ASE, EFM3D & EVL: Datasets, Models & Tools for NBV

Towards Relative Reconstruction Metrics for Next-Best-View

Jan Duchscherer

VCML Seminar WS24/25

Aria Synthetic Environments

Dataset for Egocentric 3D Scene Understanding

Figure 1: [Ave+24]

ASE Dataset Overview

Dataset Content

- 100,000 unique multi-room interior scenes
- ~2-min egocentric trajectories per scene
- Populated with 8,000 3D objects
- Aria camera & lens characteristics

Ground Truth Annotations

- 6DoF trajectories
- RGB-D frames
- 2D panoptic segmentation
- Semi-dense SLAM PC w/ visibility info
- 3D floor plan (SSL format)
- *GT meshes* as `.ply` files

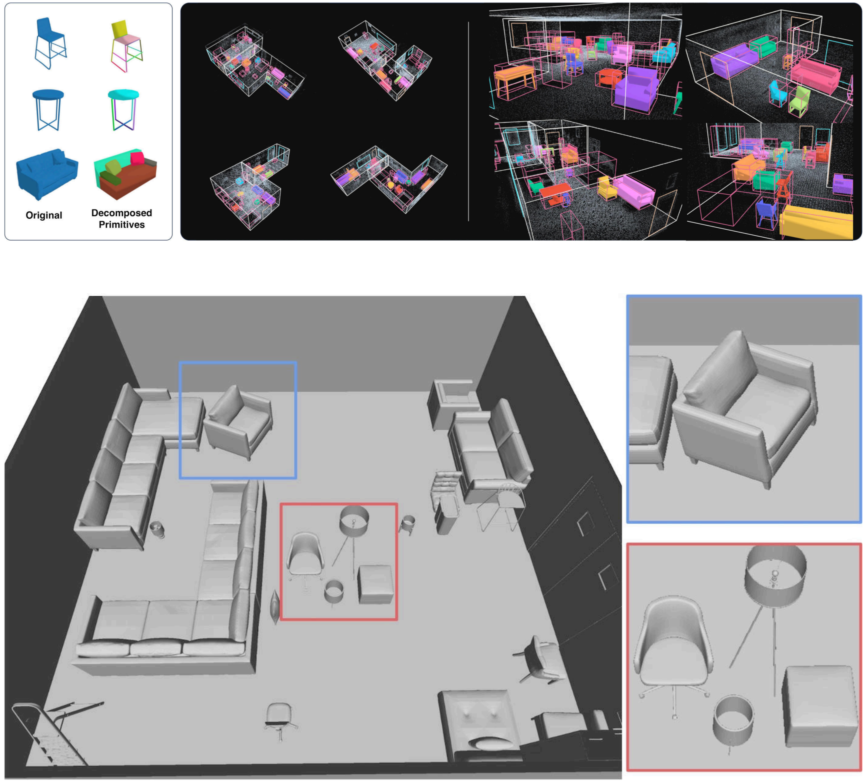

Figure 2: [Ave+24]

ASE Dataset Overview

Key Resources

- Project Aria Tools for data access
- ASE documentation [Ave+24, Met25a]

ASE Dataset Structure

```
scene_id/
├── ase_scene_language.txt          # Ground truth scene layout in SSL format
├── object_instances_to_classes.json # Mapping from instance IDs to semantic classes
├── trajectory.csv                   # 6DoF camera poses along the egocentric path
├── semidense_points.csv.gz          # Semi-dense 3D point cloud from MPS SLAM
├── semidense_observations.csv.gz    # Point observations (which images see which points)
├── rgb/                             # RGB image frames
│   ├── 000000.png
│   └── ...
├── depth/                           # Ground truth depth maps
│   ├── 000000.png
│   └── ...
└── instances/                       # Instance segmentation masks
    ├── 000000.png
    └── ...
```

EFM3D Benchmark

3D Egocentric Foundation Model:
Egocentric Voxel Lifting (EVL)

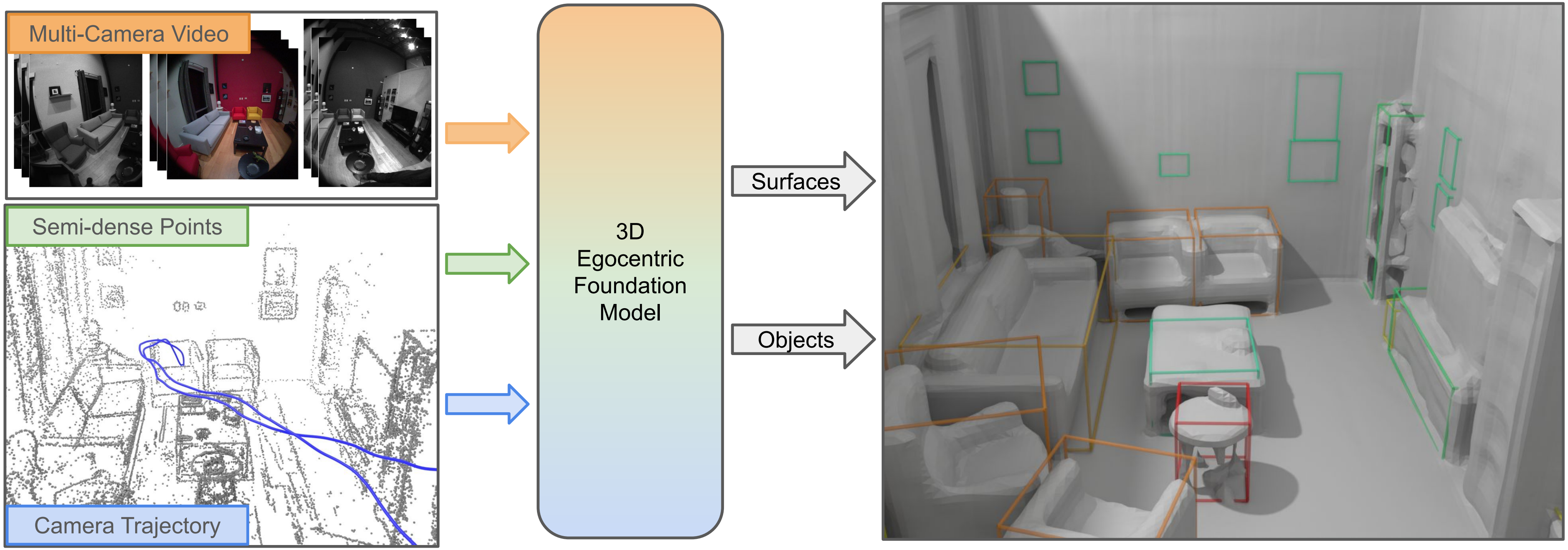

Figure 3: [Str+24]

EFM3D & EVL

EFM3D Tasks

- 3D object detection
- 3D surface regression (occupancy volumes)
  - on ASE, ADT[1], AEO[2] datasets

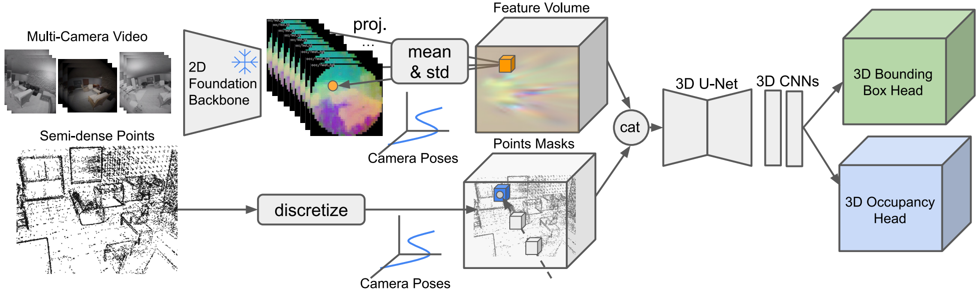

Figure 4: [Str+24]

[1] Aria Digital Twin

[2] Aria Everyday Objects: small-scale, real-world w/ 3D OBBs

EFM3D & EVL

EVL Architecture

- Utilizes **all** available egocentric modalities:
  1. multiple (rectified) RGB, grayscale, and semi-dense points inputs
  2. camera intrinsics and extrinsics
- 16.7M trainable + 86.6M frozen params
- Inherits foundational capabilities from frozen 2D model (DinoV2.5) by lifting 2D features to 3D [Str+24]

EVL: Egocentric Voxel Lifting Architecture

Model Overview

Egocentric Voxel Lifting (EVL): Multi-task 3D perception from egocentric video
Key Principle: Lift 2D image features to 3D voxel space using camera geometry

EVL: Egocentric Voxel Lifting Architecture

Input Formulation

$$\boldsymbol{X}_{\text{in}} = \{\boldsymbol{I}_1, \boldsymbol{I}_2, \ldots, \boldsymbol{I}_F, \boldsymbol{D}_{\text{semi}}, \boldsymbol{K}, \boldsymbol{T}\}$$

Where:

- $\boldsymbol{I}_f \in \mathbb{R}^{H \times W \times 3}$: RGB frames ($F$ frames)
- $\boldsymbol{D}_{\text{semi}} \in \mathbb{R}^{N \times 3}$: Semi-dense 3D points
- $\boldsymbol{K} \in \mathbb{R}^{3 \times 3}$: Camera intrinsics matrix
- $\boldsymbol{T}_f \in \mathrm{SE}(3)$: Camera pose for frame f

Multiple camera streams supported:

- RGB (high-res)
- SLAM cameras (grayscale, rectified)

Output Formulation

3D Occupancy Volume:

$$\boldsymbol{V}_{\text{out}} \in \mathbb{R}^{D_x \times D_y \times D_z \times C}$$

- Voxel grid dimensions: $D_x \times D_y \times D_z$
- C channels for:
  1. Occupancy probability
  2. Object class scores
  3. Surface normals

Detected Objects:

$$\mathcal{O} = \left\{\left(\boldsymbol{b}_i^{\text{3D}}, c_i, s_i\right)\right\}_{i=1}^{N}$$

- $\boldsymbol{b}_i^{\text{3D}} \in \mathbb{R}^9$: Oriented bounding box
- c_i: Object class
- s_i: Confidence score

EVL: Egocentric Voxel Lifting Architecture

Feature Lifting Process

1. **2D Feature Extraction**: Frozen DinoV2.5 backbone

$$F_{2D} = \varphi_{\mathrm{DINOv2.5}(I_f)} \in \mathbb{R}^{H' \times W' \times D_{\mathrm{feat}}}$$

2. **3D Projection**: For each voxel $v \in \mathbb{R}^3$, aggregate features from all frames

$$F_{3D(v)} = \mathrm{Aggregate}\left(\left\{\pi\left(T_f^{-1} v, K, F_{2D}^f\right)\right\}_{f=1}^{F}\right)$$

where $\pi(\cdot)$ is the camera projection function

3. **3D Convolution**: Process lifted features

$$V_{\mathrm{out}} = \psi_{\mathrm{3D\text{-}CNN}(F_{3D}, D_{\mathrm{semi}})}$$

ATEK Toolkit

Streamlined ML Workflows for Aria Datasets

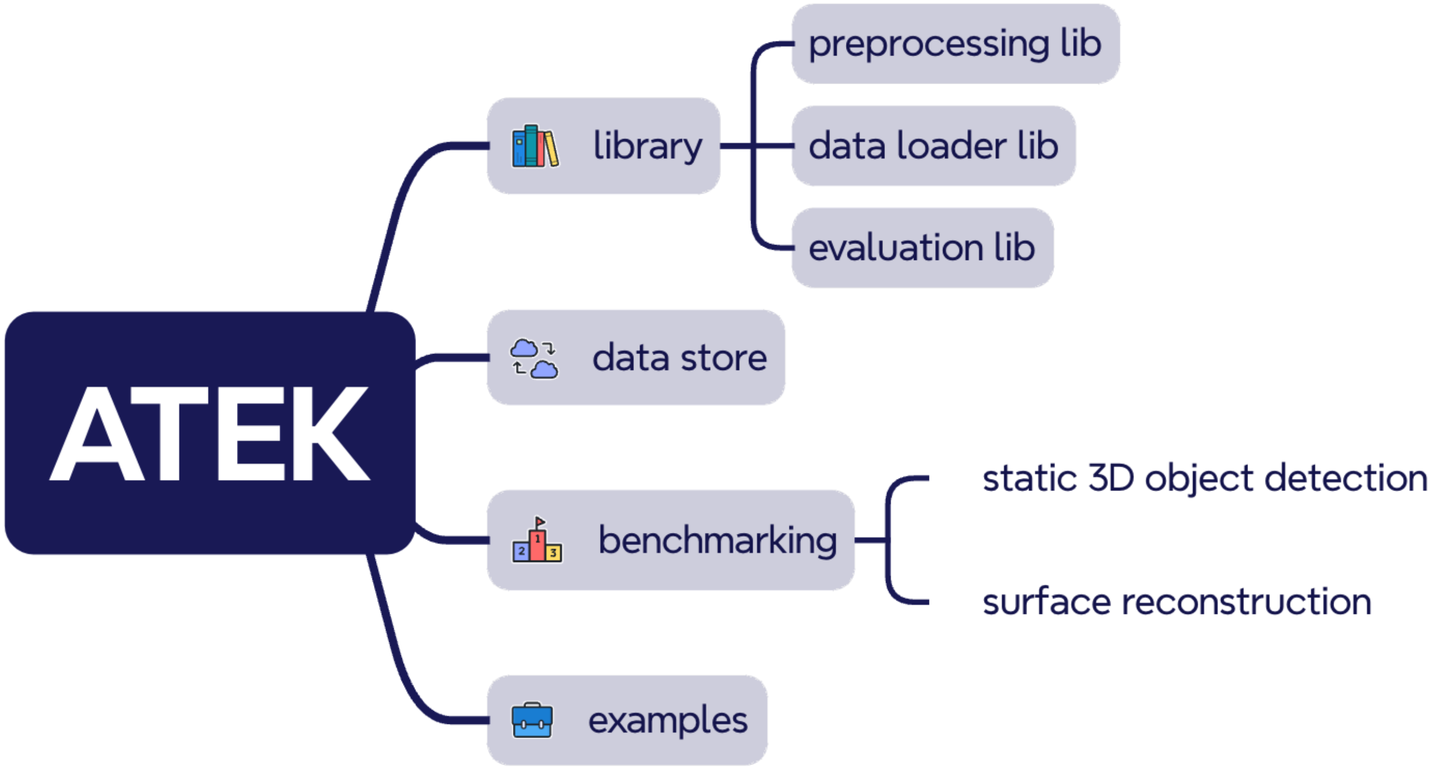

Figure 5: [Met25b]

ATEK Toolkit

ATEK Data Store

- Pre-processed for various tasks $\rightarrow$ ready for PyTorch training
- Local download or cloud streaming
- Eval metrics (accuracy, completeness, F-score) $\rightarrow$ adaptation for RRI
- Integration w/ Meta's MPS
- Various example notebooks

Provided Models

- *Cube R-CNN* [Bra+23] for OBBs
- *EFM* [Str+24] for OBBs & surface reconstruction

Resources

- ATEK GitHub [Met25c]
- ECCV 2024 Tutorial: Egocentric Research with Project Aria
- Atek Context7 ID: `/facebookresearch/atek`

ATEK provides streamlined ML workflows for rapid prototyping and benchmarking on Aria datasets.

Next Steps & TODOs

Literature Review

- Read Project Aria paper [Met25a]
- Study EFM3D & EVL architecture in depth [Str+24]
- Deep dive into GenNBV's multi-source embeddings [Che+24]
- Compare VIN-NBV vs. GenNBV: RRI prediction vs. coverage-based rewards

Technical Exploration

- Explore GT meshes (`.ply` files) in ASE dataset
- Get familiar with ATEK and ATEK Data Store
- Test mesh-based evaluation metrics (accuracy, completeness, F-score)
- Experiment with probabilistic 3D occupancy grids

Next Steps & TODOs

Implementation Goals

- Implement ray-casting for mesh-based visibility computation
- Develop entity-wise RRI computation pipeline using GT meshes
- Design 5DoF action space for scene exploration
- Build multi-source state embedding (geometric + semantic + action)
- Prototype RRI prediction network architecture

VIN-NBV: Learning-Based Next-Best-View

Key Innovation

- First NBV method to directly optimize **reconstruction quality** (not coverage)
- Predicts Relative Reconstruction Improvement (RRI) without capturing new images
- 30% improvement over coverage-based baselines
- Trained 24h on 4 A6000 GPUs [Fra+25]

VIN Architecture

Predicts RRI from current reconstruction state:

$$\widehat{\mathrm{RRI}}(\boldsymbol{q}) = \mathrm{VIN}_\theta\left(\mathcal{R}_{\text{base}}, \boldsymbol{C}_{\text{base}}, \boldsymbol{C}_{\boldsymbol{q}}\right)$$

- **Input**: Partial point cloud + camera poses
- **Features**: Surface normals, visibility counts, depth, coverage
- **Output**: Predicted RRI via ordinal classification (15 bins)

VIN-NBV: Learning-Based Next-Best-View

Relative Reconstruction Improvement

For a candidate view $\boldsymbol{q}$, RRI quantifies expected improvement:

$$\mathrm{RRI}(\boldsymbol{q}) = \frac{\mathrm{CD}(\mathcal{R}_{\mathrm{base}}, \mathcal{R}_{\mathrm{GT}}) - \mathrm{CD}\big(\mathcal{R}_{\mathrm{base}\,\cup\boldsymbol{q}}, \mathcal{R}_{\mathrm{GT}}\big)}{\mathrm{CD}(\mathcal{R}_{\mathrm{base}}, \mathcal{R}_{\mathrm{GT}})}$$

- Range: $[0, 1]$ where higher = better view
- Normalized by current error $\rightarrow$ scale-independent
- CD measures reconstruction quality

VIN-NBV demonstrates that *learning reconstruction-aware NBV policies* significantly outperforms traditional coverage-based approaches.

GenNBV: Generalizable Next-Best-View Policy

Key Innovations

- **5DoF free-space action space**: 3D position + 2D rotation (yaw, pitch)
- Multi-source state embedding: geometric, semantic, action representations
- *Probabilistic 3D occupancy grid* vs. binary (distinguishes unscanned from empty)
- Cross-dataset generalization: 98.26% coverage on Houses3K, 97.12% on OmniObject3D

State Representation

Geometric Embedding s_t^G:

- Probabilistic 3D occupancy grid from depth maps
- Bresenham ray-casting with log-odds update
- Three states: occupied, free, unknown

Semantic Embedding s_t^S:

- RGB images $\rightarrow$ grayscale $\rightarrow$ 2D CNN
- Helps distinguish holes from incomplete scans

Action Embedding s_t^A:

- Historical viewpoint sequence encoding

Combined: $s_t = \mathrm{Linear}\left(s_t^G; s_t^S; s_t^A\right)$

GenNBV: Generalizable Next-Best-View Policy

Action Space Design

$$\mathcal{A} = \underbrace{\mathbb{R}^3}_{\text{position}} \times \underbrace{SO(2)}_{\text{heading}}$$

- Approximately 20m x 20m x 10m position space
- Omnidirectional heading subspace
- *No hand-crafted constraints* (e.g., hemisphere)

RL-based framework with PPO. Reward: Δ CR between steps. [Che+24]

Reconstruction Quality Metrics

Surface-to-Surface Distance Metrics

Accuracy (Prediction $\rightarrow$ GT):

$$\mathrm{Acc} = \frac{1}{|\mathcal{P}|} \sum_{\boldsymbol{p} \in \mathcal{P}} \min_{\boldsymbol{q} \in \mathcal{M}_{\mathrm{GT}}} \|\boldsymbol{p} - \boldsymbol{q}\|_2$$

Completeness (GT $\rightarrow$ Prediction):

$$\mathrm{Comp} = \frac{1}{|\mathcal{M}_{\mathrm{GT}}|} \sum_{\boldsymbol{q} \in \mathcal{M}_{\mathrm{GT}}} \min_{\boldsymbol{p} \in \mathcal{P}} \|\boldsymbol{p} - \boldsymbol{q}\|_2$$

Where:

- $\mathcal{P}$: Predicted PC from dense or semi-dense reconstruction or sampled from pred mesh
- $\mathcal{M}_{\mathrm{GT}}$: Sampled points from GT mesh

Precision, Recall & F-score

At threshold τ (typically 5cm):

$$\mathrm{Pr}_{@\tau} = \frac{\left|\left\{\boldsymbol{p} \in \mathcal{P} : \min_{\boldsymbol{q} \in \mathcal{M}_{\mathrm{GT}}} \|\boldsymbol{p} - \boldsymbol{q}\| < \tau\right\}\right|}{|\mathcal{P}|}$$

$$\mathrm{Re}_{@\tau} = \frac{\left|\left\{\boldsymbol{q} \in \mathcal{M}_{\mathrm{GT}} : \min_{\boldsymbol{p} \in \mathcal{P}} \|\boldsymbol{p} - \boldsymbol{q}\| < \tau\right\}\right|}{|\mathcal{M}_{\mathrm{GT}}|}$$

$$\text{F-score@}\tau = \frac{2 \cdot \mathrm{Precision} \cdot \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}$$

Chamfer Distance (Bidirectional)

$$\mathrm{CD}(\mathcal{P}, \mathcal{M}_{\mathrm{GT}}) = \mathrm{Acc} + \mathrm{Comp}$$

Combines both directions of surface error

Reconstruction Quality Metrics

ATEK Implementation:

- `evaluate_single_mesh_pair()`[1] computes all metrics using:
  - [trimesh.Trimesh](): Load meshes + sample surfaces uniformly
  - `compute_pts_to_mesh_dist()`: Point-to-mesh distance via batched triangle projection
  - `point_to_closest_tri_dist()`: Barycentric coordinate projection test + plane distance
  - Fallback: `point_to_closest_vertex_dist()` when projection fails

See [`metrics.qmd`]() for detailed formulas and algorithm explanations.

[1] src

Computing RRI with GT Meshes

RRI from GT Mesh

Given:

- $\mathcal{M}_{\mathrm{GT}}$: GT mesh (from ASE `.ply` files)
- $\mathcal{P}_t$: Current reconstruction from first t views
- $\boldsymbol{q} \in \mathrm{SO}(2) \ltimes \mathbb{R}^3$: Candidate viewpoint, 5DoF (position + yaw, pitch)
- $\mathcal{P}_{t \cup \boldsymbol{q}}$: Updated reconstruction after capturing from $\boldsymbol{q}$

Mesh-based RRI (oracle):

$$\mathrm{RRI}(\boldsymbol{q}) = \frac{\mathrm{CD}(\mathcal{P}_t, \mathcal{M}_{\mathrm{GT}}) - \mathrm{CD}\big(\mathcal{P}_{t \cup \boldsymbol{q}}, \mathcal{M}_{\mathrm{GT}}\big)}{\mathrm{CD}(\mathcal{P}_t, \mathcal{M}_{\mathrm{GT}})}$$

RRI Oracle Pipeline

1. Load GT mesh from ASE
2. Build $\mathcal{P}_t$ from captured views
   - dense PC from depth maps
   - or semi-dense SLAM PC[1]
3. Simulate view from $\boldsymbol{q}$
   - Ray-cast to $\mathcal{M}_{\mathrm{GT}} \to \mathcal{P}_{\boldsymbol{q}}$
4. Merge: $\mathcal{P}_{t \cup \boldsymbol{q}} = \mathcal{P}_t \cup \mathcal{P}_{\boldsymbol{q}}$
   - Voxel downsample for consistency (e.g., 1cm)
5. Compute RRI using CD metric

[1] `semidense_points.csv`

Computing RRI with GT Meshes

Key Functions from EFM3D & ATEK

Point Cloud Generation:

- `dist_im_to_point_cloud_im()`: Depth $\rightarrow$ 3D points
- `collapse_pointcloud_time()`: Merge temporal PCs
- `pointcloud_to_voxel_counts()`: PC $\rightarrow$ density grid

Ray-Mesh Operations:

- `ray_obb_intersection()`: Ray-box intersection
- `sample_depths_in_grid()`: Sample depths along rays

Distance Computation:

- `compute_pts_to_mesh_dist()`: Min distance to triangles
- `eval_mesh_to_mesh()`: Full evaluation pipeline

RRI-based NBV for Scene-Level Reconstruction

VIN with EVL Backbone

Our Approach: Adapt RRI prediction to scene-level environments with 5DoF action space

RRI-based NBV for Scene-Level Reconstruction

RRI with GT Meshes

Use ASE visibility data + GT meshes for *oracle RRI*:

$$\mathrm{RRI}(\boldsymbol{q}) = \frac{d\big(\mathcal{P}_{\text{partial}}, \mathcal{M}_{\text{GT}}\big) - d\big(\mathcal{P}_{\text{partial}} \cup \boldsymbol{q}, \mathcal{M}_{\text{GT}}\big)}{d\big(\mathcal{P}_{\text{partial}}, \mathcal{M}_{\text{GT}}\big)}$$

where $\mathcal{M}$ represents meshes, $d(\cdot,\cdot)$ is mesh distance

Proposed Pipeline

1. **Reconstruct**: Build $\mathcal{P}_{\text{partial}}$ from historical trajectory
2. **Sample**: Generate candidate viewpoints in free space around latest pose
3. **Compute Features**: Extract geometric + semantic embeddings from **EVL**
4. **Predict**: Use **VIN** to predict RRI per candidate
5. **Select**: Choose NBV based on RRI

Key Challenge: Ray-casting from candidate views to compute visibility on GT meshes for entity-wise RRI computation

RRI-based NBV for Scene-Level Reconstruction

Extension Entity-wise RRI:

$$\mathrm{RRI}_{\text{total}} = \sum_{e \in \mathcal{E}} w_e \cdot \mathrm{RRI}_e \text{where } \mathcal{E} = \{\text{walls, doors, objects, ...}\}$$

- This could be done by segmenting the GT meshes and PCs per entity type and computing the RRI separately.

Bibliography

[Ave+24] A. Avetisyan *et al.*, “SceneScript: Reconstructing Scenes With An Autoregressive Structured Language Model.” [Online]. Available: https://arxiv.org/abs/2403.13064

[Met25] Meta Platforms Inc., “Aria Synthetic Environments Dataset.” [Online]. Available: https://facebookresearch.github.io/projectaria_tools/docs/open_datasets/aria_synthetic_environments_dataset

[Str+24] J. Straub, D. DeTone, T. Shen, N. Yang, C. Sweeney, and R. Newcombe, “EFM3D: A Benchmark for Measuring Progress Towards 3D Egocentric Foundation Models.” [Online]. Available: https://arxiv.org/abs/2406.10224

[Met25] Meta Platforms Inc., “Aria Training and Evaluation toolkit (ATEK).” [Online]. Available: https://github.com/facebookresearch/ATEK

[Bra+23] G. Brazil, A. Kumar, J. Straub, N. Ravi, J. Johnson, and G. Gkioxari, “Omni3D: A Large Benchmark and Model for 3D Object Detection in the Wild.” [Online]. Available: https://arxiv.org/abs/2207.10660

Bibliography

[Met25] Meta Platforms Inc., “Aria Training and Evaluation toolkit (ATEK) documentation.” [Online]. Available: https://facebookresearch.github.io/projectaria_tools/docs/ATEK/about_ATEK

[Che+24] X. Chen, Q. Li, T. Wang, T. Xue, and J. Pang, “GenNBV: Generalizable Next-Best-View Policy for Active 3D Reconstruction.” [Online]. Available: https://arxiv.org/abs/2402.16174

[Fra+25] N. Frahm *et al.*, “VIN-NBV: A View Introspection Network for Next-Best-View Selection.” [Online]. Available: https://arxiv.org/abs/2505.06219