FalconEye: A Prompt-Guided Vision–Language Intelligent Tracking Robot

P.Varun Sai
Department of Computer Science and Engineering
Keshav Memorial Institute of Technology, Hyderabad, India
varunsaip2005@gmail.com

December 2025

Abstract

This paper presents *FalconEye*, a prompt-guided vision–language intelligent tracking robot for real-time object following in dynamic environments. The proposed system enables target specification through three modalities: spatial clicks, reference images, and natural language descriptions. Click-based prompts are processed using the Segment Anything Model (SAM) to generate object masks, while reference-image and text-based prompts are handled using CLIPSeg, a vision–language segmentation model built upon the joint embedding space of the CLIP framework. These foundation models allow flexible object grounding without task-specific retraining.

The segmented target is converted into a bounding box and tracked using DaSiamRPN, a distractor-aware Siamese tracking framework designed to maintain target identity under occlusion and reappearance. The tracked bounding box is further utilized for closed-loop motion control, enabling the mobile rover to regulate its heading and distance such that the target remains centered in the camera's field of view while maintaining a predefined separation. The proposed framework demonstrates stable and adaptive tracking behavior across diverse targets and environmental conditions, indicating the effectiveness of prompt-guided vision–language perception for autonomous robotic tracking.

1 Introduction

Robust object tracking is a fundamental capability for autonomous robotic systems operating in dynamic and unstructured environments. Tasks such as human following, mobile surveillance, and assistive robotics require a robot to continuously maintain target identity while adapting to changes in appearance, viewpoint, and scene context. Achieving stable tracking in real-world settings remains challenging due to occlusions, background clutter, and target re-identification after temporary disappearance.

Conventional visual tracking approaches primarily rely on appearance-based cues extracted from initial target observations. While such methods have shown strong performance in controlled scenarios, they often degrade when the target undergoes significant appearance variation or when visually similar distractors are present. Moreover, most traditional tracking pipelines lack semantic understanding of the target, limiting their ability to recover from drift or re-identify objects after occlusion.

Recent advances in foundation models have significantly improved visual perception by enabling open-vocabulary recognition and semantic grounding across diverse object categories. Vision–language models, in particular, allow objects to be specified using high-level semantic descriptions rather than fixed class labels. However, most existing vision–language approaches are designed for offline image understanding or static scene analysis, and their integration into real-time robotic tracking systems remains limited. Bridging semantic object specification with continuous, low-latency tracking and closed-loop robotic control poses practical challenges related to robustness, computational efficiency, and interaction design.

To address these limitations, this paper proposes *FalconEye*, a prompt-guided vision–language intelligent tracking robot that unifies semantic object specification, robust visual tracking, and autonomous motion control within a single framework. The proposed system enables flexible target definition through user-provided prompts, including spatial clicks, reference images, and natural language descriptions. By leveraging foundation segmentation models for object grounding and a distractor-aware Siamese tracker for temporal consistency, FalconEye achieves stable target tracking while remaining adaptable to varying target types and environmental conditions.

The main contributions of this work are summarized as follows:

- We propose a prompt-guided vision–language tracking framework that enables object specification through spatial clicks, reference images, and natural language descriptions.
- We integrate foundation segmentation models with a distractor-aware Siamese tracker to achieve robust target tracking under occlusion and reappearance.
- We develop a closed-loop robotic control strategy that utilizes tracked bounding box information to maintain target centering and distance during autonomous following.

2 Related Work

Visual object tracking has been extensively studied in the computer vision community, with early approaches relying on correlation filters and appearance-based matching to estimate target motion across frames. While such methods

offer computational efficiency, they often struggle in complex real-world scenarios involving background clutter, occlusion, and significant appearance variation. These limitations motivated the development of learning-based tracking frameworks that improve robustness by leveraging discriminative feature representations.

Recent advances in deep learning have led to the widespread adoption of Siamese network-based trackers, which learn a similarity function between a target template and candidate search regions. This paradigm has demonstrated strong performance in real-time tracking applications due to its balance between accuracy and efficiency. Extensions such as distractor-aware Siamese trackers further enhance robustness by explicitly mitigating interference from visually similar background objects and false positives. Despite these improvements, most Siamese tracking approaches assume a fixed target initialization and operate purely on visual appearance cues, without incorporating high-level semantic information about the target specified by the user.

In parallel, vision–language models have emerged as a powerful tool for semantic grounding and open-vocabulary perception. By embedding visual and textual information into a shared representation space, these models enable flexible object specification through natural language descriptions or reference images, rather than predefined class labels. Such approaches have shown strong results in image-level understanding and segmentation tasks; however, their integration into real-time robotic tracking systems remains limited. Existing works often focus on static images or offline processing and do not address the challenges of continuous tracking, robustness under occlusion, and closed-loop control required for autonomous robotic operation.

The proposed FalconEye system bridges these research directions by combining prompt-guided semantic object specification with a distractor-aware Siamese tracking framework and real-time robotic control. By integrating foundation vision–language segmentation models with high-speed tracking and closed-loop motion control, FalconEye enables flexible and robust object following while remaining suitable for deployment on mobile robotic platforms.

3 System Architecture

Figure 1 illustrates the overall system architecture of the proposed target-following rover. The system is divided into a base station and an onboard computing unit deployed on a Jetson AGX Xavier. The base station provides a user interface for target specification through visual clicks or semantic queries, while all perception, tracking, decision-making, and control components execute onboard the rover. Onboard the rover, perception and tracking subsystems operate independently to provide target hypotheses and continuous target state estimates. A high-level decision module implemented in Python integrates these outputs to generate motion commands. Low-level actuation is handled by a deterministic real-time controller implemented in C++, ensuring safe execution of velocity commands.

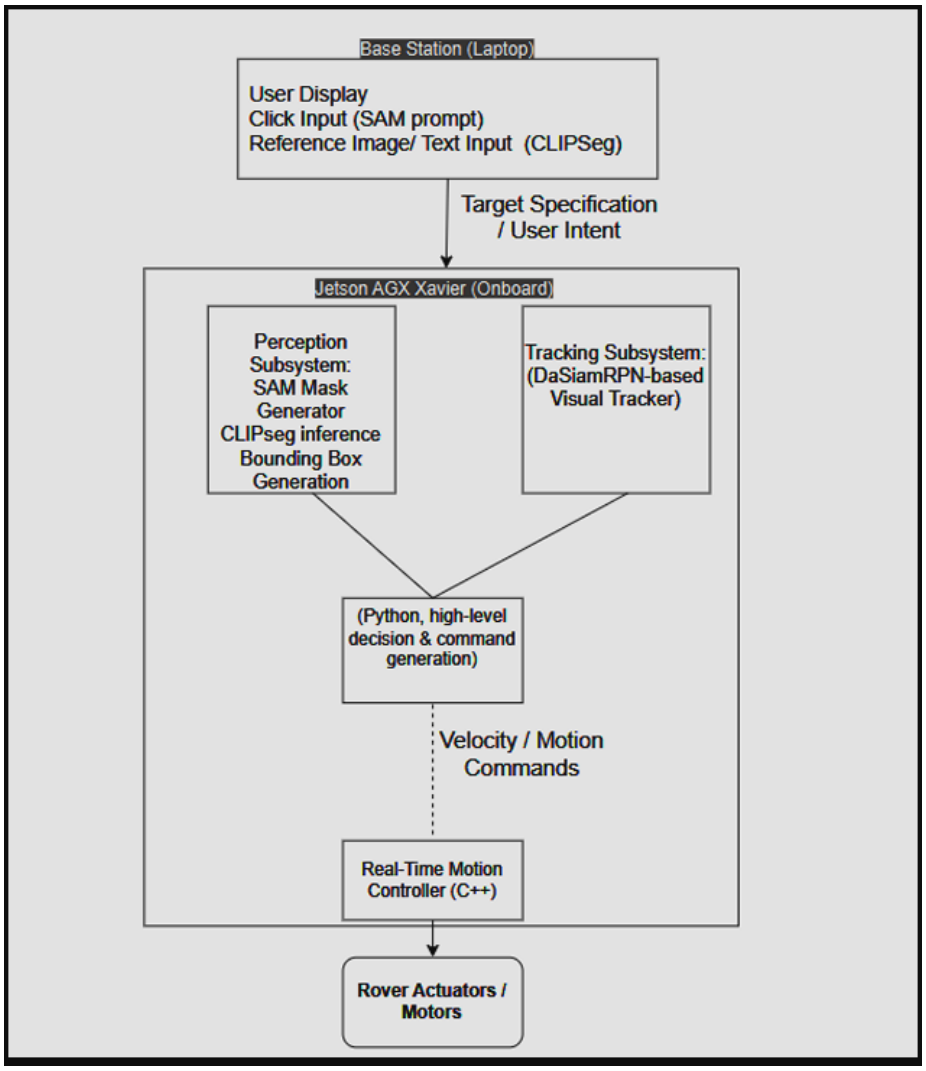

Figure 1: System architecture of the proposed rover platform, showing the separation between base-station user interaction, onboard perception and tracking, high-level decision-making, and real-time motion control.

3.1 Perception and Tracking Pipeline

This section describes the algorithmic workflow used for target acquisition and continuous tracking within the FalconEye system. The pipeline transforms user-specified intent into a persistent target state estimate suitable for real-time robotic control.

Target specification is initiated through one of three input modalities: spatial

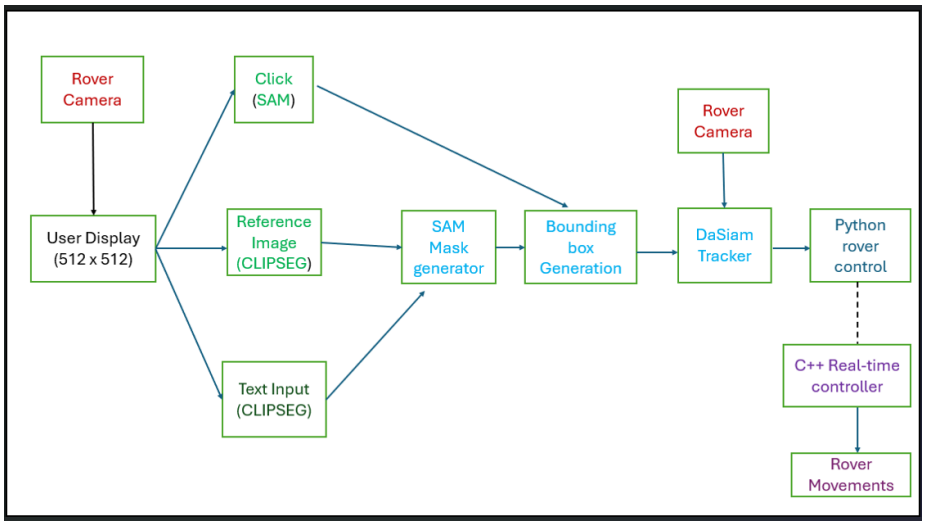

Figure 2: Perception and tracking pipeline of the FalconEye system. User prompts are grounded into object masks using foundation segmentation models, converted into bounding boxes, and tracked over time to provide continuous target state estimates.

clicks, reference images, or natural language descriptions. Click-based prompts are processed using the Segment Anything Model (SAM) to obtain an initial object mask, while reference-image and text-based prompts are handled using CLIPSeg, which performs vision–language-based semantic segmentation. The resulting segmentation mask is converted into a bounding box representation that initializes the visual tracking module.

Following initialization, the target is tracked across successive frames using a distractor-aware Siamese network-based tracker. The tracker produces continuous estimates of the target's spatial location, which are forwarded to the decision-making and control modules for closed-loop rover operation.

4 Methodology

This section describes the core components of the FalconEye system, detailing the prompt-guided object specification mechanism, vision–language-based segmentation, distractor-aware tracking strategy, and the closed-loop rover control framework.

4.1 Prompt-Guided Object Specification

FalconEye enables flexible target specification through three user-driven prompt modalities: spatial clicks, reference images, and natural language descriptions.

These prompt mechanisms allow the user to define the target object without relying on predefined object classes or task-specific retraining.

In the click-based mode, the user selects a spatial location corresponding to the target object within the camera view. This spatial prompt provides coarse localization, which is subsequently refined through segmentation. In reference-image and text-based modes, the user specifies the target using either an example image or a semantic description, enabling concept-level object specification. This prompt-guided design allows FalconEye to generalize across diverse target types and environments while maintaining a consistent interaction interface.

4.2 Vision–Language Segmentation

To convert user prompts into precise object representations, FalconEye employs foundation segmentation models capable of grounding visual and semantic information into pixel-level masks. Click-based prompts are processed using the Segment Anything Model (SAM), which generates segmentation masks based on sparse spatial cues. Reference-image and text-based prompts are handled using CLIPSeg, a vision–language segmentation model built upon the joint embedding space of CLIP.

These models enable open-vocabulary object segmentation by aligning visual features with semantic concepts, allowing objects to be specified beyond fixed category labels. The resulting segmentation mask is post-processed to remove noise and converted into a bounding box representation. This bounding box serves as a compact and efficient initialization for the downstream tracking module.

4.3 Distractor-Aware Visual Tracking

Following initialization, the target object is tracked across successive video frames using a distractor-aware Siamese network-based tracker. Specifically, FalconEye employs DaSiamRPN, which estimates target location by learning a similarity function between a reference template and candidate search regions. The distractor-aware design improves robustness by suppressing interference from visually similar background objects and false positives.

The tracker operates in real time and maintains target continuity under moderate appearance variation and short-term occlusion. High frame-rate operation further reduces the likelihood of tracking failure by limiting inter-frame motion and appearance changes. The tracker outputs an updated bounding box for each frame, which is directly used by the control module for motion regulation.

4.4 Closed-Loop Rover Control

The FalconEye control module translates tracking outputs into motion commands for the mobile rover using a closed-loop feedback strategy. The center coordinates and scale of the tracked bounding box are used to compute lateral

and longitudinal control signals. Horizontal deviation of the bounding box center from the image center determines steering adjustments, while bounding box size is used as a proxy for target distance.

Control logic is implemented using a hybrid software architecture, where high-level perception and tracking are executed in Python, and real-time motor control is handled by a C++ controller to ensure low-latency response. This separation allows efficient perception processing while maintaining stable and responsive rover motion. The control strategy ensures that the target remains centered within the camera's field of view and that a predefined separation distance is maintained during autonomous following.

5 Experiments

This section describes the experimental setup, evaluation scenarios, and performance metrics used to assess the effectiveness of the proposed FalconEye system. The experiments are designed to evaluate prompt-guided target specification, real-time tracking stability, and closed-loop rover following behavior in dynamic environments.

5.1 Hardware and Software Setup

Experiments were conducted on a mobile ground rover equipped with an onboard web camera for real-time visual perception. The camera captures video frames in its native format, which are subsequently converted to RGB representation within the perception pipeline to ensure compatibility with downstream vision and segmentation models.

Video frames from the camera were streamed to the perception module for prompt-based segmentation and tracking. The rover platform supports differential drive motion and enables continuous closed-loop control based on visual feedback.

The perception and tracking pipeline, including prompt handling, segmentation, and tracking, was implemented in Python. Real-time motor control and low-level actuation were handled by a C++ controller to ensure low-latency execution. The system was designed to operate at high frame rates, enabling responsive tracking and reducing the impact of rapid target motion and short-term occlusion.

5.2 Evaluation Scenarios

The FalconEye system was evaluated across multiple scenarios designed to test its robustness under different forms of user input and environmental conditions. Target objects were specified using three prompt modalities: spatial clicks, reference images, and natural language descriptions. Each modality was tested independently to assess prompt-guided initialization accuracy and subsequent tracking stability.

To evaluate tracking robustness, experiments included scenarios with background clutter, visually similar distractors, and partial occlusions. Additional tests involved varying target motion speed and direction to examine the system's ability to maintain target centering and distance during continuous rover following. All experiments were performed in real time, without offline processing or post hoc corrections.

5.3 Evaluation Metrics

System performance was evaluated using a combination of qualitative and quantitative metrics. Tracking stability was assessed based on the continuity of the tracked bounding box across frames and the system's ability to avoid target switching in the presence of distractors. Real-time performance was measured in terms of achieved frame rate during continuous operation.

Rover following behavior was evaluated by analyzing the deviation of the target from the center of the camera's field of view and the consistency of the maintained distance, estimated using bounding box scale. Qualitative observations were also used to assess responsiveness, smoothness of motion, and recovery from short-term occlusions.

6 Results

This section presents the experimental results obtained using the proposed FalconEye system. The results focus on prompt-guided initialization accuracy, tracking stability during continuous operation, real-time performance, and closed-loop rover following behavior.

6.1 Qualitative Demonstration

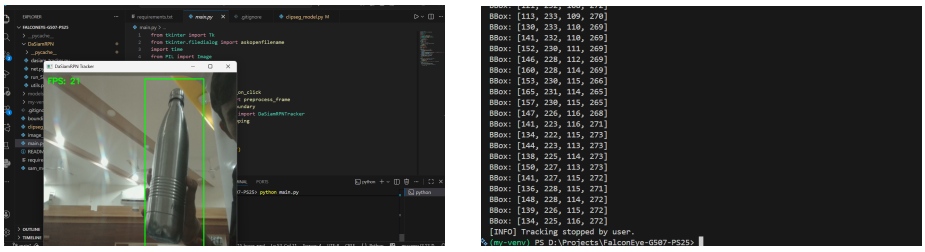

Figure 3: Qualitative demonstration of the FalconEye system. The left image shows prompt-guided target acquisition through segmentation, while the right image illustrates continuous visual tracking using the initialized bounding box during operation.

6.2 Prompt-Guided Initialization

The FalconEye system successfully initialized target tracking using all three supported prompt modalities: spatial clicks, reference images, and natural language descriptions. Click-based prompts enabled rapid and precise target initialization through spatial grounding, while reference-image and text-based prompts allowed semantic object specification without requiring predefined object categories.

Qualitative observations indicate that vision–language prompts provided flexible target selection across a wide range of objects and scenes. In all tested cases, the segmentation module produced coherent object masks that were suitable for initializing the downstream tracking process.

6.3 Tracking Stability and Robustness

The distractor-aware Siamese tracking module demonstrated stable target tracking across continuous video sequences, maintaining target identity in the presence of background clutter and visually similar distractors. High frame-rate operation contributed to reduced inter-frame target displacement, thereby lowering the likelihood of tracking drift. The system remained robust under moderate appearance variation and short-term occlusion, with the tracker maintaining continuity once the target reappeared in the camera view. While long-term target re-identification after complete tracking failure was not explicitly addressed, the observed results indicate reliable short-term robustness suitable for real-time robotic following tasks.

6.4 Real-Time Performance

The FalconEye system operated at an average frame rate of approximately 40 frames per second during continuous tracking and rover following experiments. This performance exceeds the real-time requirements for closed-loop robotic control and enables smooth and responsive target following behavior.

High frame-rate operation significantly reduced inter-frame target displacement, thereby lowering the likelihood of tracking failure due to rapid motion or short-term occlusion. In addition, the closed-loop control strategy actively maintained the target near the center of the camera's field of view, further minimizing occlusion events during operation. These factors jointly contributed to stable and reliable tracking performance in dynamic environments.

6.5 Rover Following Behavior

Closed-loop control based on tracked bounding box information enabled the rover to maintain consistent target centering and distance during autonomous following. Lateral control adjustments successfully corrected deviations in the horizontal position of the target, while bounding box scale served as an effective proxy for regulating longitudinal motion.

Qualitative evaluation showed smooth and responsive rover behavior across varying target motion patterns. The rover consistently adjusted its trajectory to follow the target while maintaining a predefined separation distance, demonstrating the effectiveness of the perception-driven control strategy.

7 Conclusion

This paper presented *FalconEye*, a prompt-guided vision–language intelligent tracking robot designed for real-time object following in dynamic environments. The proposed system integrates flexible user-driven target specification with vision–language segmentation, distractor-aware visual tracking, and closed-loop robotic control within a unified framework. By supporting spatial clicks, reference images, and natural language descriptions, FalconEye enables intuitive and open-vocabulary object specification without task-specific retraining. Experimental results demonstrate that the system achieves stable and smooth tracking performance during continuous operation, maintaining real-time responsiveness at approximately 40 frames per second. The combination of high frame-rate tracking and closed-loop centering effectively reduces the likelihood of occlusion and tracking drift, enabling reliable autonomous following behavior on a mobile rover platform. The use of a distractor-aware Siamese tracker further contributes to robustness in cluttered environments and in the presence of visually similar objects.

While the current system focuses on short-term tracking robustness, future work will explore extensions for long-term target re-identification and recovery from complete tracking failure. Additional directions include adaptive control strategies, integration of multi-camera sensing, and evaluation across more diverse real-world scenarios. Overall, FalconEye demonstrates the effectiveness of combining prompt-guided vision–language perception with real-time tracking and control for practical robotic applications.

References

[1] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment Anything," *arXiv preprint arXiv:2304.02643*, 2023.

[2] T. Lüddecke and A. Ecker, "Image Segmentation Using Text and Image Prompts," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 7086–7096.

[3] Z. Zhu, Q. Wang, L. Bo, W. Wu, J. Yan, and W. Hu, "Distractor-Aware Siamese Networks for Visual Object Tracking," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.