Technical Report

IRI-TR-26-02

Institut de Robòtica i Informàtica Industrial

"OptiSight: Bridging semantic reasoning and geometric control for embodied navigation"

Alperen Avan
Jordi Sanchez Riera

August 2026

CSIC
Consejo Superior de Investigaciones Científicas

UPC
UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

OptiSight: Bridging Semantic Reasoning and Geometric Control for Embodied Navigation

Alperen Avan Jordi Sanchez-Riera
Institut de Robòtica i Informàtica Industrial (CSIC-UPC)

***Abstract*—Autonomous indoor navigation requires both semantic understanding and precise geometric control. We propose OptiSight, a hybrid framework that combines Vision-Language Model reasoning with deterministic visual servoing through a finite-state Chain-of-Thought architecture. Grounded-SAM localizes open-vocabulary targets, while camera projection geometry converts visual observations into navigation commands without requiring dense mapping. The VLM is queried only at key decision points, reducing computational overhead while geometric control handles continuous navigation. Experiments in AI Habitat demonstrate reliable zero-shot navigation across diverse indoor scenarios, including obstacle avoidance and semantic ambiguity, while operating within an 8 GB VRAM budget. The source code is available at https://github.com/avanalperen/OptiSight-Python-Multimodal-CoT-for-Visual-Reasoning.**

***Index Terms*—Embodied AI, Vision-Language Models, Chain-of-Thought Reasoning, Visual Servoing, Autonomous Navigation.**

I. Introduction

Autonomous navigation in complex, unstructured indoor environments remains a fundamental challenge in robotics and embodied artificial intelligence. Beyond reaching a target location while avoiding obstacles, intelligent agents must understand the semantic structure of their surroundings to execute high-level tasks such as locating a doorway, identifying a navigable passage, or searching for a previously unseen object. Achieving this capability requires combining geometric navigation with semantic scene understanding, enabling robots to reason about both the physical layout and the meaning of the environment.

Traditional navigation systems are predominantly built upon Simultaneous Localization and Mapping (SLAM) and classical geometric path-planning algorithms. These methods have demonstrated remarkable success in localization, map construction, and collision-free navigation while remaining computationally efficient enough for real-time operation. However, because they rely primarily on geometric representations, their ability to interpret semantic information is inherently limited. Consequently, they struggle with tasks involving open-vocabulary object recognition, contextual scene understanding, and natural-language instructions, making them unsuitable for many embodied AI applications.

Manuscript created October, 2020; This work was developed by the IEEE Publication Technology Department. This work is distributed under the LaTeX Project Public License (LPPL) (http://www.latex-project.org/) version 1.3. A copy of the LPPL, version 1.3, is included in the base LaTeX documentation of all distributions of LaTeX released 2003/12/01 or later. The opinions expressed here are entirely that of the author. No warranty is expressed or implied. User assumes all risk.

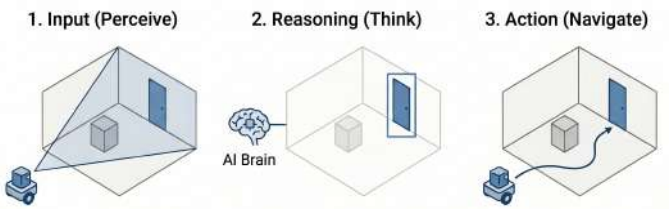

Fig. 1: Conceptual overview of the OptiSight framework. The system perceives the environment, isolates the target via VLM reasoning, and executes a smooth geometric path.

Recent advances in Vision-Language Models (VLMs) have significantly expanded the semantic capabilities of robotic systems. Trained on large-scale multimodal datasets, these models provide powerful zero-shot perception and semantic reasoning capabilities, enabling robots to recognize previously unseen objects, understand contextual relationships, and interpret natural-language instructions without task-specific retraining. These advances have brought embodied agents closer to human-like scene understanding and represent an important step toward more intelligent autonomous systems.

Despite these capabilities, VLMs are not designed to directly control robotic platforms. While they excel at interpreting visual scenes and reasoning about semantic concepts, they do not naturally translate high-level goals into executable navigation strategies or precise motion commands. Furthermore, state-of-the-art VLMs impose substantial computational and memory requirements, resulting in inference latencies that are often incompatible with real-time control on resource-constrained platforms. Even when a VLM correctly infers a semantic concept—for example, identifying the doorway that leads outside a room—it lacks the geometric precision required to generate collision-free trajectories and continuous motion commands.

Consequently, neither purely geometric navigation nor purely semantic reasoning alone provides a complete solution for autonomous embodied navigation. An effective navigation framework must combine the semantic understanding and reasoning capabilities of VLMs with the efficiency, precision, and robustness of classical geometric navigation. Moreover, it should decompose abstract user instructions, such as *"get out of the room"*, into a sequence of intermediate semantic objectives that can be safely executed by a low-level navigation controller while remaining computationally lightweight enough for deployment on edge devices and wearable robotic platforms.

To address these challenges, we propose **OptiSight**, a hy-

brid autonomous navigation framework that bridges semantic understanding and geometric navigation through three complementary stages: *Perceive*, *Think*, and *Navigate* (Fig. 1). Given a high-level instruction, OptiSight first perceives the surrounding environment, then reasons about the semantic actions required to accomplish the task, and finally translates these decisions into precise geometric control commands for robust real-time navigation.

Operating within the AI Habitat simulation environment, OptiSight first perceives the surrounding scene, then employs a Vision-Language Model to identify and reason about semantic targets, and finally converts these high-level observations into precise geometric navigation commands through mathematically grounded visual servoing. By combining semantic reasoning with deterministic geometric control, OptiSight enables robust and efficient navigation while remaining suitable for real-time deployment on computationally constrained platforms.

A key component of OptiSight is a semantic-to-geometric translation module that directly converts semantic observations into navigation actions. Existing embodied navigation approaches typically address this problem either through end-to-end reinforcement learning or by constructing dense semantic scene representations. Reinforcement learning methods often suffer from poor sim-to-real transfer and require extensive task-specific training, whereas semantic SLAM and neural scene representation techniques incur substantial computational and memory costs that limit their deployment on lightweight hardware. Instead, OptiSight leverages Grounded-SAM [1] to identify open-vocabulary targets and extracts their semantic boundaries in the image plane. Using differentiable camera projection geometry, these observations are transformed directly into perspective-correct navigation angles (θ, α), enabling precise closed-loop visual servoing without requiring dense mapping or neural scene reconstruction.

A second challenge lies in efficiently integrating VLM reasoning into the navigation loop. Many recent embodied AI systems continuously query the VLM throughout navigation, introducing substantial inference latency due to repeated multimodal reasoning. This frequent interaction not only limits real-time responsiveness but also increases computational and memory requirements, making deployment on edge devices impractical. OptiSight addresses this limitation through a state-driven Chain-of-Thought architecture based on a finite-state machine comprising *Search*, *Find*, *Scan*, *Navigate*, and *Recover* states. Rather than invoking the VLM continuously, semantic reasoning is performed only at critical state transitions, while deterministic visual servoing governs continuous low-level motion. This separation between high-level cognition and low-level control substantially reduces inference overhead while preserving explainable reasoning and real-time navigation performance.

Finally, we validate OptiSight through extensive experiments in AI Habitat [2] using the Habitat-Sim simulator [3]. We demonstrate that the proposed framework achieves reliable semantic navigation in complex indoor environments while maintaining the computational efficiency required for deployment on resource-constrained robotic platforms.

II. Related work

A. Vision-Language Models for Embodied Navigation

The rapid progress of Large Language Models (LLMs) and Vision-Language Models (VLMs) has significantly advanced embodied artificial intelligence by enabling robots to reason about complex environments using natural language instructions. Rather than relying solely on geometric representations, these models provide semantic understanding, open-vocabulary perception, and contextual reasoning, allowing agents to execute tasks beyond traditional navigation objectives.

Recent foundation models have demonstrated increasingly general embodied capabilities. HabitatGS [4] extends AI Habitat with photorealistic Gaussian Splatting scenes populated by dynamic humans, providing a more realistic benchmark for embodied reasoning. Embodied foundation models such as EmbodiedFM [5] are trained on millions of navigation demonstrations spanning multiple robotic domains—including household robots, drones, and autonomous vehicles—and demonstrate strong zero-shot generalization across diverse navigation tasks.

Several works have also investigated reasoning-driven navigation. OctoNav [6] introduces a think-before-action paradigm in which navigation policies are trained from instruction-trajectory pairs using reinforcement learning. NavR1 [7] further incorporates Group Relative Policy Optimization (GRPO) together with Chain-of-Thought supervision to jointly learn dialogue, planning, reasoning, and navigation. These methods demonstrate that explicit reasoning improves embodied decision making, although they generally require extensive training datasets and computationally expensive foundation models.

Unlike these approaches, OptiSight does not require end-to-end policy learning or large-scale navigation datasets. Instead, it leverages pretrained VLMs only for semantic perception and high-level reasoning, while delegating continuous motion generation to deterministic geometric control.

B. Semantic and Object-Goal Navigation

Object-goal navigation has emerged as a fundamental benchmark for embodied AI, requiring agents to locate objects specified by semantic categories rather than predefined coordinates. OVON [8] significantly expanded this setting by introducing over 15,000 object instances and demonstrating open-vocabulary object navigation using semantic representations.

Subsequent work has focused on improving navigation efficiency through memory-augmented reasoning. Efficient-Nav [9] combines zero-shot LLM planning with semantic memory retrieval and memory clustering to reduce repeated reasoning and inference latency. Hierarchical reasoning frameworks such as HiRobot [10] decompose complex user instructions into hierarchical subgoals while incorporating user feedback to refine navigation plans.

Although these approaches improve semantic navigation, they typically rely on repeated LLM or VLM inference throughout execution, increasing computational overhead. In

contrast, OptiSight invokes semantic reasoning only at critical decision points while maintaining continuous navigation through lightweight visual servoing, substantially reducing latency.

C. Semantic Grounding and Chain-of-Thought Reasoning

A growing body of work has explored explicit reasoning mechanisms for embodied agents. Chain-of-Thought (CoT) prompting was introduced by Wei *et al.* [11], demonstrating that explicitly generating intermediate reasoning steps substantially improves the ability of large language models to solve complex multi-step problems. Building upon this foundation, recent research has extended CoT reasoning to robotics by integrating semantic reasoning with perception and control. For example, Shen *et al.* [12] proposed a multimodal CoT framework for collaborative multi-robot semantic navigation, while Wang *et al.* [13] incorporated deductive CoT reasoning into socially-aware navigation world models. These works highlight the growing importance of structured reasoning for embodied intelligence.

More recent methods integrate CoT directly into embodied navigation pipelines. NavCoT [14], CoTVLA [15], UniNaVid [16], and FantasyVLN [17] incorporate explicit reasoning into navigation policies to improve long-horizon planning. Dynamic 3D-VLP [18] further proposes a unified 3D Chain-of-Thought framework that jointly performs planning, grounding, navigation, and question answering within a single 3D vision-language model. In parallel, SceneGraph-VLM-based approaches [19], [20], [21], [22], [23], [24], [25] exploit structured scene graphs to improve semantic grounding and spatial reasoning.

Beyond navigation, similar reasoning paradigms have also been explored for robotic manipulation. ALRM [26] combines ReAct-style reasoning with executable code generation and tool-based planning to perform long-horizon manipulation tasks.

While these methods demonstrate the benefits of explicit reasoning, many continuously query large VLMs throughout task execution, resulting in substantial computational cost and inference latency. In contrast, OptiSight adopts a state-driven Chain-of-Thought architecture in which semantic reasoning is invoked only at key state transitions within a finite-state machine, while continuous low-level motion is handled by deterministic visual servoing. This separation between high-level reasoning and geometric control enables explainable navigation while maintaining real-time performance on resource-constrained robotic platforms.

III. Methodology

OptiSight is a closed-loop autonomous navigation framework that integrates Vision-Language Model (VLM) reasoning, open-vocabulary object grounding, geometric projection, and deterministic motion control. In its current implementation, the system is designed to execute the high-level navigation instruction *"Get out of the room"*. To accomplish this task, the framework decomposes the instruction into a sequence of semantic reasoning and geometric navigation stages, as illustrated in Figure 2.

The execution flow is governed by a finite-state machine (FSM) comprising five states: *Search*, *Find*, *Scan*, *Navigate*, and *Recover*. Each state encapsulates a specific subtask and defines both the required perception module and the transition conditions to subsequent states. Rather than continuously querying the VLM throughout execution, OptiSight invokes it only at semantic decision points. Each state is associated with a dedicated prompt template that provides task-specific context, allowing the FSM itself to maintain the execution context and eliminating the need for long conversational histories. During the *Search* state, the VLM analyzes the current scene to determine whether the semantic target is visible and decides the next navigation action. Once the target has been identified, the framework transitions to deterministic geometric processing, and the VLM is not queried again unless the system enters the *Recover* state.

The nominal execution pipeline follows the sequence *Search* $\rightarrow$ *Find* $\rightarrow$ *Scan* $\rightarrow$ *Navigate*. In the *Find* stage, Grounded-SAM [1] localizes the target object using open-vocabulary segmentation. The segmented target is then projected into the 3D camera reference frame to estimate its spatial location. Subsequently, during the *Scan* stage, the camera is automatically tilted downward by 40° to analyze the traversable floor area and detect potential obstacles. Based on the resulting geometric representation, the planner computes either a direct trajectory toward the target or a collision-free bypass trajectory around detected obstacles.

During the *Navigate* state, the camera maintains its downward orientation to continuously monitor the planned path while proportional steering commands drive the robot along a sequence of 3D waypoints. Waypoints are generated from the planned trajectory and are sequentially discarded once the robot approaches within $0.30\,\mathrm{m}$ of the current target waypoint, ensuring smooth trajectory tracking and stable motion execution. If the semantic target is lost or the planned path becomes invalid, the FSM transitions to the *Recover* state, where the VLM is invoked again to re-establish the navigation objective before resuming the execution pipeline.

The following subsections describe the operations performed in each state of the finite-state machine and the actual implementation of the proposed method.

A. Finite State Machine (FSM)

Search. The *Search* state is responsible for identifying the semantic target specified by the navigation task, see Figure 3. The current RGB image is provided to the Vision-Language Model (VLM), which determines whether the target (e.g., a doorway) is visible and recommends the next navigation action. If the target is not detected, the robot executes an autonomous scanning procedure consisting of nine successive rotations of 10°, separated by $0.3\,\mathrm{s}$ intervals, allowing the camera to inspect the surrounding environment until the target enters the field of view.

Find. Once the target has been detected, the framework transitions to the *Find* state, where Grounded-SAM [1] per-

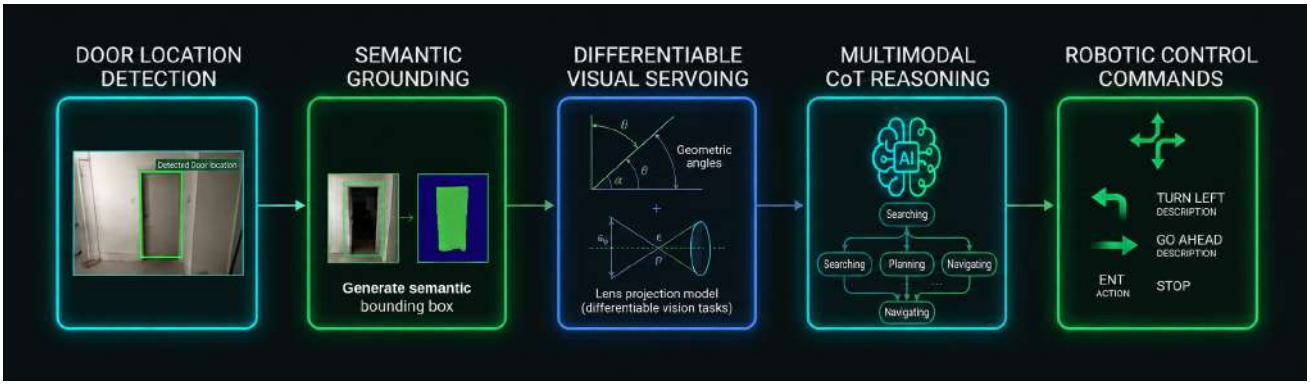

Fig. 2: Conceptual overview of the OptiSight framework. The system perceives the environment, isolates the target via VLM reasoning, and executes a smooth geometric path.

Fig. 3: (Step Find) Semantic object localization and segmentation for a given object target.

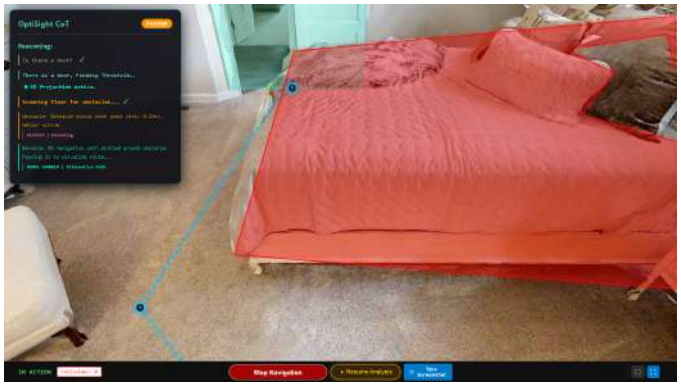

Fig. 4: (Step Scan) Obstacle detection given waypoint path to achieve final goal.

forms open-vocabulary localization to obtain an accurate segmentation of the target. To improve localization stability, the segmentation is refined through up to three consecutive inference passes, producing a consistent low-noise bounding region. The resulting image coordinates are then projected into the 3D camera reference frame using the geometric projection model, yielding the spatial location of the navigation target. Upon successful projection, the system proceeds to the *Scan* state.

Scan. During the *Scan* state, the camera is automatically tilted downward by 40° to maximize visibility of the traversable floor area. Grounded-SAM is then employed to segment potential obstacles, whose image coordinates are projected into 3D space to construct a local obstacle representation. If an obstacle lies within a predefined safety margin of $0.50\,\mathrm{m}$, the planner generates a collision-free bypass trajectory around the obstacle while maintaining continuous visual monitoring through the downward-facing camera, see Figure 4. Otherwise, a direct trajectory toward the target is computed and the camera is restored to its nominal horizontal orientation. This procedure enables deterministic obstacle avoidance without requiring repeated VLM inference during navigation.

Navigate. After a valid trajectory has been generated, the FSM transitions to the *Navigate* state. The robot follows the planned trajectory through a sequence of 3D waypoints using proportional visual servoing. At each control cycle, the controller computes the lateral deviation of the active waypoint with respect to the camera center and generates steering commands scaled according to the waypoint's projected size, ensuring stable motion under the simulator's kinematic constraints, see Figure 5. A waypoint is considered reached once the Euclidean distance between the robot and the waypoint falls below $0.30\,\mathrm{m}$, after which the controller automatically switches to the next waypoint. To account for dynamic changes in the environment, the navigation process is periodically interrupted every 15 control steps, returning the FSM to the *Scan* state to reassess floor traversability and update the planned trajectory if new obstacles have appeared.

B. Dual-Isolated System Architecture

OptiSight adopts a dual-isolated execution architecture that separates semantic reasoning from geometric navigation into two independent runtime environments. The two environments communicate through a lightweight socket-based interface, allowing each component to operate within its own software stack while exchanging only the information required for navigation.

The first environment executes Habitat-Sim together with the geometric navigation pipeline, including the finite-state machine, camera control, geometric projection, path planning, waypoint generation, and motion control. The second environment hosts the Vision-Language Model (VLM) and the

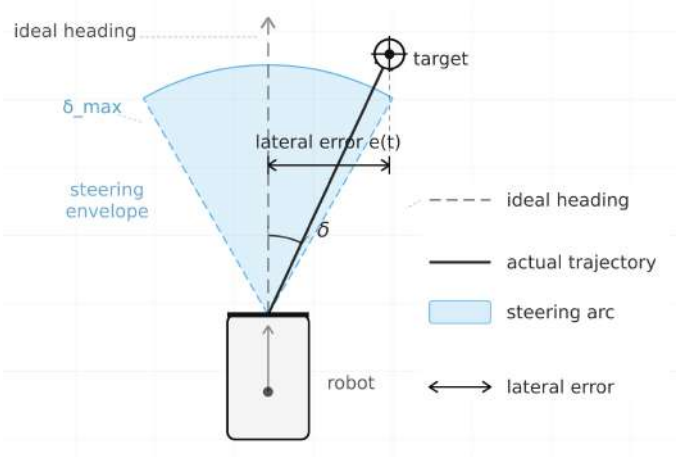

Fig. 5: (Step Navigate) Steering calculation towards the target.

Grounded-SAM inference pipeline responsible for semantic reasoning and open-vocabulary object localization.

Communication between both environments is asynchronous. Whenever semantic reasoning is required, the navigation module sends the current observation and task prompt to the VLM server, which performs inference independently and returns the resulting semantic prediction. Since the VLM is queried only at the semantic decision points defined by the finite-state machine, geometric navigation continues uninterrupted while reasoning is executed in the background. Once the response is received, the finite-state machine resumes execution using the inferred semantic information.

This architecture isolates the dependencies of the robotics and VLM frameworks, enabling the use of modern multimodal models together with legacy robotics software without compatibility issues. In addition, separating semantic inference from geometric control improves resource utilization and allows the complete system to operate within an 8,GB VRAM budget suitable for deployment on resource-constrained edge platforms such as NVIDIA Jetson [27].

IV. Experiments

We evaluate OptiSight in the AI Habitat simulation environment across 24 diverse indoor navigation scenarios. In each scenario, the agent is tasked with executing the high-level instruction *"Get out of the room."* and is evaluated over 10 independent runs. The first 12 scenarios use Qwen3.5-2B [28] for VLM reasoning, while the remaining 12 scenarios use Moondream2-2B [29]. Grounded-SAM is used throughout all experiments for open-vocabulary target localization.

The selected scenarios are designed to evaluate the robustness of OptiSight under six representative navigation challenges: (i) single-obstacle avoidance, (ii) dual-obstacle reasoning, (iii) semantic disambiguation between structurally similar objects, (iv) partial observability requiring active visual search, (v) extreme viewpoints producing geometric distortions, and (vi) perceptual ambiguities caused by reflective surfaces. The first three scenario types primarily evaluate navigation and semantic grounding, whereas the latter three focus on perception and geometric ambiguities that can challenge VLM-based reasoning. This set of scenarios therefore provides a diverse evaluation of both the semantic and geometric components of the proposed framework.

All experiments are executed using a fully automated evaluation pipeline. During each run, the system records quantitative navigation metrics together with structured step-by-step execution traces, including state transitions, VLM requests, parsing events, collisions, and recovery actions. For each scenario, the reported results correspond to the mean values computed over the 10 independent runs. The results are summarized in Tables I and II.

We evaluate OptiSight using the following metrics:

- **Mission Success:** Binary indicator of whether the navigation objective is successfully completed. A scenario is considered successful if the mission is completed in at least half of the 10 independent runs.
- **Success Rate:** Percentage of successful runs across the 10 trials.
- **Execution Time:** Time required to complete the navigation task, measured in seconds.
- **Total Distance:** Cumulative distance traveled by the agent, measured in meters.
- **Total Steps:** Total number of recorded state transitions, actions, and movement steps during execution.
- **Collisions:** Number of physical collisions with environmental obstacles.
- **Recoveries:** Number of recovery or obstacle-avoidance procedures triggered during navigation.
- **VLM Requests:** Number of VLM inference calls performed during a navigation episode.
- **Minimum Obstacle Distance:** Minimum Euclidean distance, in meters, between the agent and any detected 3D obstacle during navigation.

Together, these metrics characterize both task-level performance and the internal behavior of the system. Success rate, execution time, and traveled distance measure navigation effectiveness, while collisions and minimum obstacle clearance quantify safety. VLM requests, parsing failures, and recovery actions provide additional measures of computational and operational robustness.

A. Results

Table I summarizes the performance of OptiSight across the first 12 experiments. Overall, the framework successfully completes 8 out of 12 scenarios according to the defined mission-success criterion, with successful scenarios generally requiring only a small number of VLM requests and no collisions or recovery actions. Experiments 01–03, 07, 09, 11, and 12 achieve success rates between 60% and 100%, while Experiments 03, 07, and 09 achieve a 100% success rate. In Figure 6 we show the starting frame in the scenario.

The results also reveal a clear relationship between navigation difficulty and the amount of reasoning required. In the successful scenarios, OptiSight typically requires only one to three VLM requests, whereas the more challenging scenarios require substantially more interactions. For example, Experiments 05, 06, and 08 require 8, 6, and 6 VLM requests, respectively,

Metric	Exp 01	Exp 02	Exp 03	Exp 04	Exp 05	Exp 06	Exp 07	Exp 08	Exp 09	Exp 10	Exp 11	Exp 12
Mission Success	✓	✓	✓	✗	✗	✗	✓	✗	✓	✗	✓	✓
Success Rate	80%	60%	100%	20%	0%	0%	100%	0%	100%	0%	80%	70%
Execution Time	31.5 s	29.2 s	24.7 s	21.3 s	59.0 s	62.4 s	31.6 s	68.7 s	32.1 s	55.3 s	19.0 s	29.7 s
Total Distance	5.03 m	5.28 m	4.00 m	1.28 m	8.07 m	6.18 m	4.5 m	6.92 m	2.5 m	5.74 m	2 m	6.03 m
Total Steps	64	62	50	35	111	118	50	95	44	71	32	57
Collisions	0	0	0	0	3	6	0	3	0	2	0	0
Recoveries	0	0	0	0	3	3	0	3	0	2	0	0
VLM Requests	2	2	2	3	8	6	2	6	3	2	1	1
Min Obstacle Dist	0.34 m	0.27 m	1.33 m	N/A	0.95 m	0.71 m	1.67 m	0.53 m	N/A	0.76 m	0.74 m	N/A

TABLE I: Performance of OptiSight across Experiments 01–12 using Qwen3.5-2B for VLM reasoning. Results are averaged over 10 independent runs for each scenario.

(a) Exp 01 (b) Exp 02 (c) Exp 03 (d) Exp 04

(e) Exp 05 (f) Exp 06 (g) Exp 07 (h) Exp 08

(i) Exp 09 (j) Exp 10 (k) Exp 11 (l) Exp 12

Fig. 6: Initial frames across Experiments 01–12 indoor navigation scenarios used for the experimental evaluation of OptiSight. Each frame illustrates the initial observation provided to the agent before navigation begins.

and involve multiple recovery actions and collisions. These experiments also exhibit the longest execution times, ranging from 59.0 to 68.7 s, indicating that the additional reasoning and recovery cycles are associated with increased navigation complexity.

Obstacle avoidance is particularly reflected in the relationship between collisions and recovery actions. Experiments 05, 06, 08, and 10 are the only scenarios in which collisions are recorded, and each collision-prone experiment also triggers recovery actions. In contrast, all scenarios without collisions require no recovery procedure. This indicates that the recovery mechanism is effectively activated when the planned trajectory encounters an obstacle, allowing the system to reassess the environment rather than continuing along an invalid trajectory.

The variation in execution time and traveled distance further reflects the diversity of the evaluated scenarios. Successful experiments generally complete within 19.0–32.1 s and traverse between 2.0 and 6.03 m, although more complex layouts can require longer trajectories. In comparison, Experiments 05, 06, and 08 exhibit substantially longer execution times and trajectories, consistent with the additional obstacle-avoidance and recovery operations.

Table II reports the performance of OptiSight across Experiments 13–24. Overall, 7 of the 12 scenarios satisfy the defined mission-success criterion, with four experiments (19, 20, 23, and 24) achieving a 100% success rate. Experiments 15 and 18 also demonstrate relatively high success rates of 80% and 70%, respectively, indicating that the framework can successfully complete several challenging scenarios despite the increased execution time observed for this group of experiments. In Figure 7 we show the starting frame in the scenario.

Compared with the first set of experiments, these scenarios generally require longer execution times, with values ranging from 46.5 s to 118.1 s. The longest execution times are ob-

Metric	Exp 13	Exp 14	Exp 15	Exp 16	Exp 17	Exp 18	Exp 19	Exp 20	Exp 21	Exp 22	Exp 23	Exp 24
Mission Success	✗	✓	✓	✗	✗	✓	✓	✗	✗	✗	✓	✓
Success Rate	0%	60%	80%	0%	0%	70%	100%	100%	30%	0%	100%	100%
Execution Time	97.3 s	85.4 s	61.0 s	87.6 s	98.9 s	91.8 s	73.1 s	84.0 s	94.2 s	118.1 s	46.5 s	67.7 s
Total Distance	5.59 m	7.96 m	4.75 m	5.37 m	5.12 m	1.49 m	5.0 m	4.5 m	4.5 m	4.48 m	3.75 m	3.49 m
Total Steps	72	88	67	72	77	68	55	70	79	76	40	61
Collisions	2	1	0	2	2	0	0	0	0	4	0	0
Recoveries	2	1	0	2	2	0	0	0	0	2	0	0
VLM Requests	2	1	1	2	3	3	1	4	1	3	1	1
Min Obstacle Dist	0.7 m	0.5 m	0.34 m	1.37 m	0.68 m	0.59 m	0.42 m	0.56 m	0.56 m	0.15 m	0.52 m	0.25

TABLE II: Performance of OptiSight across Experiments 13–24 using Moondream2-2B for VLM reasoning. Results are averaged over 10 independent runs for each scenario.

(a) Exp 13 (b) Exp 14 (c) Exp 15 (d) Exp 16

(e) Exp 17 (f) Exp 18 (g) Exp 19 (h) Exp 20

(i) Exp 21 (j) Exp 22 (k) Exp 23 (l) Exp 24

Fig. 7: Initial frames across Experiments 13–24 indoor navigation scenarios used for the experimental evaluation of OptiSight. Each frame illustrates the initial observation provided to the agent before navigation begins.

served in Experiments 13, 17, and 22, which require 97.3, 98.9, and 118.1 s, respectively. These longer durations are not directly associated with a larger number of VLM requests, since most experiments require only one to three requests. This behavior highlights the benefit of the state-driven architecture, where extended execution can result primarily from geometric navigation, obstacle avoidance, or repeated scanning rather than continuous VLM inference.

Obstacle interactions provide a clearer explanation for several unsuccessful scenarios. Collisions are reported in Experiments 13, 14, 16, 17, and 22, and each of these experiments also triggers recovery actions. In particular, Experiment 22 records four collisions and two recovery procedures and has the lowest minimum obstacle clearance in the table, at only 0.15,m. Experiments 13, 16, and 17 similarly combine collisions with unsuccessful missions, suggesting that navigation becomes more challenging when obstacles are encountered at close proximity. In contrast, Experiments 19–21 and 23–24 complete without collisions or recovery actions, despite requiring between one and four VLM requests.

The minimum obstacle distance further illustrates the range of geometric conditions encountered in these scenarios. Successful experiments can operate with relatively small clearances, such as 0.25 m in Experiment 24 and 0.34 m in Experiment 15, while Experiment 22 reaches a minimum clearance of only 0.15 m and experiences multiple collisions. This suggests that maintaining sufficient clearance remains an important factor for reliable navigation, particularly in scenarios involving constrained passages or closely positioned obstacles.

An important observation is that the number of VLM requests remains relatively low across all experiments, ranging from one to four calls. Several successful scenarios, including Experiments 19, 23, and 24, require only a single VLM request. This result is consistent with the state-driven design

of OptiSight, in which semantic reasoning is invoked only at key decision points while geometric planning and waypoint navigation are handled independently.

V. Conclusions

We presented **OptiSight**, a hybrid autonomous navigation framework that combines selective Vision-Language Model reasoning with deterministic geometric navigation. By integrating a finite-state machine, Grounded-SAM-based target grounding, and visual-to-3D geometric projection, OptiSight translates high-level semantic instructions into executable navigation actions while avoiding continuous VLM inference and computationally expensive mapping pipelines.

Evaluation in AI Habitat across 24 indoor navigation scenarios demonstrates that OptiSight can successfully handle diverse navigation conditions, including obstacle avoidance, semantic ambiguity, partial observability, and challenging viewpoints, while requiring only a limited number of VLM queries. These results highlight the potential of combining semantic reasoning with lightweight geometric control for efficient embodied navigation.

Future work will focus on extending the framework to a broader range of navigation tasks and validating OptiSight on physical robotic platforms.

References

[1] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan, Z. Zeng, H. Zhang, F. Li, J. Yang, H. Li, Q. Jiang, and L. Zhang, "Grounded sam: Assembling open-world models for diverse visual tasks," *arXiv preprint arXiv:2401.14159*, 2024.

[2] A. Szot, A. Clegg, E. Undersander, E. Wijmans, Y. Zhao, J. Turner, N. Maestre, M. Mukadam, D. Chaplot, O. Maksymets, A. Gokaslan, V. Vondrus, S. Dharur, F. Meier, W. Galuba, A. Chang, Z. Kira, V. Koltun, J. Malik, M. Savva, and D. Batra, "Habitat 2.0: Training home assistants to rearrange their habitat," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[3] Facebook Research, "Habitat-sim," GitHub Repository, accessed: 2026-03-25. [Online]. Available: https://github.com/facebookresearch/habitat-sim

[4] Z. Xia, J. Xu, C. Cui, Y. Yu, J. Zhang, Q. Yan, T. Ni, J. Chen, X. Zhou, H. Bao, R. Hu, and S. Peng, "Habitat-gs: A high-fidelity navigation simulator with dynamic gaussian splatting," 2026. [Online]. Available: https://arxiv.org/abs/2604.12626

[5] J. Zhang, A. Li, Y. Qi, M. Li, J. Liu, S. Wang, H. Liu, G. Zhou, Y. Wu, X. LI, Y. Fan, W. Li, Z. Chen, F. Gao, Q. Wu, Z. Zhang, and H. Wang, "Embodied navigation foundation model," in *The Fourteenth International Conference on Learning Representations*, 2026. [Online]. Available: https://openreview.net/forum?id=kkBOIsrCXh

[6] C. Gao, L. Jin, X. Peng, J. Zhang, Y. Deng, A. Li, H. Wang, and S. Liu, "Octonav: Towards generalist embodied navigation," in *CVPR*, 2026.

[7] Q. Liu, T. Huang, Z. Zhang, and H. Tang, "Nav-r1: Reasoning and navigation in embodied scenes," *arXiv preprint arXiv:2509.10884*, 2025.

[8] N. Yokoyama, R. Ramrakhya, A. Das, D. Batra, and S. Ha, "Hm3d-ovon: A dataset and benchmark for open-vocabulary object goal navigation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024.

[9] Z. Yang, S. Zheng, T. Xie, T. Xu, B. Yu, F. Wang, J. Tang, S. Liu, and M. Li, "Efficientnav: Towards on-device object-goal navigation with navigation map caching and retrieval," in *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. [Online]. Available: https://openreview.net/forum?id=qMm7tC1zvj

[10] L. X. Shi, B. Ichter, M. Equi, L. Ke, K. Pertsch, Q. Vuong, J. Tanner, A. Walling, H. Wang, N. Fusai, A. Li-Bell, D. Driess, L. Groom, S. Levine, and C. Finn, "Hi robot: open-ended instruction following with hierarchical vision-language-action models," in *Proceedings of the 42nd International Conference on Machine Learning*, ser. ICML'25, 2025.

[11] J. Wei, X. Wang, D. Schuurmans, M. Bosma, b. ichter, F. Xia, E. Chi, Q. V. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 24 824–24 837. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf

[12] Z. Shen, H. Luo, K. Chen, F. Lv, and T. Li, "Enhancing multi-robot semantic navigation through multimodal chain-of-thought score collaboration," in *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, ser. AAAI'25/IAAI'25/EAAI'25, 2025.

[13] W. Wang, O. Ike, S. Choi, S. Hong, and B.-C. Min, "Deductive chain-of-thought augmented socially-aware robot navigation world model," *arXiv preprint arXiv:2510.23509*, 2025. [Online]. Available: https://arxiv.org/abs/2510.23509

[14] B. Lin, Y. Nie, Z. Wei, J. Chen, S. Ma, and J. Han, "Navcot: Boosting llm-based vision-and-language navigation via learning disentangled reasoning," *arXiv preprint arXiv:2403.07376*, 2024. [Online]. Available: https://arxiv.org/abs/2403.07376

[15] Q. Zhao *et al.*, "Cot-vla: Visual chain-of-thought reasoning for vision-language-action models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. [Online]. Available: https://arxiv.org/abs/2503.22020

[16] J. Zhang, K. Wang, S. Wang, M. Li *et al.*, "Uni-navid: A video-based vision-language-action model for unifying embodied navigation tasks," *arXiv preprint arXiv:2412.06224*, 2024. [Online]. Available: https://arxiv.org/abs/2412.06224

[17] J. Zuo, L. Mu, F. Jiang, C. Ma, M. Xu, and Y. Qi, "Fantasyvln: Unified multimodal chain-of-thought reasoning for vision-language navigation," *arXiv preprint arXiv:2601.13976*, 2026. [Online]. Available: https://arxiv.org/abs/2601.13976

[18] Z. Wang, S. Lee, G. Dai, and K. M. Lee, "D3d-vlp: Dynamic 3d vision-language-planning model for embodied grounding and navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

[19] V. Makarov, M. Vasilkov, and A. Sanakoyeu, "Scenegraphvlm: Dynamic scene graph generation from video with vision-language models," 2026. [Online]. Available: https://arxiv.org/abs/2605.13667

[20] J. Strader, N. Hughes, W. Chen, A. Speranzon, and L. Carlone, "Indoor and outdoor 3d scene graph generation via language-enabled spatial ontologies," *IEEE Robotics Autom. Lett.*, vol. 9, no. 6, pp. 4886–4893, 2024.

[21] J. Loo, Z. Wu, and D. Hsu, "Open scene graphs for open-world object-goal navigation," *The International Journal of Robotics Research*, 2025.

[22] A. Rajvanshi, K. Sikka, X. Lin, B. Lee, H.-P. Chiu, and A. Velasquez, "Saynav: Grounding large language models for dynamic planning to navigation in new environments," in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 34, 2024, pp. 464–474. [Online]. Available: https://ojs.aaai.org/index.php/ICAPS/article/view/31506

[23] Y. Hu, J. Wu, R. Xu, H. Liu, A. Xi, H. X. Liu, R. Vasudevan, and M. Ghaffari, "Imaginative world modeling with scene graphs for embodied agent navigation," *arXiv preprint arXiv:2508.06990*, 2025.

[24] X. Huang, J. Zhao *et al.*, "Msgnav: Unleashing the power of multi-modal 3d scene graph for zero-shot embodied navigation," *arXiv preprint arXiv:2511.10376*, 2025.

[25] M. Zawalski, W. Chen, K. Pertsch, O. Mees, C. Finn, and S. Levine, "Robotic control via embodied chain-of-thought reasoning," *arXiv preprint arXiv:2407.08693*, 2024. [Online]. Available: https://arxiv.org/abs/2407.08693

[26] V. G. d. Santos, I. Khadraoui, I. Farhat, H. Yous, S. Teffahi, and H. Hacid, "Alrm: Agentic llm for robotic manipulation," *arXiv preprint arXiv:2601.19510*, 2026.

[27] NVIDIA, "Deploying open source vision language models (vlm) on jetson," Hugging Face Blog, Feb. 2026, accessed: 2026-03-25. [Online]. Available: https://huggingface.co/blog/nvidia/cosmos-on-jetson

[28] Q. Team, "Qwen3.5-omni technical report," 2026. [Online]. Available: https://arxiv.org/abs/2604.15804

[29] vikhyat, "moondream2," 2024. [Online]. Available: https://huggingface.co/vikhyatk/moondream2