2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

ARTEMIS: Active Real-time Textured Environment Meshing with Interactive Semantics

Yigu Ge[1], Zhenhuan Ma[1], Shihao Tang[1], Yangxi Shi[1], Xinkai Liang[1] and Hao Fang[1,2]

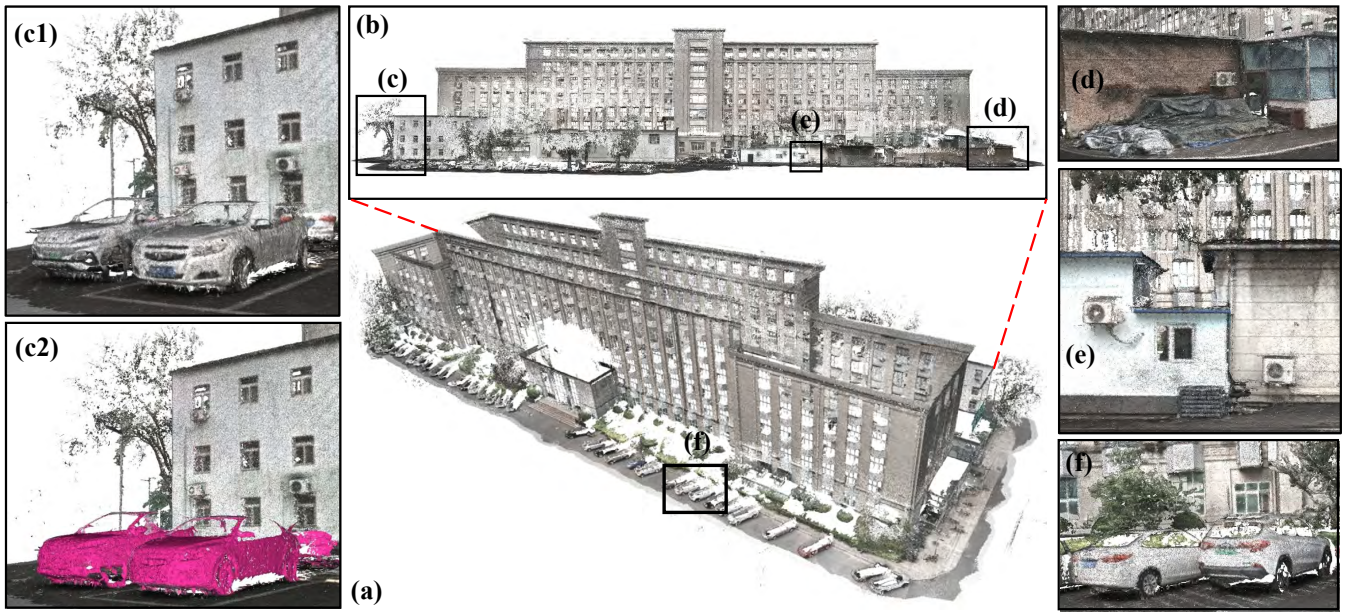

Fig. 1. Reconstruction results of our approach for large-scale environments. (a) shows the reconstruction results of our self-collected campus building dataset. (b) is the rear view of the building. (d)–(f) show the details of our reconstruction, including accurate geometric structures and photorealistic textures. (c1) and (c2) demonstrate the effect of our Semantic Brush, which accurately highlights the car category.

***Abstract*— To advance 3D reconstruction from static digital replicas towards semantically interactive Living Maps responsive to an agent's queries, we propose ARTEMIS, a system for Active Real-time Textured Environment Meshing with Interactive Semantics. At its core, our Semantic Brush is a methodology comprised of tightly-coupled modules for segmentation, constraint, and refinement that operate in a two-stage, coarse-to-fine pipeline. Initially, its segmentation and constraint modules translate natural language into a semantically-aware mesh, enforcing sharp object boundaries with a unified energy function. Subsequently, its refinement module computes a unified reliability metric from color and depth consistency to guide the joint optimization of the texture map and semantic labels. This holistic process inherently filters unreliable measurements, establishing a complete interactive workflow from language input to real-time highlighting on a high-fidelity textured mesh. We evaluated ARTEMIS on public datasets and in real-world scenarios. The results demonstrate its state-of-the-art accuracy in mesh reconstruction, while simultaneously attaining high fidelity in both texture and semantics. To share our findings and make contributions to the community, our code will be made publicly available.**

*This work was supported in part by the National Nature Science Foundation of China (NSFC) under Grant (No.62133002).

[1]All authors are with School of Automation, Beijing Institute of Technology. Yigu Ge (bengay@bit.edu.cn), Shihao Tang (shihaotang@bit.edu.cn).

[2]The corresponding author: Hao Fang (fangh@bit.edu.cn).

I. INTRODUCTION

In the evolution towards embodied and general artificial intelligence [1], traditional maps, being mere geometric replicas, no longer suffice for the autonomous interaction needs of intelligent agents. To empower the next generation of AI, the concept of a Living Map, which transcends static representations, becomes critically important. The realization of such a map relies on the seamless, real-time unification of three core attributes. First, it requires a precisely structured geometric mesh that can represent continuous surfaces with accurate topology, thus providing a meaningful scaffold for interaction. Second, it demands rich, interactive semantics, enabling an agent to query and understand the environment through high-level instructions like natural language. Third, it must possess a photorealistic, high-fidelity texture to create a visually immersive digital twin. Furthermore, the simultaneous generation and fusion of these components in real-time is a fundamental requirement, as it ensures the map remains a current and responsive representation of the physical world.

However, existing methods often compromise among these critical attributes, precluding the realization of a true Living Map. Purely geometric reconstruction frameworks [2], [3] can produce highly accurate meshes but lack the capacity to capture essential semantic and textural information. Conversely, semantic SLAM systems [4], [5] integrate rich

semantic labels but often at the expense of texture fidelity, real-time interactivity, and geometric precision. Furthermore, RGB-colored LiDAR-Inertial-Visual SLAM systems [6], [7], while visually compelling, represent the world as a dense point cloud. Their discrete representation fails to form a continuous surface, which is fundamental for meaningful rendering and interaction. Therefore, developing a framework that simultaneously integrates precise geometry, interactive semantics, and high-fidelity texture within a real-time mesh representation remains a significant open challenge.

To address this challenge, we propose ARTEMIS, an **A**ctive **R**eal-time **T**extured **E**nvironment **M**eshing system with **I**nteractive **S**emantics that successfully instantiates the Living Map concept through our core methodology, the Semantic Brush. This methodology is comprised of tightly-coupled modules that holistically unify the four critical requirements. To achieve precise geometry, its constraint module refines the mesh using a novel, semantically-guided energy function that aligns structural boundaries with object distinctions. To enable interactive semantics, its segmentation module translates natural language queries into 3D semantic masks that are fused and refined throughout the pipeline. For high-fidelity texture, its refinement module employs a parallel optimization process, governed by a unified reliability metric, to ensure photorealistic mapping. Crucially, the entire methodology is engineered for real-time performance, with its modules operating efficiently on active-only mesh regions and employing view frustum culling to focus computation.

The main contributions of our work are as follows:

- We propose ARTEMIS, a complete system that implements our novel Semantic Brush methodology. To the best of our knowledge, ARTEMIS is the first framework among real-time LiDAR-visual-inertial fusion based SLAM systems to establish a direct workflow from a natural language query to the highlighting of a target object.
- We design a semantically-guided mesh optimization framework, where a unified energy function leverages semantic priors to significantly improve geometric accuracy and preserve sharp object boundaries.
- We evaluate ARTEMIS on public datasets and real-world sequences, showing state-of-the-art geometric accuracy with superior texture fidelity and semantic correctness.

II. RELATED WORK

A. Online 3D Reconstruction with Point Cloud

Early works like KinectFusion [8] popularized TSDF-based dense reconstruction, with subsequent efforts improving scalability [9], [10], multi-resolution support [11], [12], and efficiency [13], [14], [15]. Another mainstream technical route employs points or surfels as a discrete representation of the scene [16], [17]. More directly related to our work are online frameworks such as ImMesh [2] and SLAMesh [3], which explicitly reconstruct the scene as a continuous triangular mesh. However, being purely geometric, they often produce erroneous connections and topological artifacts, particularly when reconstructing structurally similar or proximate objects. More importantly, they lack the ability to capture semantic and textural information, resulting in geometrically accurate but otherwise sterile digital replicas. In contrast, ARTEMIS leverages semantic information as a structural prior to resolve the ambiguities inherent in purely geometric approaches, yielding a map that is not only geometrically precise but also semantically rich and photorealistic.

B. Semantic SLAM

Endowing 3D maps with semantics is critical for scene understanding and has been an active area of research. Early systems like SLAM++ [18] operated at the object level. Subsequent works such as SemanticFusion [19] and MaskFusion [20] advanced this by fusing 2D semantic segmentation into 3D representations. While Kimera [5] integrates metric-semantic mapping, its reliance on VIO can limit its robustness in large-scale, texture-scarce environments. For LiDAR-based systems, methods such as SuMa++ [4], Sdf-slam [21], and the recent SGS-SLAM [22] have successfully combined semantic information with representations like surfels, SDFs, or Gaussian splatting. Although these systems produce richly annotated maps, their semantic generation is passive, creating a static output that precludes real-time interaction based on an agent's intent. Our Semantic Brush methodology directly addresses this gap by transforming static label generation into a dynamic workflow that extends from a natural language query to a target highlighted in real-time.

C. Real-time LiDAR-Visual-Inertial Fusion Based SLAM Systems

Fusing multimodal sensor data is crucial for the construction of photorealistic maps. State-of-the-art systems such as R3LIVE [7], FAST LIVO2 [6] and LVI-Fusion [23] generate dense, colored maps in real-time through robust state estimation. These methods produce visually compelling results by projecting RGB camera data onto LiDAR point clouds. However, their final output is fundamentally a discrete RGB point cloud. While appealing from a distance, the inherent sparsity of this representation prevents the formation of a continuous surface, which is essential for fine-grained interaction and high-fidelity rendering applications. In contrast, ARTEMIS incrementally constructs and maintains a continuous, high-fidelity textured mesh. This continuous representation not only provides a geometrically sound scaffold for photorealistic texturing but also serves as the essential foundation for the precise and effective deployment of our Semantic Brush.

III. APPROACH

A. System Overview

As shown in Figure 2, the core of our approach is the Semantic Brush, a methodology that integrates semantic understanding throughout the entire workflow via four key modules. The first stage begins as the Semantic Segmentation module translates natural language queries into 2D masks, which are fused with LiDAR geometry at the voxel level.

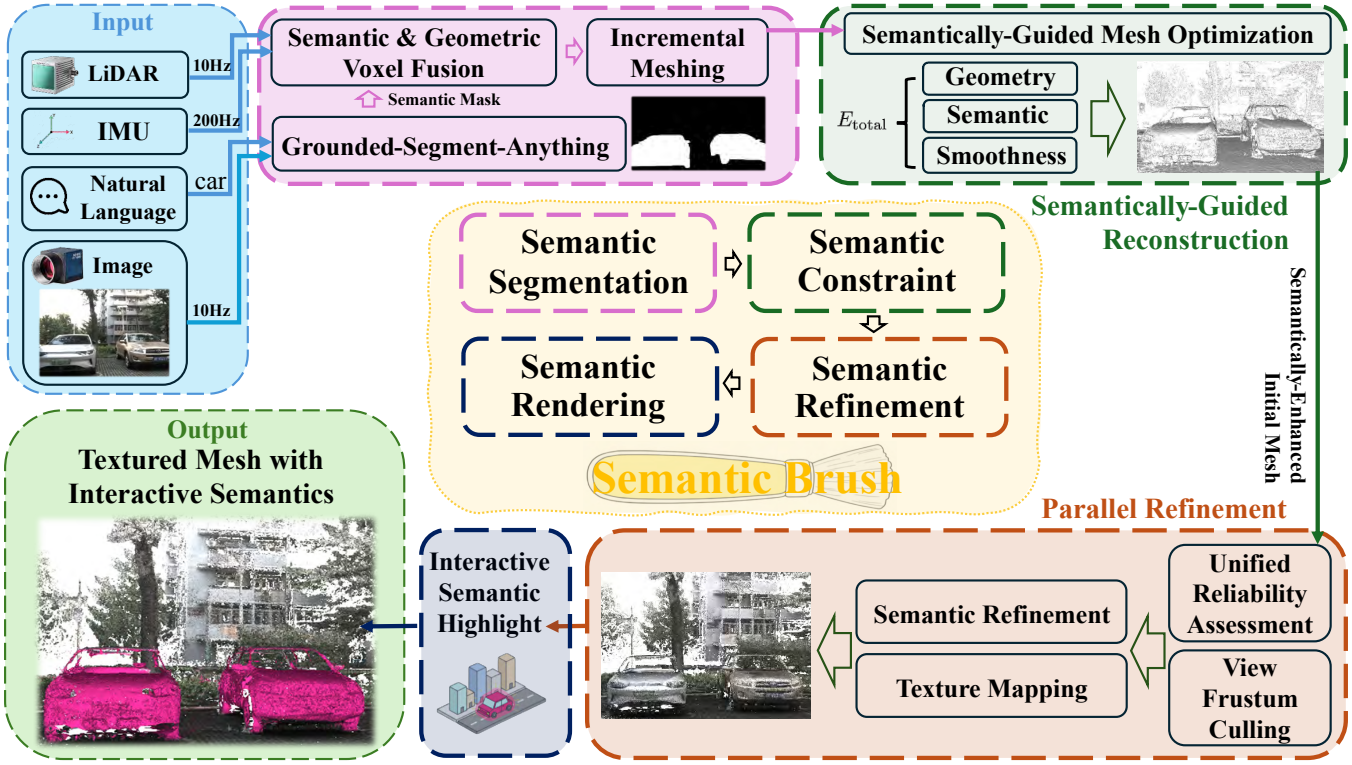

Fig. 2. The system overview of our proposed ARTEMIS. Central to our framework, the four modules of the Semantic Brush methodology coordinate and regulate the execution of the outer coarse-to-fine, two-stage reconstruction pipeline.

An incremental meshing procedure, which only updates active voxels for efficiency, then generates an initial semantic mesh. Finally, the Semantic Constraint module optimizes this mesh using a unified energy function to align its geometric boundaries with semantic distinctions. The second stage, Parallel Refinement, then enhances the mesh's attributes. To focus computation, this stage begins with view frustum culling on the mesh. Governed by a unified reliability metric, which is computed at the beginning of this stage, the Semantic Refinement module then operates in parallel with a texture mapping process to co-optimize per-vertex semantics and high-fidelity textures. The final output is a photorealistic, textured mesh, where the Semantic Rendering module uses the refined data to highlight target objects on the fly in response to queries, enabling real-time interaction.

B. Semantically-Guided Reconstruction

This stage aims to construct an initial, semantically-enhanced mesh through a three-step process.

As a new LiDAR scan arrives, its points are allocated to their corresponding voxels in a 3D grid. Concurrently, we leverage the zero-shot capabilities of Grounded-Segment-Anything (G-SAM) [24] to generate 2D semantic masks from natural language queries on synchronized images. To ensure real-time performance, we employ the FastSAM model [25] for this task. These masks are then fused into the voxel grid. Each voxel maintains a semantic probability distribution, which is updated via confidence-weighted averaging. The confidence of a single observation, $\omega_{\text{mask}}^{(t)}$, is defined as:

$$\omega_{\text{mask}}^{(t)} = \sigma_{\text{mask}} \cdot \cos(\theta_{\text{view}}) \cdot \exp\left(-\frac{d_{\text{depth}}}{\sigma_d}\right) \quad (1)$$

where σ_{mask} is the segmentation network's score, θ_{view} is the viewing angle, d_{depth} is the depth uncertainty, and σ_d is a scaling factor for the depth uncertainty. This ensures that geometry and semantics are tightly coupled at the most fundamental data level—the voxel.

To achieve real-time performance, we employ an incremental meshing strategy where only voxels that have received new LiDAR points are activated for updates. Within an active voxel, the system fits a local plane, projects the contained points onto it, and performs a 2D Delaunay triangulation. The resulting connectivity is then lifted back into 3D space to form new mesh facets. Critically, as new vertices are created, each one directly inherits the semantic probability distribution from its parent voxel. This process generates an initial mesh that is inherently a semantic mesh, carrying preliminary semantic labels from the moment of its creation.

The generated semantic mesh serves as input to an optimization module that refines vertex positions by minimizing a unified energy function, E_{total}. This optimization aligns the geometric structure with semantic boundaries. The function is defined as:

$$E_{\text{total}} = E_{\text{geometry}} + \lambda_{\text{sem}} E_{\text{semantic}} + \lambda_{\text{smooth}} E_{\text{smoothness}} \quad (2)$$

where E_{geometry} is a standard point-to-plane metric. Our innovation lies in the latter two terms. The semantic consistency term, E_{semantic}, penalizes connections between vertices with dissimilar semantic distributions:

$$E_{\text{semantic}} = \sum_{(v_i,v_j)\in E} w_{ij}^{\text{sem}} \, \text{JS}\left(P_{\text{sem}}(v_i) \| P_{\text{sem}}(v_j)\right) \quad (3)$$

The sum is over all mesh edges E. We use the Jensen-Shannon (JS) divergence to measure the difference between vertex semantic distributions $P_{\text{sem}}(v_i)$ and $P_{\text{sem}}(v_j)$. The smoothness term, $E_{\text{smoothness}}$, encourages coplanarity but is modulated by semantic similarity:

$$E_{\text{smoothness}} = \sum_{v_i} \sum_{v_j \in \mathcal{N}(v_i)} w_{ij}^{\text{smooth}} \|\mathbf{n}_i - \mathbf{n}_j\|_2^2 \quad (4)$$

where for each vertex v_i and its neighbors $\mathcal{N}(v_i)$, n_i and n_j are their respective surface normals. The weights are designed to be semantically aware. The semantic similarity $\text{sim}_{\text{semantic}}(v_i, v_j)$ is derived from the JS divergence and a temperature parameter τ:

$$\text{sim}_{\text{semantic}}(v_i, v_j) = \exp\left(-\tau \sqrt{\text{JS}(P_{\text{sem}}(v_i) \| P_{\text{sem}}(v_j))}\right) \quad (5)$$

The semantic weight w_{ij}^{sem} is a function of vertex semantic confidence ξ_{sem} and edge length L_{ij}:

$$w_{ij}^{\text{sem}} = \frac{\xi_{\text{sem}}(v_i, t)\xi_{\text{sem}}(v_j, t)}{L_{ij}} \quad (6)$$

The vertex semantic confidence $\xi_{\text{sem}}(v_i, t)$ at the current time t is derived from the accumulated observation weight $W_{\text{sem}}^{(t)}$ of its parent voxel V_i:

$$\xi_{\text{sem}}(v_i, t) = 1 - \exp\left(-\lambda W_{\text{sem}}^{(t)}(V_i)\right), \quad \lambda > 0 \quad (7)$$

Finally, the smoothness weight couples geometry with semantics using a geometric weight w_{ij}^{geo}(e.g.,inverse edge length):

$$w_{ij}^{\text{smooth}} = \text{sim}_{\text{semantic}}(v_i, v_j) \cdot w_{ij}^{\text{geo}} \quad (8)$$

This formulation ensures that the smoothing constraint is weakened across semantic boundaries, preserving sharp object edges. The energy function is optimized using an iterative non-linear Gauss-Seidel approach, yielding a geometrically precise mesh with initial semantic labels.

C. Parallel Refinement via a Unified Reliability Metric

After the initial mesh is constructed, its attributes are refined in parallel, governed by a unified metric that ensures data quality.

As an acceleration strategy, the system first performs view frustum culling, ensuring that only vertices visible within the current camera frame are considered for refinement. For each visible vertex, we then assess the quality of its 3D-to-2D projection by computing a unified reliability metric, W. This metric robustly penalizes both photometric and geometric inconsistencies:

$$W = \exp\left(-\frac{1}{2}\left(\left(\frac{r_{\text{photo}}}{\sigma_{\text{photo}}}\right)^2 + \left(\frac{r_{\text{geom}}}{\sigma_{\text{geom}}}\right)^2\right)\right) \quad (9)$$

where σ_{photo} and σ_{geom} are parameters that control the sensitivity to photometric and geometric noise, respectively. The photometric residual r_{photo} and geometric residual r_{geom} are defined as:

$$r_{\text{photo}} = \left\|\mathbf{c}(u, v) - \mu_i^-\right\|_2 \quad (10)$$

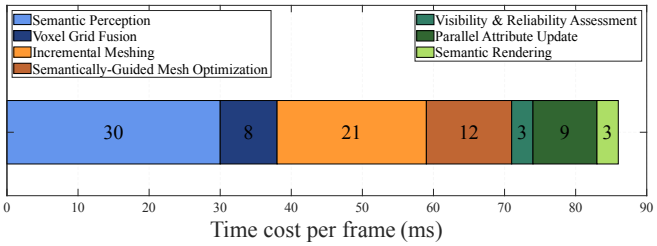

Fig. 3. Per-stage time cost analysis for a single-frame reconstruction in ARTEMIS, showing a total average processing time of approximately 86ms, which meets the real-time requirements of standard sensors like a 10Hz LiDAR.

$$r_{\text{geom}} = \frac{|d_{\text{proj}} - d_{\text{ref}}|}{\max(d_{\text{ref}}, \epsilon)} \quad (11)$$

where $c(u, v)$ is the color of the pixel at the projected coordinates (u, v), and μ_i^- is the current mean color of the vertex before the update. For the geometric residual, d_{proj} is the projected depth of the vertex, d_{ref} is the depth from the reference depth map, and ϵ is a small constant to prevent division by zero. Only vertices corresponding to highly reliable observations (i.e., a high W value) are passed to the next step.

Guided by the reliability metric W, the system performs parallel Bayesian updates for both texture and semantics. For texture mapping, each vertex maintains a Gaussian color state (μ, Σ), which is updated within a Bayesian framework modulated by W. In parallel, the per-vertex semantic distribution is enhanced via a recursive Bayesian update in the log-probability domain:

$$\lambda_{i,c}^{(k)} = \lambda_{i,c}^{(k-1)} + \log p\left(o_i^{(k)} | c\right) \quad (12)$$

where $\lambda_{i,c}^{(k)}$ is the unnormalized log-probability of vertex i belonging to class c after the observation $o_i^{(k)}$ in frame k. The key to our unified approach is that the formulation of the log-likelihood is modulated by the exact same reliability metric W:

$$\log p\left(o_i^{(k)} | c\right) = W_i^{(k)} \cdot \log P_{\text{mask}}\left(o_i^{(k)} | c\right) \quad (13)$$

where $W_i^{(k)}$ is the reliability for vertex i in frame k, and $P_{\text{mask}}(o_i^{(k)}|c)$ is the probability of the observation belonging to class c according to the 2D segmentation mask. This formulation interprets W as a parameter that controls the sharpness of the likelihood distribution. A highly reliable observation ($W \to 1$) yields a confident update, while an unreliable one ($W \to 0$) results in a near-uniform likelihood. This direct reuse of W creates a tight, principled coupling, ensuring that information from an unreliable viewpoint is consistently discounted when updating both texture and semantics. Finally, the refined probability vector is recovered via the softmax function and used by the Semantic Rendering Module to highlight target objects in the final render. This enables the direct highlighting of target object instances on the final textured mesh in response to the initial natural language queries.

TABLE I

QUANTITATIVE COMPARISON OF RECONSTRUCTION QUALITY AND PERFORMANCE ON KITTI SEQUENCES 01, 04, AND 07. WE REPORT ACCURACY (PRECISION, RECALL, F1-SCORE) AND AVERAGE PER-FRAME PROCESSING TIME. ↑ DENOTES LARGER IS BETTER, WHILE ↓ INDICATES LOWER IS BETTER.

Sequence	Scenarios	LiDAR frames	Method	Semantics	Texture	Interactivity	FrameTime Avg (ms) ↓	Correctness		
								Precision ↑	Recall ↑	F1-score ↑
Kitti_01	High way	1101	SLAMmesh	—	—	—	**24.8**	0.7262	0.7541	0.7399
			ImMesh	—	—	—	34.5	0.7717	0.8156	0.7930
			Poi	—	—	—	4028.7	0.7821	0.8354	0.8097
			Ours	✓	✓	✓	85.1	**0.8113**	**0.8613**	**0.8356**
Kitti_04	Urban city; Road	271	SLAMmesh	—	—	—	**25.7**	0.7532	0.7746	0.7638
			ImMesh	—	—	—	30.1	0.8156	0.8439	0.8295
			Poi	—	—	—	3379.5	0.8436	0.8761	0.8593
			Ours	✓	✓	✓	82.3	**0.8514**	**0.8976**	**0.8739**
Kitti_07	Urban city	1101	SLAMmesh	—	—	—	21.4	0.7496	0.7636	0.7565
			ImMesh	—	—	—	**20.7**	0.8028	0.8492	0.8254
			Poi	—	—	—	2386.2	0.8513	0.8829	0.8668
			Ours	✓	✓	✓	78.6	**0.8651**	**0.9076**	**0.8858**

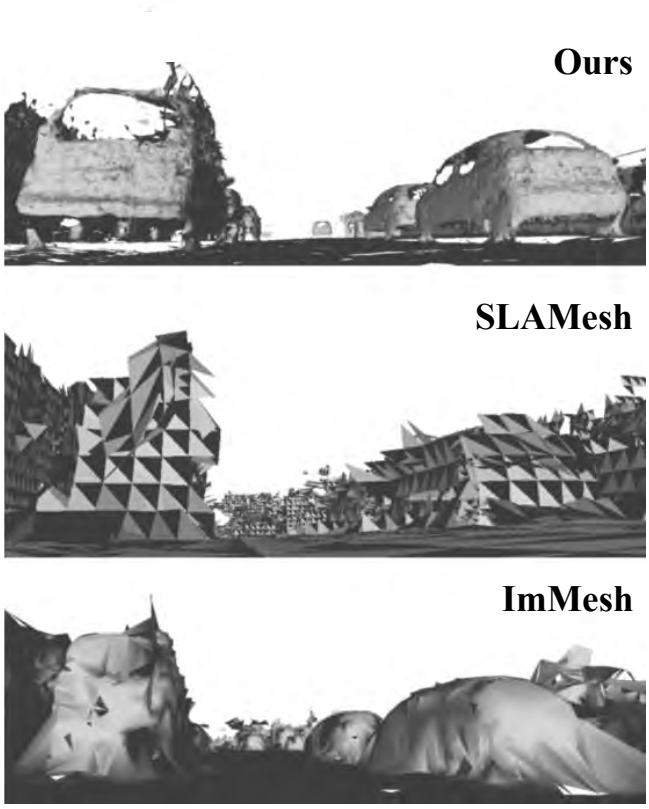

Fig. 4. Qualitative comparison of geometric reconstruction on Kitti sequence 07. From top to bottom: our method (ARTEMIS), SLAMesh, and ImMesh. Our method's result demonstrates better structural preservation, particularly in distinguishing the vehicle from the ground.

IV. EXPERIMENTS AND RESULTS

This section first introduces the experimental setup, including the datasets, comparison methods, and hardware platform. Subsequently, we compare ARTEMIS with state-of-the-art methods across three core metrics: geometric reconstruction quality, texture mapping quality, and semantic segmentation accuracy with interactive functionality. Finally, we validate the efficacy of the key components of our system through ablation studies.

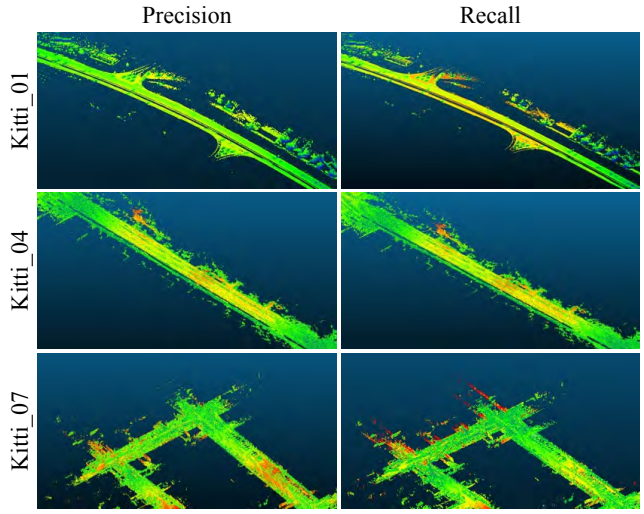

Fig. 5. Mesh maps compared with the ground-truth on the Kitti dataset. The left and right columns illustrate Precision and Recall values, respectively. Darker colors indicate worse performance and vice versa. Our ARTEMIS achieves high values for both Recall and Precision.

A. Experimental Setup

Our evaluation utilized public datasets and real-world sequences to ensure a thorough assessment. For geometric accuracy, we used sequences 01, 04, and 07 from the Kitti Odometry Benchmark [26]. For semantic accuracy, we benchmarked against sequence 07 of the SemanticKitti dataset [27]. To evaluate texture quality and demonstrate generalization, we used public sequences from FAST-LIVO2 dataset [6] and data from a large-scale campus environment captured with our custom handheld device. This device, shown in Fig. 7(f), is equipped with a Livox Avia LiDAR and a global shutter RGB camera. Our system was benchmarked against state-of-the-art methods, including ImMesh [2], SLAMesh [3], and Poisson surface reconstruction(Poi) [28] for geometry; R3LIVE [7] and FAST-LIVO2 [6] for texture; and SuMa++ [4] for semantics.

All experiments are conducted on a PC with an Intel Core i7-11800H CPU and an NVIDIA GeForce RTX 4070 GPU. Baselines were executed using their publicly available

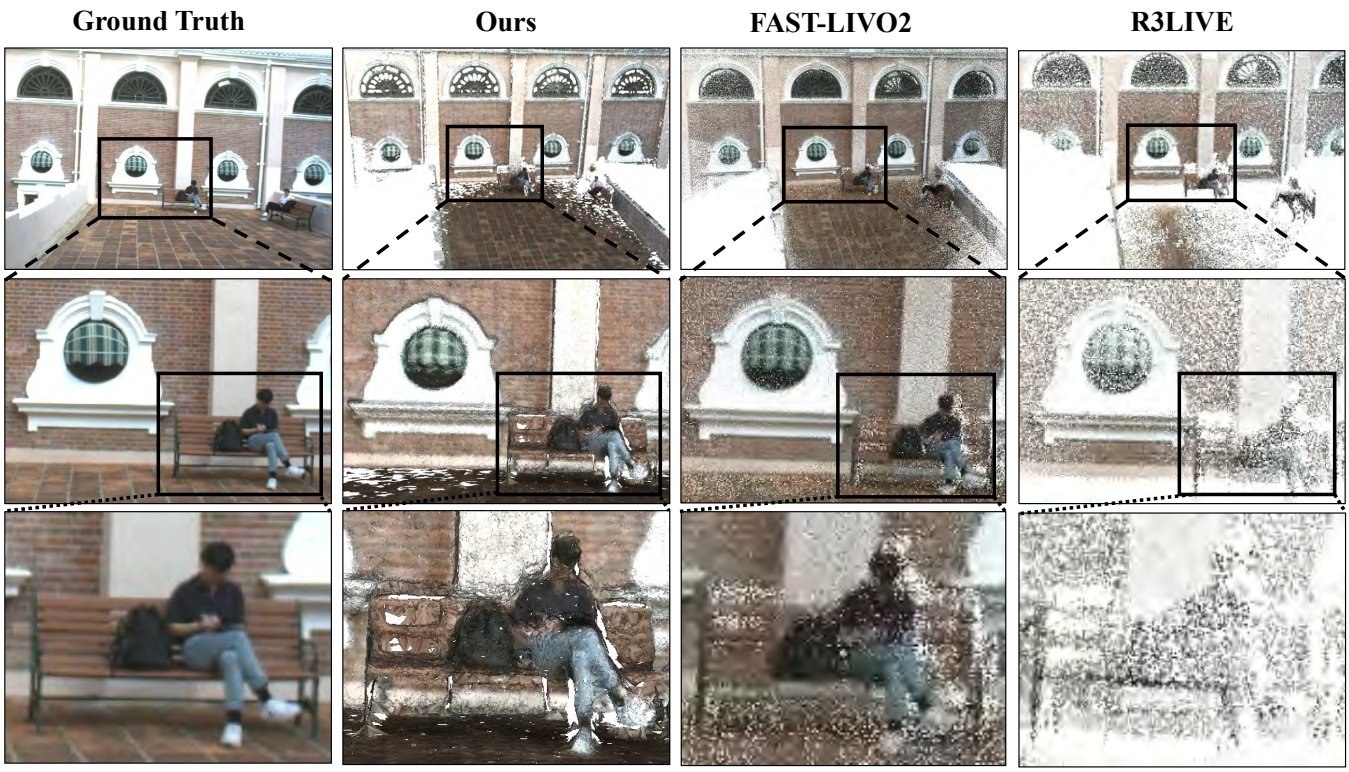

Fig. 6. Qualitative comparison of texture mapping quality on a challenging real-world sequence from the FAST-LIVO2 public dataset. From left to right: Ground Truth, ARTEMIS (Ours), FAST-LIVO2, and R3LIVE. The second and third rows provide zoomed-in views, which highlight that our continuous mesh-based texturing preserves sharpness and avoids the disintegration seen in point cloud representations.

implementations with default configurations to ensure a fair comparison.

B. Geometric Reconstruction Quality

The results in Table I show that ARTEMIS achieves a higher reconstruction quality, outperforming both real-time geometric methods (ImMesh, SLAMesh) and the high-fidelity offline method (Poisson Reconstruction) across all accuracy metrics. This superior performance comes at a modest computational cost. While our system is slower than purely geometric approaches, the increased processing time enables the integration of rich semantics and photorealistic textures—features that those methods lack. Crucially, ARTEMIS remains well within real-time operational limits. As shown in Table I and the per-stage breakdown in Fig. 3, our average processing time of $\sim$86ms is orders of magnitude faster than Poisson reconstruction and meets the real-time requirements of standard 10Hz LiDAR sensors, thus achieving a high-quality reconstruction complete with semantics and photorealistic textures, all while maintaining real-time performance.

Fig. 4 provides a visual comparison of the reconstruction results between ARTEMIS and the baseline methods on Kitti sequence 07. By leveraging semantic information as a structural prior, our method generates a more detailed and physically plausible mesh when reconstructing distinct objects like vehicles. As shown in the figure, ARTEMIS not only preserves high-frequency details such as window frames and sharp silhouettes but also maintains clear topological separation, correctly distinguishing the vehicle's chassis from the ground plane and avoiding the artifacts of adhesion and distortion present in SLAMesh and ImMesh. The precision and recall heatmaps in Fig. 5 further validate the high completeness and accuracy of our method from a global perspective.

C. Texture Mapping Quality

In Fig. 6, we compare our texturing approach against R3LIVE and FAST-LIVO2, which are state-of-the-art systems in the field of real-time LiDAR-Visual-Inertial fusion based SLAM that produce excellent photorealistic coloring results. While their point cloud coloring can be visually appealing from a distance, its inherent sparsity causes the scene to disintegrate upon closer inspection, failing to provide an immersive close-up experience. In contrast, ARTEMIS overcomes this fundamental limitation by performing sub-pixel color fusion on a continuous triangular mesh surface. This allows our model to maintain a high degree of visual realism and surface integrity at any viewing distance. The sharp details visible on both the brick wall and the person demonstrate the superiority of our method in texturing both large planar surfaces and complex non-rigid objects.

Furthermore, Fig. 1 showcases our system's capability in a large-scale, self-collected campus scene. From the overall building structure (Fig. 1 a, b) to the fine textures of walls, air conditioning units, and vegetation (Fig. 1 d–f), ARTEMIS demonstrates exceptional performance in generating highly detailed, photorealistic textures that support full-scale observation.

TABLE II

QUANTITATIVE COMPARISON OF SEMANTIC SEGMENTATION ACCURACY (MIOU %) ON SEMANTICKITTI SEQUENCE 07.

Category	Scenarios	Method	mIoU(%)
car	Urban city	SuMa++	91.4
		ARTEMIS(Ours)	**94.5**
building	Urban city	SuMa++	86.3
		ARTEMIS(Ours)	**92.6**
vegetation	Urban city	SuMa++	80.6
		ARTEMIS(Ours)	**87.2**

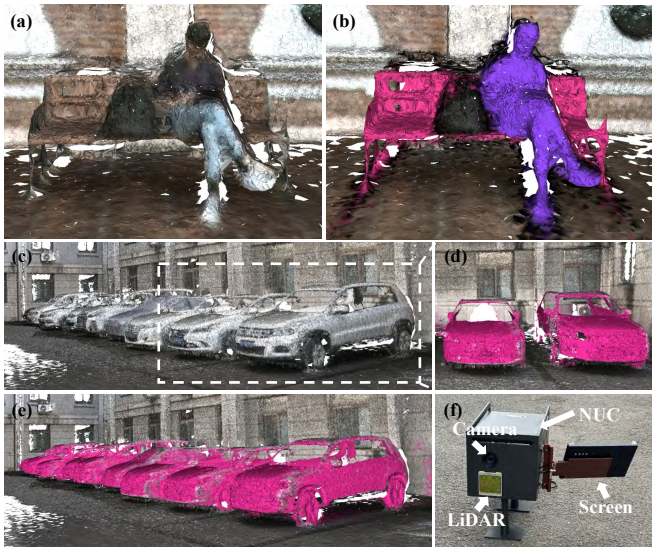

Fig. 7. Demonstration of the Semantic Brush functionality. (a)–(b) Highlighting multiple categories (person and bench) simultaneously on the FAST-LIVO2 dataset. The method's high precision is evident as the backpack placed on the bench is correctly excluded from the highlight, resulting in exceptionally clear boundaries. (c)–(e) Highlighting the car category on our self-collected dataset, showing generalization across categories and datasets. (f) Our handheld data collection device.

D. Semantic Accuracy and Interactive Functionality

As demonstrated in Fig. 7, our system's interactive functionality is both precise and generalizable. It accurately identifies and highlights multiple instances of a single category, such as car, in our large-scale dataset (Fig. 7(c)–(e)). To further showcase its advanced capabilities, we demonstrate its ability to process multiple queries simultaneously. In the FAST-LIVO2 dataset (Fig. 7(a)–(b)), the system successfully segments and highlights two distinct categories, the person and the bench, within the same view. This combination is particularly challenging due to the non-rigid, articulated nature of the person and the close proximity and partial occlusion of the bench. Crucially, this scene also includes a backpack placed by the person on the bench. Our Semantic Brush demonstrates exceptional precision by flawlessly highlighting only the person and the bench, while correctly excluding the adjacent backpack from the semantic mask. This ability to maintain sharp, accurate boundaries between distinct object instances, even when they are in direct contact, powerfully validates the effectiveness of our methodology. This result highlights a robust semantic understanding that extends across different object classes and environments, proving its utility in complex, multi-object scenes.

TABLE III

ABLATION STUDY OF THE SEMANTIC CONSTRAINT MODULE ON THE KITTI DATASET.

Sequence	Variant	Correctness		
		Precision ↑	Recall ↑	F1-score ↑
Kitti_01	w/o Sem. Constraint	0.7514	0.8134	0.7812
	ARTEMIS	**0.8113**	**0.8613**	**0.8356**
Kitti_04	w/o Sem. Constraint	0.7947	0.8441	0.8187
	ARTEMIS	**0.8514**	**0.8976**	**0.8739**
Kitti_07	w/o Sem. Constraint	0.8179	0.8534	0.8353
	ARTEMIS	**0.8651**	**0.9076**	**0.8858**

TABLE IV

ABLATION STUDY OF THE SEMANTIC REFINEMENT MODULE (MIOU %) ON SEMANTICKITTI SEQUENCE 07.

Category	Variant	mIoU(car)(%)
car	ARTEMIS w/o Sem. Refinement	89.7
	ARTEMIS	**94.5**
building	ARTEMIS w/o Sem. Refinement	86.1
	ARTEMIS	**92.6**
vegetation	ARTEMIS w/o Sem. Refinement	82.4
	ARTEMIS	**87.2**

E. Ablation Study

To validate the efficacy of the key designs in our system, we conducted a series of ablation studies. We first evaluated the guiding role of semantic information in geometric reconstruction. As shown in Table III, after removing the Semantic Constraint Module, the F1-score for geometric reconstruction drops significantly across all test sequences. This result directly demonstrates that our semantic constraints are crucial for improving the accuracy and completeness of the reconstruction.

Next, we verified the necessity of the Semantic Refinement Module. In this variant, the module was disabled, and semantic labels were derived solely from the initial probabilities of the voxel fusion stage. As presented in Table IV, without the parallel Bayesian updates, the semantic segmentation mIoU decreases substantially across all tested categories. This indicates that our proposed Semantic Refinement Module is indispensable for achieving high-precision semantic maps.

V. CONCLUSIONS

In this paper, we introduced ARTEMIS, a system for active real-time textured environment meshing with interactive semantics. Our core contribution, the Semantic Brush, holistically integrates high-fidelity geometry, photorealistic textures, and interactive semantics within a single, real-time framework. Extensive experiments validate that ARTEMIS achieves state of-the-art geometric accuracy alongside superior performance in texture fidelity and semantic correctness. Future efforts will address current limitations by incorporating semantics based filtering of dynamic objects and expanding the system's perceptual range with a 360° LiDAR and a multi-camera setup.

REFERENCES

[1] Sonia Raychaudhuri and Angel X Chang. Semantic mapping in indoor embodied ai–a survey on advances, challenges, and future directions. *arXiv preprint arXiv:2501.05750*, 2025.

[2] Jiarong Lin, Chongjian Yuan, Yixi Cai, Haotian Li, Yunfan Ren, Yuying Zou, Xiaoping Hong, and Fu Zhang. Immesh: An immediate lidar localization and meshing framework. *IEEE Transactions on Robotics*, 39(6):4312–4331, 2023.

[3] Jianyuan Ruan, Bo Li, Yibo Wang, and Yuxiang Sun. Slamesh: Real-time lidar simultaneous localization and meshing. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3546–3552, 2023.

[4] Xieyuanli Chen, Andres Milioto, Emanuele Palazzolo, Philippe Giguère, Jens Behley, and Cyrill Stachniss. Suma++: Efficient lidar-based semantic slam. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4530–4537, 2019.

[5] Antoni Rosinol, Marcus Abate, Yun Chang, and Luca Carlone. Kimera: an open-source library for real-time metric-semantic localization and mapping. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1689–1696, 2020.

[6] Chunran Zheng, Wei Xu, Zuhao Zou, Tong Hua, Chongjian Yuan, Dongjiao He, Bingyang Zhou, Zheng Liu, Jiarong Lin, Fangcheng Zhu, Yunfan Ren, Rong Wang, Fanle Meng, and Fu Zhang. Fast-livo2: Fast, direct lidar–inertial–visual odometry. *IEEE Transactions on Robotics*, 41:326–346, 2025.

[7] Jiarong Lin and Fu Zhang. R3live: A robust, real-time, rgb-colored, lidar-inertial-visual tightly-coupled state estimation and mapping package. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 10672–10678, 2022.

[8] Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pages 127–136, 2011.

[9] Jiawen Chen, Dennis Bautembach, and Shahram Izadi. Scalable real-time volumetric surface reconstruction. *ACM Trans. Graph.*, 32(4), 2013.

[10] Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Marc Stamminger. Real-time 3d reconstruction at scale using voxel hashing. *ACM Trans. Graph.*, 32(6), 2013.

[11] Olaf Kähler, Victor Prisacariu, Julien Valentin, and David Murray. Hierarchical voxel block hashing for efficient integration of depth images. *IEEE Robotics and Automation Letters*, 1(1):192–197, 2016.

[12] Emanuele Vespa, Nikolay Nikolov, Marius Grimm, Luigi Nardi, Paul H. J. Kelly, and Stefan Leutenegger. Efficient octree-based volumetric slam supporting signed-distance and occupancy mapping. *IEEE Robotics and Automation Letters*, 3(2):1144–1151, 2018.

[13] Olaf Kähler, Victor Adrian Prisacariu, Carl Yuheng Ren, Xin Sun, Philip Torr, and David Murray. Very high frame rate volumetric integration of depth images on mobile devices. *IEEE Transactions on Visualization and Computer Graphics*, 21(11):1241–1250, 2015.

[14] Matthew Klingensmith, Ivan Dryanovski, Siddhartha Srinivasa, and Jizhong Xiao. Chisel: Real time large scale 3d reconstruction onboard a mobile device using spatially hashed signed distance fields. In *Proceedings of Robotics: Science and Systems*, 2015.

[15] Helen Oleynikova, Zachary Taylor, Marius Fehr, Roland Siegwart, and Juan Nieto. Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1366–1373, 2017.

[16] Damien Lefloch, Markus Kluge, Hamed Sarbolandi, Tim Weyrich, and Andreas Kolb. Comprehensive use of curvature for robust and accurate online surface reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2349–2365, 2017.

[17] Damien Lefloch, Tim Weyrich, and Andreas Kolb. Anisotropic point-based fusion. In *2015 18th International Conference on Information Fusion (Fusion)*, pages 2121–2128, 2015.

[18] Renato F. Salas-Moreno, Richard A. Newcombe, Hauke Strasdat, Paul H.J. Kelly, and Andrew J. Davison. Slam++: Simultaneous localisation and mapping at the level of objects. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1352–1359, 2013.

[19] John McCormac, Ankur Handa, Andrew Davison, and Stefan Leutenegger. Semanticfusion: Dense 3d semantic mapping with convolutional neural networks. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4628–4635, 2017.

[20] Martin Runz, Maud Buffier, and Lourdes Agapito. Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects. In *2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 10–20, 2018.

[21] Linyan Cui and Chaowei Ma. Sdf-slam: Semantic depth filter slam for dynamic environments. *IEEE Access*, 8:95301–95311, 2020.

[22] Mingrui Li, Shuhong Liu, Heng Zhou, Guohao Zhu, Na Cheng, Tianchen Deng, and Hongyu Wang. Sgs-slam: Semantic gaussian splatting for neural dense slam. In *Computer Vision – ECCV 2024*, pages 163–179, 2025.

[23] Zhenbin Liu, Zengke Li, Ao Liu, Kefan Shao, Qiang Guo, and Chuanhao Wang. Lvi-fusion: A robust lidar-visual-inertial slam scheme. *Remote Sensing*, 16(9), 2024.

[24] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3992–4003, 2023.

[25] Xu Zhao, Wenchao Ding, Yongqi An, Yinglong Du, Tao Yu, Min Li, Ming Tang, and Jinqiao Wang. Fast segment anything. *arXiv preprint arXiv:2306.12156*, 2023.

[26] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012.

[27] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jürgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9296–9306, 2019.

[28] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, page 61–70, 2006.