an answer based on facts.

Our 5 step hybrid RAG + ReAct Design:

1. **Scope Classification:** A quick LLM call classifies the query (SINGLE_ROOM vs. MULTI_ROOM).
2. **Data Grounding (Retrieve):** Fetches the precise data from the database based on the scope.
3. **Tool Invocation (Act):** Calls external tools (**VOL, CLR, BBD**) if needed.
4. **Info Synthesis (Augment & Generate):** The main LLM synthesizes all data into a system prompt for final answer.
5. **Proactive Visualization:** Appends a helpful 3D viewer link to the response, if VIS function is called

Impact: This process ensures all answers are **data-driven, traceable, and accurate**, solving the core LLM hallucination problem.



System Prompt, Agent's 'Brain' : *Using Declarative Programming to Define Agent Logic*

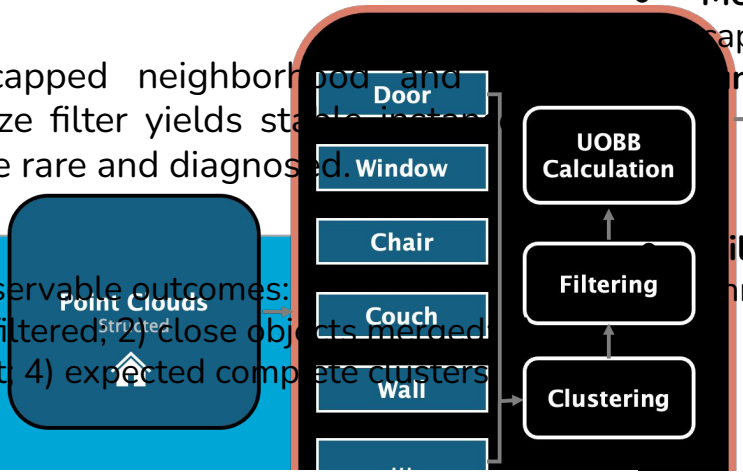
The agent's behavior is controlled by a comprehensive **system prompt**. This prompt is dynamically generated and functions as the agent's "brain," providing all rules, data, and context for its reasoning.

- **Role and Persona:** Instructs the agent to act as an advanced Spatial AI assistant with the mindset of a construction professional.
- **Data Grounding:** A large section of the prompt is dynamically filled with the exact data retrieved from the database (the RAG context).
- **Behavioral Guidelines:** Enforces a BE DIRECT & PRACTICAL tone and mandates the use of tables for clarity.
- **Reasoning Heuristics:** Provides rules for MULTI-SOURCE ANALYSIS (e.g., "prioritize 3D data") and SPATIAL INTELLIGENCE (e.g., "analyze object relationships").
- **Output Formatting:** Mandates a FINAL ANSWER INSTRUCTION to provide a concise summary with a confidence level.

Clustering & Filtration

FEC with a capped neighborhood and mean-scaled size filter yields stable instance splits; issues are rare and diagnosed.

- **Results:** Four observable outcomes: 1) small objects filtered, 2) close objects merged, single object split, 4) expected complete clusters (dominant).



- **Method (graph view):** ε-neighborhood graph with capped degree; single-pass label propagation (**nion-by-min**) ensures efficiency and stability.

$$\text{Edge } (i,j) \in \mathcal{E} \iff |\mathbf{p}_i - \mathbf{p}_j|_2 \le \varepsilon,$$

iltration: Retain clusters above a mean-scaled hreshold to suppress long-tail fragments.

$$T = \eta, \bar{s},\ \bar{s} = \frac{1}{M}\sum_c |C_c|,\ \text{keep } |C_c| \ge T.$$

Object Clustering and Attributes

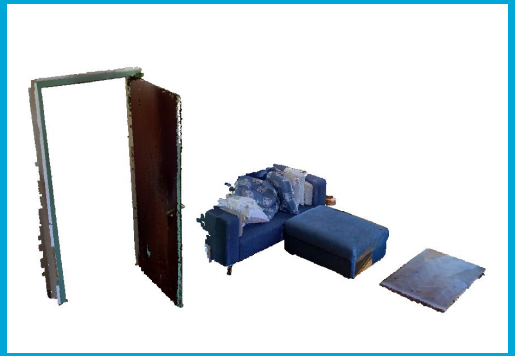
Upright Oriented Bounding Boxes

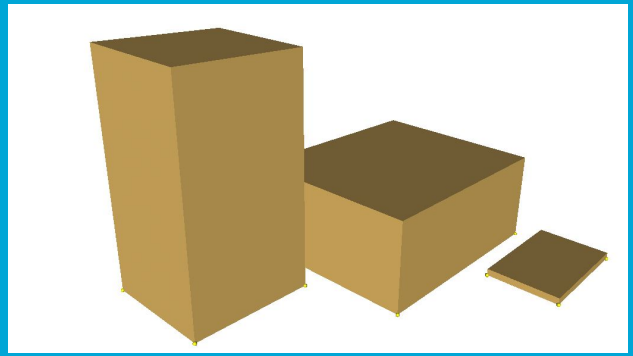
Convex-hull + rotating calipers gives tight, orientation-consistent boxes; outputs (base area, volume, yaw) are reliable.

- **Method:** Project to XY, compute convex hull, evaluate edge-aligned rectangles with rotating calipers; extrude along **z-range**.
- **Outputs:** Center, yaw, base area, volume; corners for visualization/meshing. (ℓ_x, ℓ_y, ℓ_z)

$$\ell_u = \max_i u_i - \min_i u_i,$$
$$\ell_v = \max_i v_i - \min_i v_i,$$
$$A(\mathbf{u}) = \ell_u \ell_v, \quad \min_{\mathbf{u} \in \text{hull edges}} A(\mathbf{u}),$$

- **Results:**





Visual checks show correct yaw and tightness across clusters.

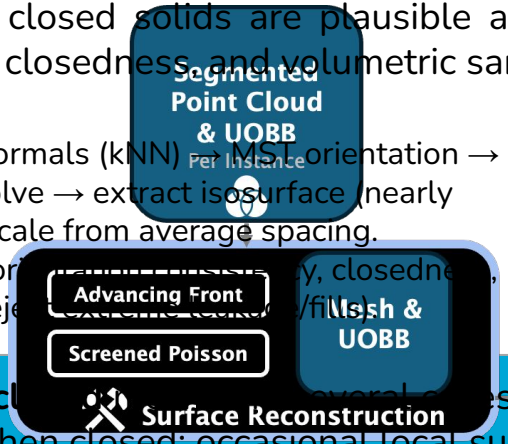
Surface Reconstruction — Poisson

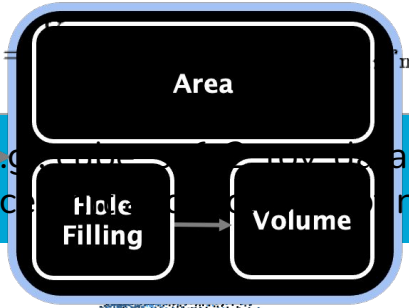
Prefer **Poisson** when closed solids are plausible and metric-accurate volumes are needed; **gate results** by orientation, closedness, and volumetric sanity.

- **Method:** estimate normals (kNN) → MST orientation → screened Poisson solve → extract isosurface (nearly uniform triangles); scale from average spacing.
- **Acceptance gates:** or [illegible], closedne[illegible], volumetric sanity (rej[illegible]).

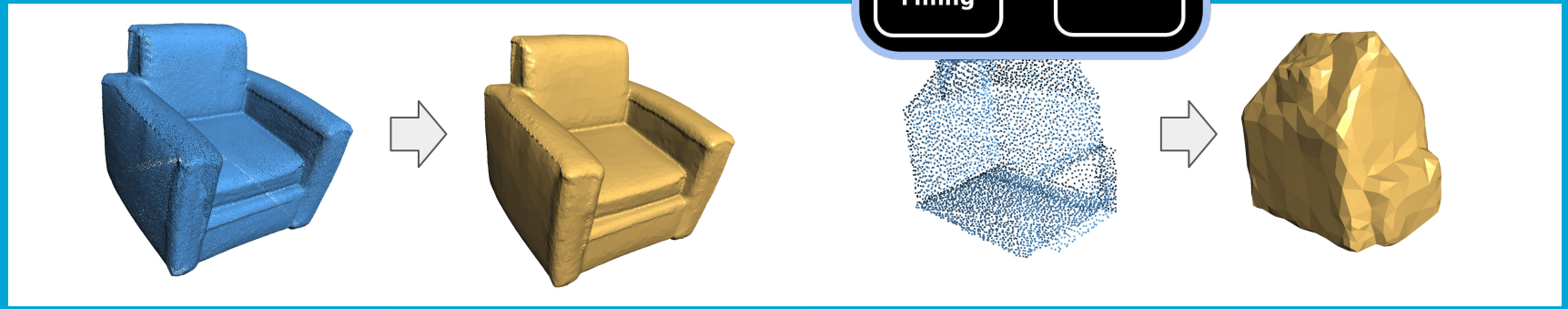
$$E(\chi) = \int ||\nabla\chi(\mathbf{x}) - \mathbf{V}(\mathbf{x})||^2 d\mathbf{x}$$

$$r = [illegible]_{max}], \text{otherwise: AF reconstruction})$$





- **Results:** produced **cl[illegible]**s (e.g. [illegible]a); **volume error** typically **−0.1% to +7.6%** when closed; occasional local surfac[illegible] noise in sparse regions.



Surface Reconstruction — Advancing Front

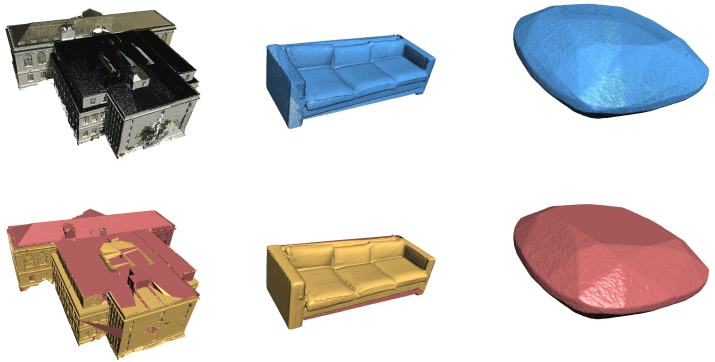
Table 3.6: Surface Reconstruction Quality Metrics

Model	Type	Method	Closed	Area (m^2)	Volume (m^3)	Error (%)
building	Reference	–	×	5941.808	19530.708	–
	Reconstructed	AF	×	8333.660	11129.577	-43.0
cube	Reference	–	×	3.919	0.362	–
	Reconstructed	Poisson	✓	5.540	0.210	-42.0
gemstone	Reference	–	×	219.385	230.032	–
	Reconstructed	AF	×	218.836	230.624	+0.3
other-ball	Reference	–	×	18550.189	205479.325	–
	Reconstructed	AF	×	20384.117	202221.695	-1.6
sofa1	Reference	–	×	8.593	1.026	–
	Reconstructed	AF	×	8.495	0.753	-26.7
sofa2	Reference	–	×	3.596	0.306	–
	Reconstructed	Poisson	✓	3.951	0.306	-0.1
toy_data	Reference	–	×	174.859	144.570	–
	Reconstructed	Poisson	✓	165.013	155.497	+7.6

- **Results:** non-closed meshes with variable volume error: **near-truth** in some (gemstone **+0.3%**, other-ball **−1.6%**), but **large under-volume** in outliers (building **≈−43%**, sofa1 **≈−26.7%**), indicating coverage/parameter limits.

AF preserves sharp/sparse geometry and adapts to local density, but often yields **open** surfaces; pair with **voxel volume** and sanity checks.

- **Method:** triangle front grows by local criteria (circumradius/edge length/proximity); naturally respects features; no global PDE.
- **Quality/topology:** frequently **non-closed**; sensitive to noise without pre-filtering; small gaps/hanging faces may remain.



Object Clustering and Attributes

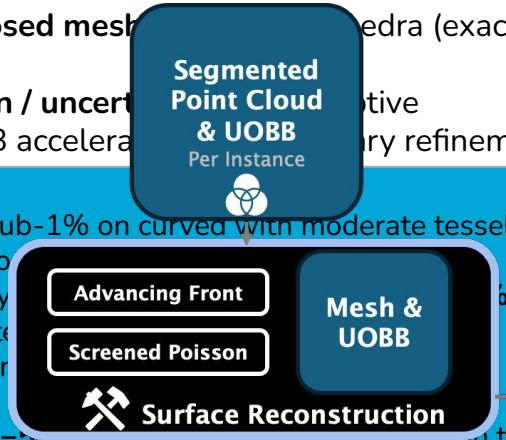
Area & Volume Computation

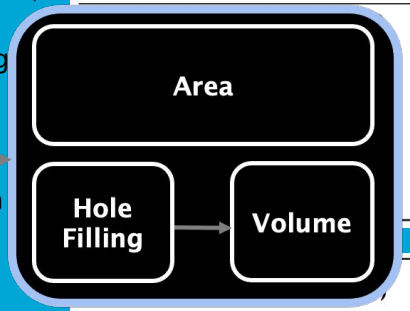
Method:

- **Area:** CGAL on triangulated faces.
- **Signed volume (for closed mesh** edra (exact for polyhedra).
- **Voxel volume (for open / uncert** tive voxelization with AABB accelera ry refinement.

$$A(\mathcal{M}) = \sum_{f=(\mathbf{v}_0,\mathbf{v}_1,\mathbf{v}_2)\in\mathcal{F}} \frac{1}{2} \|(\mathbf{v}_1 - \mathbf{v}_0) \times (\mathbf{v}_2 - \mathbf{v}_0)\|_2.$$

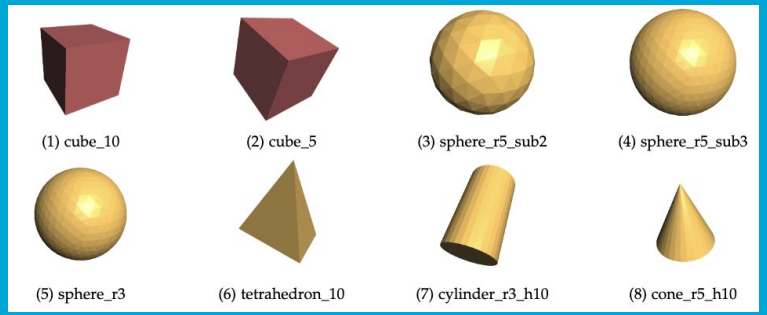
$$V_{\text{signed}}(\mathcal{M}) = \frac{1}{6} \sum_{f=(\mathbf{v}_0,\mathbf{v}_1,\mathbf{v}_2)\in\mathcal{F}} \Big((\mathbf{v}_1 - \mathbf{v}_0) \times (\mathbf{v}_2 - \mathbf{v}_0)\Big) \cdot \mathbf{v}_0.$$





Results:

- **Area** is exact on planar, sub-1% on curved with moderate tessellation;
  - Planar: **0.00%** erro
  - Curved: spheres/cy % (sig volume at higher te
- **Signed volume** is exact or well-tessellated curves;
- **Voxel volume** consistent **−5 %** on cubes, worst **−8.56 %** on thin tetra features.



Model	Closed	Faces	Expected (m^2)	Calculated (m^2)	Error (%)
			600.00	600.00	0.00
			150.00	150.00	0.00
			314.16	308.25	-1.88
			314.16	312.66	-0.48
			113.10	112.56	-0.48
			173.21	173.21	0.00
			245.04	244.38	-0.27
			254.16	253.21	-0.38

		Signed Volume		Adaptive Voxel	
		Value	Error (%)	Value	Error (%)
cube_10	1000.00	1000.00	0.00	950.00	-5.00
cube_5	125.00	125.00	0.00	118.75	-5.00
sphere_r5_sub2	523.60	505.88	-3.38	483.62	-7.63
sphere_r5_sub3	523.60	519.09	-0.86	498.82	-4.73
sphere_r3	113.10	112.12	-0.86	107.74	-4.73
tetrahedron_10	117.85	117.85	0.00	107.77	-8.56
cylinder_r3_h10	282.74	280.93	-0.64	266.88	-5.61
cone_r5_h10	261.80	260.12	-0.64	248.22	-5.19

Color Modeling with GMM

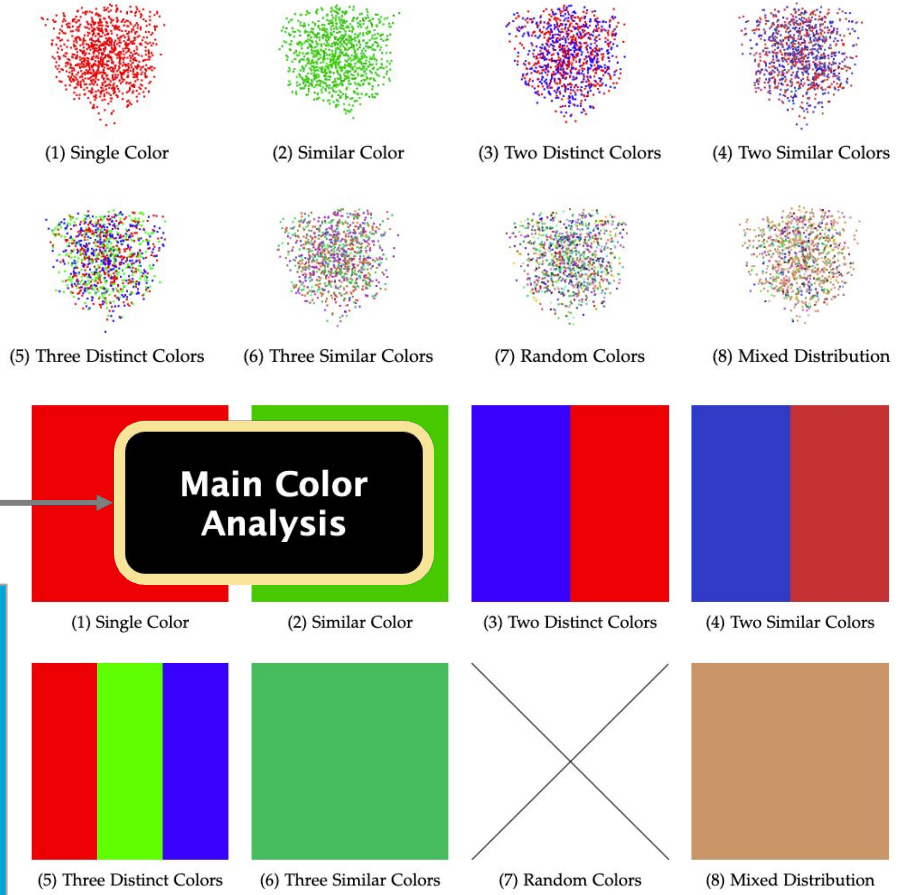
Diagonal-covariance GMM (K=1–3) robustly recovers modal colors; **BIC** prevents overfitting; merges occur when clusters are intenti

$$\mathrm{BIC}(K) = -\ldots N.$$

- **Method:** EM with k-r… **order** by BIC; MAP labeling.

Segmented Point Cloud & UOBB
Per Instance

Main Color Analysis



- **Results:**
  - Perfect on single / dual / triple distinct colors;
  - Perfect on single / dual similar colors, while triple similar colors will **merge**;
  - **Random** returns 0 clusters; **Mixed** extracts the structured mode despite 50% noise.