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

SurveilNav: Collaborative Object Goal Navigation with Robot and Surveillance System

Ming-Ming Yu[1,2], Qunbo Wang[3†], Rongtao Xu[4], Yanghong Mei[2,5], Yirong Yang[1,2], Longteng Guo[2,5], Wenjun Wu[1,6], and Jing Liu[2,5†]

***Abstract*— With the growing deployment of surveillance systems in factories, offices, and homes, integrating them with robots offers a promising direction for collaborative and efficient task execution. However, existing approaches largely focus on single-robot scenarios and struggle with multi-view collaboration in large-scale environments. In this paper, we present a novel indoor collaborative object navigation dataset built on Habitat-Sim, featuring 206 cameras across 74 floors. The dataset enables systematic evaluation of an agent's ability to exploit multi-view surveillance information. To address the limitations of single-robot perception, we propose SurveilNav, a collaborative navigation framework that integrates active camera scheduling, joint 2D/3D mapping, VLM-based value estimation, and collaborative target verification. By synergizing the robot's dynamic local perception with the static global view of surveillance, this architecture effectively overcomes both the limited perception range of single agents and the inherent blind spots of fixed cameras, resolving inefficient exploration. Experimental results on the HM3D dataset demonstrate that SurveilNav substantially outperforms existing methods, achieving state-of-the-art performance in both exploration efficiency and navigation success rate. Moreover, the system shows strong potential for applications in large-scale search, home environments, and rescue missions.**

I. Introduction

Object goal navigation (ObjectNav) requires an agent to automatically locate a specified object in previously unseen environments, posing a fundamental challenge for embodied intelligence. With the rapid advancement of large language models (LLMs) and vision-language models (VLMs), zero-shot ObjectNav has made significant progress. These approaches typically construct explicit environmental maps and leverage LLMs or VLMs to reason over them, selecting valuable frontiers or candidate waypoints for exploration. Recent works [1]–[5] have demonstrated remarkable improvements in exploration efficiency. By leveraging the extensive prior knowledge within LLMs, these methods achieve performance comparable to models trained on massive navigation trajectories. Compared with data-driven approaches, they often exhibit superior generalization to novel environments.

Nonetheless, a key limitation remains: nearly all existing methods are confined to single-robot systems, without leveraging multi-sensor or cross-view collaboration. In real-world scenarios, a single egocentric viewpoint often provides limited coverage. This problem becomes more pronounced in large-scale, multi-floor environments, where both exploration efficiency and accuracy drop significantly. A similar issue has long been recognized in autonomous driving: single-vehicle perception is insufficient for safe decision-making. To address this, researchers have developed collaborative perception, where vehicles, roadside units, and infrastructure share information to expand perceptual coverage and enhance system accuracy and robustness [6]–[8]. Inspired by this, we argue that indoor navigation can also benefit from robot–surveillance collaboration. While collaborative perception has been explored in autonomous driving, systematic exploration of how to effectively coordinate robots with surveillance systems in indoor settings remains lacking.

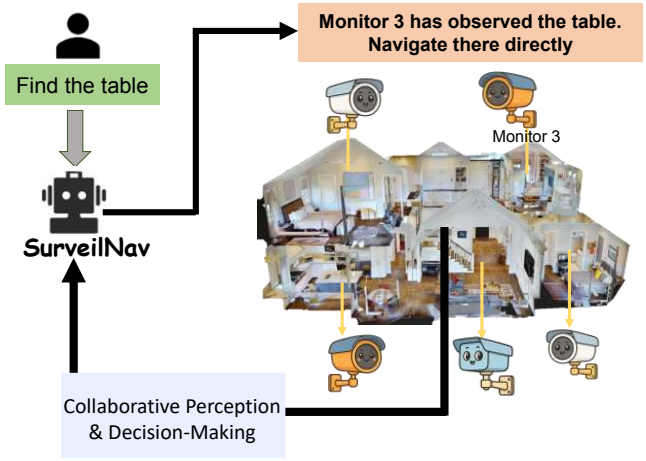

Fig. 1: SurveilNav workflow. Monitor #3 detects the target table and provides direct navigation guidance, enabling the agent to efficiently localize and approach the target through collaborative perception and decision-making.

To bridge this gap, we introduce a new research task: collaborative object-goal navigation with robots and surveillance systems. For this purpose, we construct the first dataset on Habitat-Sim, spanning 36 scenes, 74 floors, and 206 surveillance cameras. As illustrated in Figure 1, agents are required not only to explore using their onboard perception, but also to actively query surveillance viewpoints to enhance efficiency and accuracy in target localization. This formulation mirrors real-world scenarios where robots collaborate with infrastructure to accomplish complex tasks.

Building on this dataset, we introduce **SurveilNav**, a novel collaborative object navigation framework. Our framework integrates active camera invocation, joint 3D map construction, VLM-based semantic value estimation, and target

[1]Beihang University; [2]Institute of Automation, Chinese Academy of Sciences; [3]Beijing Jiaotong University; [4]ATeam; [5]University of Chinese Academy of Sciences; [6]Hangzhou International Innovation Institute, Beihang University. [†]Corresponding authors. Emails: mingmingyu@buaa.edu.cn, wangqb6@outlook.com

verification mechanisms. Specifically, the agent dynamically coordinates with the surveillance network to decide whether to directly approach a localized target or conduct autonomous exploration guided by collaborative perception. By leveraging this integrated architecture, SurveilNav achieves a powerful synergy between global and local perception. It not only breaks through the bottleneck of limited perception range in single robots and compensates for the inherent blind spots of fixed surveillance, but also reduces false detections through multi-view validation. Ultimately, this comprehensive approach substantially improves both exploration efficiency and overall navigation performance.

Experimental results on the HM3D dataset demonstrate that SurveilNav achieves state-of-the-art performance in large-scale and multi-floor scenarios, significantly surpassing existing methods in both exploration efficiency and navigation success rate. Moreover, the system shows strong potential in real-world applications such as large-scale search, household scenarios, and rescue missions where efficient collaborative perception is critical.

Our main contributions are as follows: (1)We propose the first indoor object search dataset for robot–surveillance collaboration, covering 36 scenes, 74 floors, and 206 cameras, providing a new benchmark for multi-view navigation. (2)We introduce SurveilNav, a collaborative object navigation framework based on VLMs, enabling agents to actively invoke surveillance systems for joint map construction, semantic exploration, and target verification. (3)We conduct large-scale experiments on HM3D, showing that SurveilNav significantly outperforms existing methods in both exploration efficiency and navigation success rate, while demonstrating strong practicality.

II. Related Work

A. Visual Navigation

Visual navigation is a fundamental task in embodied intelligence. Recent advancements have introduced a variety of visual navigation tasks, such as point navigation [9]–[11], image-goal navigation [12], object-goal navigation [13], and vision-language navigation [14]–[17]. Point navigation tasks utilize coordinates relative to the robot's starting point as the target, while image-goal navigation aims for a target image. In vision-language navigation, agents follow step-by-step instructions to reach the target location. Compared to these tasks, object-goal navigation employs object category names as targets, demanding more robust exploration capabilities than vision-language navigation tasks. Thus, we focus on the object navigation task within unknown environments.

B. Object Goal Navigation

Existing approaches to object-goal navigation can be broadly divided into two categories. The first class of methods leverages pre-trained vision or language models as backbones, and trains navigation policies with extensive navigation trajectories via supervised or reinforcement learning [12], [18]–[21]. While effective, these approaches face two key limitations: they are confined to the finite set of object categories seen during training, limiting generalization to open environments, and the simulation-to-reality gap reduces the transferability of task-specific training to real-world scenarios. To overcome these challenges, recent work has shifted toward Zero-Shot Object Navigation [1], [4], [5], [22]–[24]. These methods explicitly build environmental maps and employ vision-language models (VLMs) [25], [26] or large language models (LLMs) [27] for reasoning, selecting informative frontiers or waypoints for exploration. For instance, COW [1] guides the robot to explore the nearest frontier until the target is detected using CLIP features [28] and open-vocabulary object detectors [29]. ESC [2], L3MVN [3], and VoroNav [5] enhance decision-making with LLMs, while VLFM [4] uses a VLM to assign semantic values to frontiers based on egocentric observations and textual prompts. To further enhance efficiency, SEEK [30] utilizes spatial priors like floor plans, while GOAT [31] leverages accumulated memory from past tasks. Despite these advances, most approaches remain limited to single-robot systems, though some efforts have begun to explore cross-agent collaboration, such as air-ground robotic teams [32]. In contrast, we propose a collaborative framework that couples mobile robots with a surveillance system to enhance task efficiency.

C. Collaborative perception in autonomous driving

Autonomous driving has attracted significant attention in recent years, but standalone systems often suffer from limited perception ranges, which may lead to safety risks. To overcome this, *collaborative perception* leverages information from multiple vehicles, infrastructure, and roadside units to expand the field of view and improve accuracy. With advances in deep learning and the release of large-scale datasets such as V2X-Sim [6], OPV2V [7], and DAIR-V2X [8], research in this field has accelerated. Collaborative perception methods can be categorized into early (e.g., Cooper [33], Coop3D [34]), intermediate (e.g., V2VNet [35]), and late collaboration (e.g., OptiMatch [36]), each improving robustness through tailored communication, feature fusion, and optimization strategies. However, these efforts have focused almost exclusively on outdoor driving scenarios, leaving open the question of how to effectively coordinate robots and surveillance systems in indoor environments.

III. TASKS AND DATASETS

A. Task Definition

In a surveillance and agent collaborative semantic navigation task, the agent needs to rely on its first-person perspective observations while collaborating with third-person perspective observations from surveillance systems deployed throughout the building. The goal is to search for a specified object category in a previously unseen environment without prior mapping. Formally, the scene set is denoted as $\mathcal{S} = \{s_1, \ldots, s_k\}$, and the category set is denoted as $\mathcal{C} = \{c_1, \ldots, c_m\}$. For a given scene $s_i \in \mathcal{S}$, the deployed surveillance system is defined as $\mathcal{H}_i = \{h_1, h_2, \ldots, h_{n_i}\}$, where n_i represents the specific number of cameras in that

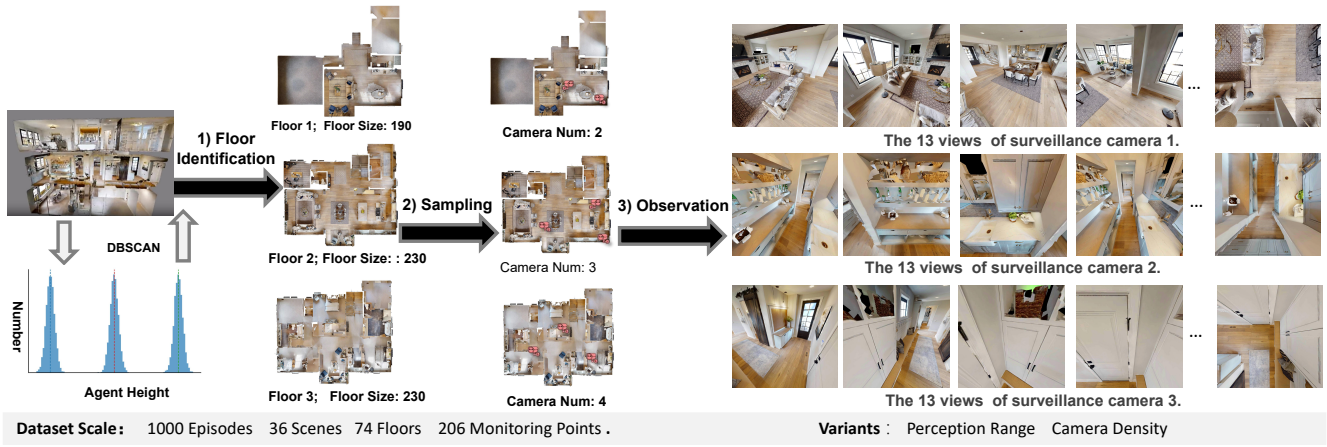

Fig. 2: The surveillance camera observation generation pipeline, consisting of (a) floor identification, (b) camera sampling, and (c) observation configuration.

scene. For each episode, the agent is randomly initialized in an unknown scene s_i and must locate a target category $c \in \mathcal{C}$. At each time step t, the robot acquires a first-person observation O_t, which includes an RGB-D image, the robot's pose (position and orientation), the target category c, as well as the observations and poses of all cameras in the surveillance system $\mathcal{H}_i$. The action space $\mathcal{A}$ consists of *move forward* (0.25m), *turn left* (30°), *turn right* (30°), *look up* (30°), *look down* (30°), and *stop*. An episode is deemed successful if the agent executes the *stop* action when its geodesic distance to the target object is sufficiently small, with a maximum episode length of 500 steps.

B. SurveilNav Dataset Construction

We develop the SurveilNav dataset for collaborative semantic navigation based on Habitat HM3Dv2 [37]. As illustrated in Fig. 2, the construction pipeline consists of three phases: **(1) Floor Identification**: The agent is first placed within the scene, and feasible points are recorded. DBSCAN clustering is then applied to the agent's height coordinates to segment and identify different floor levels. **(2) Camera Sampling**: For each floor, the number of surveillance cameras is determined according to the floor area, with one camera allocated per $100\,\mathrm{m}^2$ and placed at a height of 1.8 meters. During sampling, candidate positions with an excessive proportion of black pixels (caused by artifacts in the 3D scans) are skipped. To ensure optimal coverage, the final placements are refined using farthest point sampling. **(3) Observation Generation**: To simulate realistic, wide-area monitoring, each camera records 13 viewpoints: 12 surrounding views with a $-30°$ tilt (at 30° azimuth intervals) and one $-90°$ nadir view. Each viewpoint captures 1280×1280 RGB-D images and precise poses. While real-world surveillance often lacks depth sensors, modern monocular estimation (e.g., DepthAnything [38]) has demonstrated the feasibility of metric depth recovery. Thus, we utilize ground-truth depth from Habitat-Sim as an empirical upper-bound to isolate the core effectiveness of our collaborative framework from potential depth-estimation noise. Following this pipeline, we generate 1,000 episodes across 36 scenes, 74 floors, and 206 camera placements. In line with the standard ObjNav task definition, the dataset includes six object goal categories: chair, couch, potted plant, bed, toilet, and TV. To further analyze the impact of infrastructure density and visibility, we provide additional variants with varying camera densities (one per $200\,\mathrm{m}^2$ vs. $100\,\mathrm{m}^2$) and field-of-view configurations (panoramic vs. half views).

IV. Methodology

A. Overview

As illustrated in Fig. 3, at each time step t, SurveilNav actively selects surveillance cameras based on its current position $\mathbf{p}_t$ and the distribution of monitoring locations. The system then constructs a unified 3D map representation by integrating egocentric observations from the robot with third-person visual perspectives from multiple cameras. Building on this representation, a joint 2D frontier map is generated to separate explored regions from unexplored areas. Meanwhile, the VLM assesses the relevance between multi-source observations and the task objective, and projects the resulting importance weights onto a 2D plane to form a joint value map. In parallel, the system fuses multi-view detections and semantic information to construct a unified 3D object map, where a confidence-based fusion mechanism ensures accurate target perception. Finally, the system selects the optimal waypoint from candidate objects and frontiers according to the value map, enabling the robot to efficiently achieve object-goal navigation.

B. Active Camera Invocation

To optimize navigation efficiency and minimize computational overhead, the agent dynamically invokes a subset of surveillance cameras, $\mathbf{H}_t \subseteq \mathcal{H}_i$, based on spatial relevance. Let $\mathbf{p}_t = [x_t, y_t, z_t^a]^\top$ denote the agent's global pose at time step t, where z_t^a represents the altitude of the agent's camera center. Similarly, let $\mathbf{p}_j = [x_j, y_j, z_j^c]^\top$ denote the position of surveillance camera $h_j \in \mathcal{H}_i$, where z_j^c is its fixed installation height. To ensure floor-level consistency, we estimate the absolute elevation of the supporting floor plane for both the agent and the cameras. Specifically, let

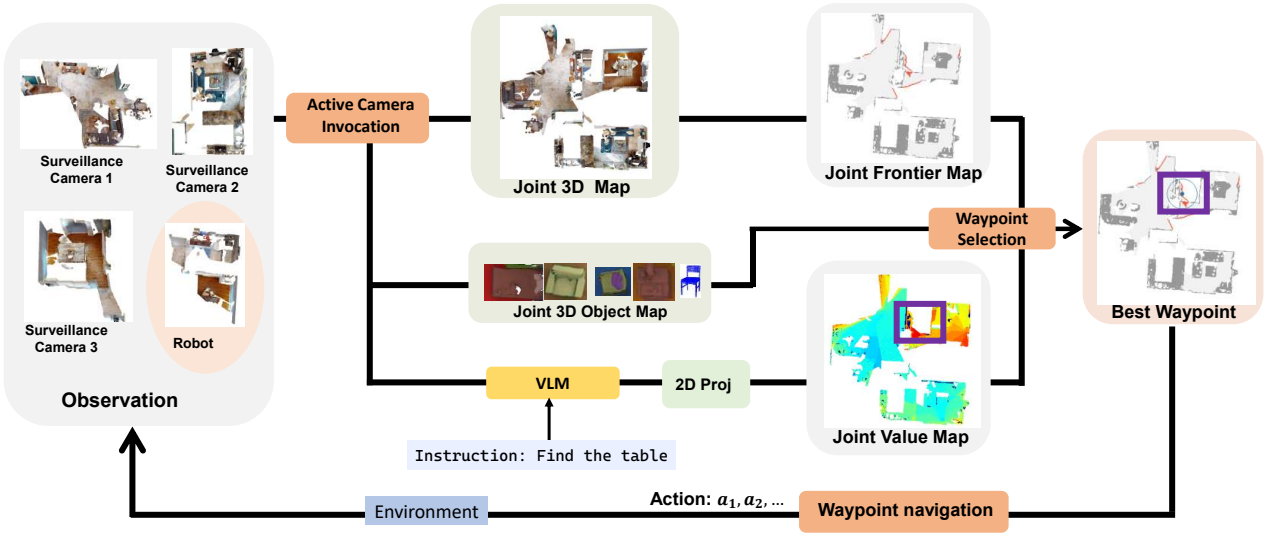

Fig. 3: The proposed system, SurveilNav, consists of several key components: active camera invocation, joint 3D map construction through multi-source observation alignment, joint value map generation with a vision-language model (VLM), joint 3D object confirmation, and waypoint selection and navigation.

$\hat{z}_t^a = z_t^a - \Delta z^a$ and $\hat{z}_j^c = z_j^c - \Delta z^c$ represent the inferred global altitude of the floor surface where the agent and camera h_j are situated, respectively. Here, Δz^a and Δz^c denote the known vertical mounting offsets of the sensors relative to their respective local floors. The activation state of camera h_j, denoted by $\alpha_{j,t} \in \{0, 1\}$, is determined by a vertical alignment constraint:

$$\alpha_{j,t} = \begin{cases} 1, & \text{if } |\hat{z}_t^a - \hat{z}_j^c| \leq \tau_z \\ 0, & \text{otherwise} \end{cases}, \tag{1}$$

where τ_z is a predefined threshold for vertical displacement. The active camera set at time t is formally defined as $\mathbf{H}_t = \{h_j \in \mathcal{H}_i \mid \alpha_{j,t} = 1\}$. This mechanism effectively filters out irrelevant observations from different floors, thereby reducing the computational complexity for subsequent multi-source fusion and ensuring the scalability of the collaborative system in multi-floor environments.

C. Map Representation

1) Joint 3D Point Cloud Map Construction: In the joint 3D point cloud map construction, the robot acquires local point clouds $\mathcal{P}_t$ through its RGB-D sensor, while the surveillance system provides point clouds $\mathcal{P}_j$ from fixed cameras. The robot's local point cloud is transformed into the global coordinate frame using its current pose $T_t = [R_t|\mathbf{t}_t]$, i.e., $\mathbf{p}_t^{\text{global}} = R_t \cdot \mathbf{p}_t + \mathbf{t}_t$, where $\mathbf{p}_t$ represents the point cloud coordinates in the local frame. Similarly, the surveillance point cloud $\mathcal{P}_j$ is transformed into the global frame using its fixed pose $T_j = [R_j|\mathbf{t}_j]$. By aligning and merging $\mathbf{p}_t^{\text{global}}$ and $\mathbf{p}_j^{\text{global}}$, a joint 3D point cloud map is constructed.

2) 2D Map Construction: Once the unified 3D point cloud map is constructed, we project its points onto a 2D plane to derive two complementary representations: an obstacle map and an exploration map. The obstacle map is obtained by projecting points located above the floor level, whereas the exploration map is generated using all available 3D points. To determine the frontier regions, we enlarge the obstacle boundaries through morphological dilation and then subtract the exploration map from the obstacle map. The resulting frontier map marks the interface between explored and unexplored areas. During navigation, both the distribution and the number of frontiers are continuously updated. When the robot has fully explored the environment, these frontiers naturally disappear.

3) Joint Value Map: The joint value map is designed to quantify the semantic relevance of each location in the explored area to the target object, and it has the same shape as the exploration map. For both surveillance images and the robot's onboard observations, we employ CLIP to evaluate task-related relevance. Specifically, we compute the cosine similarity between the RGB image features f_v and a text prompt feature f_t formulated as "seems like there is a [object name]", i.e., $\cos(f_v, f_t)$. By leveraging depth information together with the camera's pose (position and orientation), these relevance scores are projected into the top-down map space. To integrate information from multiple viewpoints, we adopt an averaging fusion strategy. Formally, for each grid cell (i, j), the joint relevance score is updated as:

$$s^{\text{joint}}(i,j) \leftarrow \frac{s^{\text{surveil}}(i,j) + s^{\text{robot}}(i,j)}{2}, \tag{2}$$

where $s^{\text{surveil}}(i,j)$ and $s^{\text{robot}}(i,j)$ denote the relevance values derived from the surveillance system and the robot's local observations, respectively.

D. Joint 3D Object Map and Target Confirmation

1) Joint 3D Object Map Construction: As shown in Figure 4, we process observations captured by both the robot and the surveillance cameras. Specifically, Grounding DINO is adopted as an open-vocabulary detector, and MobileSAM [39] is applied to produce object masks. Using

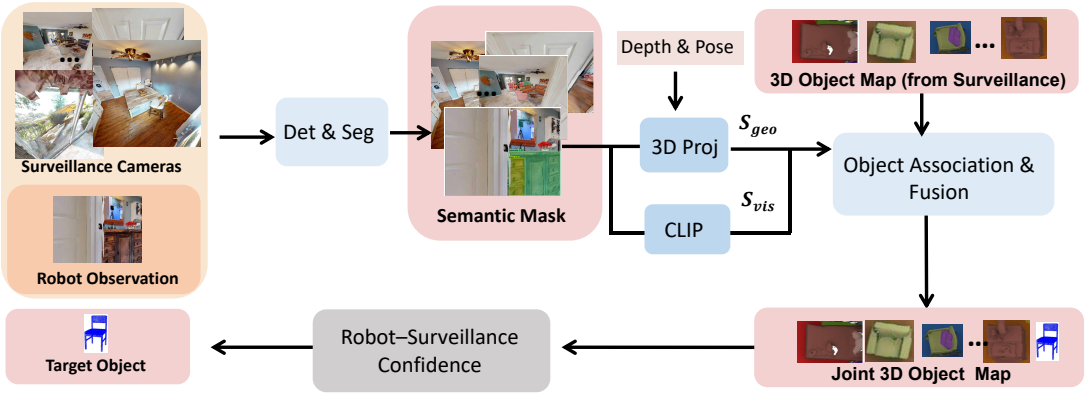

Fig. 4: The process of constructing the joint 3D object map and confirming the target.

depth information, these masks are projected back into 3D space to construct point cloud representations. For each mask, the corresponding image region is cropped according to its bounding box, and CLIP is employed to extract semantic features. This procedure yields a semantic point cloud enriched with confidence values, category labels, and feature embeddings.

Joint Object association. For each newly detected object i from either the **robot** or a **surveillance camera**, we compute both geometric and semantic similarity with respect to all existing objects in the joint map. The *geometric similarity* $S_{\text{geo}}(i,j) = \text{IoU}(\text{PointCloud}_i, \text{PointCloud}_j)$ is calculated as the Intersection over Union (IoU) of the point clouds between the newly detected object and an existing object. The *semantic similarity* $S_{\text{vis}}(i,j) = \text{Sim}(\text{SemanticFeature}_i, \text{SemanticFeature}_j)$ is computed from the semantic feature embeddings of the newly detected object and the existing objects. The overall similarity score between a new detection i (from the robot or a surveillance camera) and an existing object j is defined as a weighted sum of the two similarities:

$$s(i,j) = w_1 S_{\text{vis}}(i,j) + w_2 S_{\text{geo}}(i,j). \tag{3}$$

If this score is lower than a threshold τ, the system registers a new object instance; otherwise, the detection is matched with the existing object with the highest similarity.

Joint Object fusion. For objects successfully associated between the **robot** and **surveillance cameras**, we merge their point clouds into a unified representation. The corresponding features are then updated using a weighted average that reflects the detection frequencies from both sources.

2) Joint Object Confirmation: In this phase, candidate objects extracted from the semantic point cloud are evaluated based on their object confidence. For each object, the confidence score is defined as the maximum value between the local observations from the robot and the global observations from the surveillance cameras. The category of the object is then determined by the class associated with this maximum confidence. If the resulting object confidence surpasses a predefined threshold, the candidate is confirmed as the target. As the robot moves and integrates additional observations, these confidence scores are dynamically updated, enabling more reliable confirmation. If, during this process, the predicted category of an object changes, the detection is regarded as a false positive, which often arises from limited or suboptimal viewpoints of the surveillance system. This confidence-evolution mechanism allows static wide-range observations from surveillance cameras and dynamic local observations from the robot to complement each other, leading to more robust and consistent object confirmation.

E. Waypoint Selection and Navigation

After initialization, the robot adaptively selects waypoints for navigation according to the detection status of target objects. If a target object has been confirmed, its location is directly assigned as the waypoint. Otherwise, the robot chooses either a promising low-confidence candidate object or the highest-value frontier as the exploration waypoint. To reach the designated waypoint from its current position, the robot employs the Fast Marching Method (FMM) [40] as the local planner. The planner iteratively selects feasible local goals within the robot's vicinity and generates corresponding actions to approach the target step by step. At each iteration, both the map and the waypoint are updated with real-time observations, enabling adaptive and accurate navigation.

V. Experiments

A. Experimental Setup.

Metrics. We use navigation success rate (SR) and success rate weighted by navigation path length (SPL) as evaluation metrics. SR represents the percentage of successful episodes out of the total episodes. SPL measures the efficiency of reaching the goal in addition to the success rate.

Implementation Details. In the target association module, the similarity threshold for matching two targets is set to 1.35. The agent captures images at a resolution of 640×480 with a field of view (FOV) of 79 degrees, while the surveillance camera captures images at a resolution of 1280×1280 with an FOV of 100 degrees. For the active invocation module, the threshold for determining whether the surveillance camera and the robot are on the same floor is set to 0.4 meters. For the 2D map, the resolution is configured to 5 cm, with a map size of 2400 cm. For point cloud processing, downsampling uses a 2.5 cm voxel grid,

TABLE I: Comparison with single-robot navigation methods on the HM3D dataset (results from MCoCoNav [41]).

Method	Zero-Shot	Training-Free	LLM/VLM	HM3D-v0.2	
				SPL↑	SR↑
Random Walking	✓	✓	None	0.000	0.000
Frontier Based [40]	✓	✓	None	12.3	23.7
Random Samples	✓	✓	None	14.3	30.0
VLFM [4]	✓	✗	BLIP	32.7	64.0
L3MVN [3]	✓	✓	RoBERTa-large	23.1	50.4
ESC [2]	✓	✓	GPT-3.5	22.3	39.2
Single-NavGPT [42]	✓	✓	GPT-3.5	21.5	53.9
VoroNav [5]	✓	✓	GPT-3.5	26.0	42.0
OpenFMNav [22]	✓	✓	GPT-4/GPT-4V	24.4	54.9
MCoCoNav [41]	✓	✓	GLM-4V	29.7	63.4
InstructNav [43]	✓	✓	Linguistic	20.9	58.0
VLN-Game [44]	✓	✓	CLIP	26.9	66.7
Ours (FMM, w/o Surveillance)	✓	✓	CLIP	26.1	62.7
Ours (FMM, with Surveillance)	✓	✓	CLIP	34.5	67.4
Ours (SP, w/o Surveillance)	✓	✓	CLIP	28.0	68.7
Ours (SP, with Surveillance)	✓	✓	CLIP	**36.4**	**71.1**

and noise is removed using DBSCAN with an epsilon value of 0.05 and a minimum point count of 10. All point cloud operations are implemented using Open3D.

B. Comparison with Single-Robot Navigation Methods

Comparison with Single-Robot Navigation Methods. Table I summarizes the performance of our method on the HM3D dataset. The key advantage of SurveilNav lies in its collaborative mechanism between surveillance cameras and the robot, which significantly improves navigation compared to single-robot systems. With the Shortest Path (SP) planner, our method achieves an SPL of 36.4 and an SR of 71.1, while the Fast Marching Method (FMM) attains 34.5 SPL and 67.4 SR. These results outperform the best baseline, MCoCoNav (29.7 SPL, 63.4 SR), by +6.7 in SPL and +7.7 in SR, and also exceed VLN-Game.

Impact of Surveillance Inputs. Removing surveillance inputs leads to a clear performance drop. For FMM, the SPL/SR decreases from 34.5/67.4 to 26.1/62.7, while SP drops from 36.4/71.1 to 28.0/68.7. These results highlight the effectiveness of surveillance in enhancing both navigation efficiency and success. Surveillance cameras provide a global and static view of the environment, while the robot contributes local sensing and active exploration. Together, they yield complementary strengths that enable more robust and efficient navigation.

Visualization. We visualize the navigation process of SurveilNav in Figure 5. As shown in Figure 5, the robot (green marker) cannot detect the target due to occlusion by walls. However, as illustrated in subfigure (b), Surveillance Camera 1 directly observes the target (couch), allowing the system to set the detected location as the navigation goal. This reduces unnecessary exploration and improves navigation efficiency. Furthermore, in subfigure (d), the area containing the couch is assigned a high value, demonstrating the effectiveness of our semantic reasoning mechanism.

C. Surveillance Configuration Analysis.

TABLE II: Impact of the Surveillance Perception Range.

Surveillance Setup	**Perception Range**	**SPL**	**SR**
Camera	180°	32.1	66.8
Panoramic Camera	360°	34.5	67.4

TABLE III: Impact of the Surveillance Camera Density

Surveillance Setup	**Coverage Density**	**SPL**	**SR**
Sparse Coverage	$200m^2$/camera	33.0	66.5
Dense Coverage	$100m^2$/camera	34.5	67.4

Impact of the Surveillance Camera Perception Range. In Table II, we compare the performance of cameras with different perception ranges. We observe that a standard camera with a 180° perception range achieves an SPL of 32.1 and an SR of 66.8, while a panoramic camera with a 360° perception range significantly improves performance, achieving an SPL of 34.5 and an SR of 67.4. This demonstrates that a wider perception range enhances environmental awareness, enabling more accurate navigation and better obstacle avoidance.

Impact of the Surveillance Camera Density. In Table III, we evaluate the effect of camera density on navigation performance. Under sparse coverage ($200m^2$ per camera), the system achieves an SPL of 33.0 and an SR of 66.5, while dense coverage ($100m^2$ per camera) improves performance to an SPL of 34.5 and an SR of 67.4. This indicates that higher camera density provides more comprehensive environmental coverage, leading to more reliable navigation. Notably, even with dense surveillance, performance bottlenecks persist. This is primarily due to structural occlusions and sub-optimal camera placements that preclude a global viewpoint, as well as erroneous detections from the object perception module which limit further performance improvements.

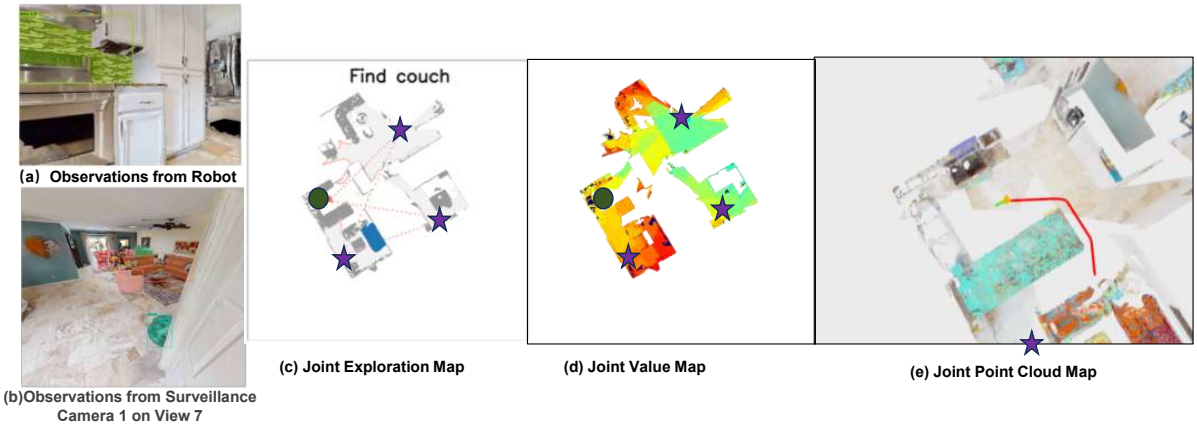

Fig. 5: The visualization of collaborative navigation in the habitat simulator. Figure (a) and Figure (b) depict the robot's first-person view observations and the observations from the 7th view of surveillance camera 1, respectively. Figure (c) illustrates the joint exploration map of the robot and all surveillance cameras, with the area representing the target region. Figures (d) and (e) display the joint exploration value and the joint point cloud map, respectively. In these figures, red areas denote high-value regions, while in Figure (e), the green lines represent the trajectory already traversed, and the red lines indicate the planned trajectory. The purple pentagram and the green circle represent the positions of the surveillance camera and the robot, respectively.

D. Ablation study of key components.

Impact of Joint Value Map. To evaluate the effectiveness of different value map construction strategies, we compare greedy search, object-centric, and region-centric approaches, as shown in Table IV. The greedy search baseline, which lacks semantic understanding, achieves an SR of 0.690 and an SPL of 0.356. The object-centric approach, leveraging CLIP features from detected objects, improves performance slightly with an SR of 0.705 and an SPL of 0.358. However, the region-centric approach, which aggregates semantic information across regions using CLIP, achieves the best results with an SR of 0.711 and an SPL of 0.364. This demonstrates that reasoning about broader regions, rather than individual objects, provides a more comprehensive understanding of the environment, leading to more efficient and successful navigation. These results highlight the importance of incorporating region-level semantic information in value map construction for visual navigation tasks.

TABLE IV: Impact of Joint Value Map.

VLM	Strategy	SPL	SR
None	Greedy	35.6	69.0
CLIP	Object-centric	35.8	70.5
CLIP	Region-centric	36.4	71.1

TABLE V: Impact of Object Detector.

Component	Variant	SPL	SR
Detector	YOLO-World	32.65	59.0
Detector	GroundingDINO	34.53	67.4

Impact of Object Detector. As shown in Table V, we compare two detectors, YOLO-World and GroundingDINO. GroundingDINO achieves superior performance with an SR of 67.4 and an SPL of 34.53, significantly outperforming YOLO-World, which achieves an SR of 59.00 and an SPL of 32.65. This improvement underscores the importance of using advanced detection models like GroundingDINO, which provide more accurate and semantically rich object detections, thereby enhancing the overall navigation system. Together, these results demonstrate that combining region-centric value map construction with a high-quality object detector offers the most effective framework for robust visual navigation tasks.

VI. Conclusions and Limitations

In this work, we introduced SurveilNav , a novel collaborative object navigation framework that leverages both robot egocentric observations and multi-view surveillance inputs. We also constructed the first dataset for robot–surveillance collaboration, spanning 36 scenes, 74 floors, and 206 cameras, providing a systematic benchmark for multi-view navigation. Extensive experiments on HM3D demonstrated that SurveilNav achieves state-of-the-art performance in large-scale and multi-floor environments, significantly improving both exploration efficiency and navigation success rate. Ultimately, SurveilNav highlights the potential of infrastructure-guided embodied intelligence, opening new directions for real-world applications such as large-scale search, household assistance, and rescue missions.

Limitations and Future Work. A primary constraint is the reliance on precise camera poses for alignment; future research could explore pose-free alignment or visual SLAM to enhance scalability. Furthermore, extending the framework to dynamic environments via active adjustments of surveillance camera FOV and orientation would enable real-time tracking of moving targets. Finally, investigating feature-level compression and adaptive transmission strategies will be crucial to mitigate communication bandwidth and latency during real-world deployment.

VII. ACKNOWLEDGEMENTS

This research is supported by Artificial Intelligence-National Science and Technology Major Project (2023ZD0121200), the National Natural Science Foundation of China (62437001, 62436001, 62441617), the Strategic Priority Research Program of Chinese Academy of Sciences under Grant XDB1350103, the Beijing Natural Science Foundation (L252146), and the Key Research Development Program of Jiangsu Province under Grant BE2023016-3.

REFERENCES

[1] S. Y. Gadre, M. Wortsman, G. Ilharco, L. Schmidt, and S. Song, "Cows on pasture: Baselines and benchmarks for language-driven zero-shot object navigation," in *CVPR*, 2023, pp. 23 171–23 181.

[2] K. Zhou, K. Zheng, C. Pryor, Y. Shen, H. Jin, L. Getoor, and X. E. Wang, "Esc: Exploration with soft commonsense constraints for zero-shot object navigation," in *ICML*, 2023, pp. 42 829–42 842.

[3] B. Yu, H. Kasaei, and M. Cao, "L3mvn: Leveraging large language models for visual target navigation," in *IROS*, 2023, pp. 3554–3560.

[4] N. Yokoyama, S. Ha, D. Batra, J. Wang, and B. Bucher, "Vlfm: Vision-language frontier maps for zero-shot semantic navigation," in *ICRA*, 2024, pp. 42–48.

[5] P. Wu, Y. Mu, B. Wu, Y. Hou, J. Ma, S. Zhang, and C. Liu, "Voronav: Voronoi-based zero-shot object navigation with large language model," *arXiv preprint arXiv:2401.02695*, 2024.

[6] Y. Li, D. Ma, Z. An, Z. Wang, Y. Zhong, S. Chen, and C. Feng, "V2x-sim: Multi-agent collaborative perception dataset and benchmark for autonomous driving," *IEEE robotics and automation letters*, vol. 7, no. 4, pp. 10 914–10 921, 2022.

[7] R. Xu, H. Xiang, X. Xia, X. Han, J. Li, and J. Ma, "Opv2v: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication," in *ICRA*, 2022, pp. 2583–2589.

[8] H. Yu, Y. Luo, M. Shu, Y. Huo, Z. Yang, Y. Shi, Z. Guo, H. Li, X. Hu, J. Yuan *et al.*, "Dair-v2x: A large-scale dataset for vehicle-infrastructure cooperative 3d object detection," in *CVPR*, 2022, pp. 21 361–21 370.

[9] E. Wijmans, A. Kadian, A. Morcos, S. Lee, I. Essa *et al.*, "Dd-ppo: Learning near-perfect pointgoal navigators from 2.5 billion frames," *arXiv preprint arXiv:1911.00357*, 2019.

[10] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik *et al.*, "Habitat: A platform for embodied ai research," in *ICCV*, 2019, pp. 9339–9347.

[11] D. S. Chaplot, D. P. Gandhi, A. Gupta, and R. R. Salakhutdinov, "Object goal navigation using goal-oriented semantic exploration," *NeurIPS*, vol. 33, pp. 4247–4258, 2020.

[12] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi, "Target-driven visual navigation in indoor scenes using deep reinforcement learning," in *ICRA*, 2017, pp. 3357–3364.

[13] D. Batra, A. Gokaslan, A. Kembhavi, O. Maksymets, R. Mottaghi, M. Savva, A. Toshev, and E. Wijmans, "Objectnav revisited: On evaluation of embodied agents navigating to objects," *arXiv preprint arXiv:2006.13171*, 2020.

[14] J. Krantz, E. Wijmans, A. Majumdar, D. Batra, and S. Lee, "Beyond the nav-graph: Vision-and-language navigation in continuous environments," in *ECCV*, 2020, pp. 104–120.

[15] J. Zhang, K. Wang, R. Xu, G. Zhou *et al.*, "Navid: Video-based vlm plans the next step for vision-and-language navigation," *arXiv preprint arXiv:2402.15852*, 2024.

[16] D. Zheng, S. Huang, L. Zhao, Y. Zhong, and L. Wang, "Towards learning a generalist model for embodied navigation," in *CVPR*, 2024, pp. 13 624–13 634.

[17] Y. Mei, Y. Yang, L. Guo, Q. Wang *et al.*, "Urbannav: Learning language-guided urban navigation from web-scale human trajectories," *arXiv preprint arXiv:2512.09607*, 2025.

[18] R. Ramrakhya, E. Undersander, D. Batra, and A. Das, "Habitat-web: Learning embodied object-search strategies from human demonstrations at scale," in *CVPR*, 2022, pp. 5173–5183.

[19] J. Zhang, K. Wang, S. Wang, M. Li, H. Liu, S. Wei, Z. Wang, Z. Zhang, and H. Wang, "Uni-navid: A video-based vision-language-action model for unifying embodied navigation tasks," *arXiv preprint arXiv:2412.06224*, 2024.

[20] K.-H. Zeng, Z. Zhang, K. Ehsani, R. Hendrix, J. Salvador, A. Herrasti, R. Girshick, A. Kembhavi, and L. Weihs, "Poliformer: Scaling on-policy rl with transformers results in masterful navigators," *arXiv preprint arXiv:2406.20083*, 2024.

[21] M.-M. Yu, F. Zhu, W. Liu, Y. Yang, Q. Wang, W. Wu, and J. Liu, "C-nav: Towards self-evolving continual object navigation in open world," *arXiv preprint arXiv:2510.20685*, 2025.

[22] Y. Kuang, H. Lin, and M. Jiang, "Openfmnav: Towards open-set zero-shot object navigation via vision-language foundation models," in *Findings of the Association for Computational Linguistics: NAACL 2024*, 2024, pp. 338–351.

[23] X. Sun, L. Liu, H. Zhi, R. Qiu, and J. Liang, "Prioritized semantic learning for zero-shot instance navigation," in *ECCV*, 2024, pp. 161–178.

[24] M.-M. Yu, Y. Chen, B. F. Karlsson, and W. Wu, "Ranger: A monocular zero-shot semantic navigation framework through contextual adaptation," *arXiv preprint arXiv:2512.24212*, 2025.

[25] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[26] J. Bai, S. Bai, S. Yang, S. Wang, S. Tan, P. Wang, J. Lin, C. Zhou, and J. Zhou, "Qwen-vl: A frontier large vision-language model with versatile abilities," *arXiv preprint arXiv:2308.12966*, 2023.

[27] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan *et al.*, "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.

[28] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *ICML*. PmLR, 2021, pp. 8748–8763.

[29] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *ICCV*, 2021, pp. 9650–9660.

[30] M. F. Ginting, S.-K. Kim, D. D. Fan, M. Palieri, M. J. Kochenderfer, and A.-a. Agha-Mohammadi, "Seek: Semantic reasoning for object goal navigation in real world inspection tasks," *arXiv preprint arXiv:2405.09822*, 2024.

[31] M. Chang, T. Gervet, M. Khanna, S. Yenamandra, D. Shah, S. Y. Min, K. Shah, C. Paxton, S. Gupta, D. Batra *et al.*, "Goat: Go to any thing," *arXiv preprint arXiv:2311.06430*, 2023.

[32] I. D. Miller, F. Cladera, T. Smith, C. J. Taylor, and V. Kumar, "Stronger together: Air-ground robotic collaboration using semantics," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9643–9650, 2022.

[33] Q. Chen, S. Tang, Q. Yang, and S. Fu, "Cooper: Cooperative perception for connected autonomous vehicles based on 3d point clouds," in *ICDCS*, 2019, pp. 514–524.

[34] E. Arnold, M. Dianati, R. De Temple, and S. Fallah, "Cooperative perception for 3d object detection in driving scenarios using infrastructure sensors," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 3, pp. 1852–1864, 2020.

[35] T.-H. Wang, S. Manivasagam, M. Liang, B. Yang, W. Zeng, and R. Urtasun, "V2vnet: Vehicle-to-vehicle communication for joint perception and prediction," in *ECCV*, 2020, pp. 605–621.

[36] Z. Song, F. Wen, H. Zhang, and J. Li, "A cooperative perception system robust to localization errors," in *2023 IEEE Intelligent Vehicles Symposium (IV)*, 2023, pp. 1–6.

[37] K. Yadav, R. Ramrakhya, S. K. Ramakrishnan, T. Gervet *et al.*, "Habitat-matterport 3d semantics dataset," in *CVPR*, 2023, pp. 4927–4936.

[38] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao, "Depth anything v2," *NeurIPS*, vol. 37, pp. 21 875–21 911, 2024.

[39] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," in *ICCV*, 2023, pp. 4015–4026.

[40] J. A. Sethian, "A fast marching level set method for monotonically advancing fronts." *proceedings of the National Academy of Sciences*, vol. 93, no. 4, pp. 1591–1595, 1996.

[41] Z. Shen, H. Luo, K. Chen, F. Lv, and T. Li, "Enhancing multi-robot semantic navigation through multimodal chain-of-thought score collaboration," in *AAAI*, vol. 39, no. 14, 2025, pp. 14 664–14 672.

[42] B. Yu, H. Kasaei, and M. Cao, "Co-navgpt: Multi-robot cooperative visual semantic navigation using large language models," *arXiv preprint arXiv:2310.07937*, 2023.

[43] Y. Long, W. Cai, H. Wang, G. Zhan, and H. Dong, "Instructnav: Zero-shot system for generic instruction navigation in unexplored environment," *arXiv preprint arXiv:2406.04882*, 2024.

[44] B. Yu, Y. Liu, L. Han, H. Kasaei, T. Li, and M. Cao, "Vln-game: Vision-language equilibrium search for zero-shot semantic navigation," *arXiv preprint arXiv:2411.11609*, 2024.