FollowMe: Assessing Vision and Bluetooth Sensor Feasibility for Smartwatch-Guided Human-Following Robots

JULIAN RAHEEMA, University Of California San Diego, USA
CONNOR GAG, University Of California San Diego, USA
TYLER FLAR, University Of California San Diego, USA
HELENA BENDER, University Of California San Diego, USA

Human-following robots are increasingly vital in demanding fields such as construction, firefighting, search and rescue, and maritime operations, where individuals often need to carry heavy equipment. Existing systems, however, typically lack the capability to track users via wearable technology. To address this, we developed a wearable-controlled robot that combines Bluetooth and computer vision (CV) for real-time user tracking. Initial exploration of Bluetooth-based following, using Angle of Arrival (AoA) direction-finding, aimed to estimate the leader's pose when out of the robot's view. However, this method proved unreliable due to significant data inaccuracies. As a more effective alternative, we implemented a vision-based tracking system using a depth camera. YOLOv8 was used for person detection, with MediaPipe and OSNet providing robust feature extraction and re-identification. Traditional CV techniques, including color-based tracking, further enhanced pose estimation accuracy. We also successfully integrated smartwatch control, enabling intuitive user interaction with the robot. This hybrid system allows the robot to follow the user reliably in complex environments. In conclusion, while Bluetooth tracking showed limitations, computer vision provided a feasible and accurate solution for autonomous human-following, with smartwatch control offering an added layer of usability. Future work will focus on refining sensor fusion and model performance.

Additional Key Words and Phrases: Robot, Follower, Autonomous, HRI, Collaborative

1 INTRODUCTION

Robots that are capable of following humans have the potential to significantly enhance various aspects of daily life, professional work, and emergency response operations. The concept of deploying robots for tasks that are dirty, dull, or dangerous—often referred to as the "3Ds" of robotics—is well-established and widely supported within the robotics community [9]. From bomb disposal and hazardous material inspection to delivering first-aid kits in search and rescue missions, the application of autonomous followers in high-risk environments is both compelling and impactful.

To date, most research and commercial implementations of follower robots have focused on small-scale platforms or drones. For example, systems like Skydio have demonstrated autonomous visual tracking in aerial drones for tasks such as cinematography and inspection [19]. In the domain of ground robotics, follower behavior has often been explored in structured environments like airports, where wheeled robots assist passengers [16]. However, few systems have explored the use robots for user-following in unstructured or disaster environments [8]. Current search and rescue robots are typically teleoperated, requiring human operators to focus on navigation and control. This divided attention reduces the operator's ability to concentrate on high-level tasks such as situational assessment or victim identification. A robot capable of autonomously following a human operator—while equipped with perception sensors and payload capacity—would greatly enhance mission efficiency by allowing the operator to focus on complex, cognitively demanding tasks. Moreover, a legged robot offers mobility advantages in rough terrain, where wheeled or tracked platforms struggle. By autonomously following a responder through such environments, the robot can serve as both a mobile sensor platform and a load-carrying assistant, increasing the effectiveness of field operations and reducing physical strain on human team members. Despite the growing adoption of robotic technologies in industrial and service sectors, there remains a lack of user-following robots designed for large-scale, legged platforms capable of operating in disaster response scenarios. In this paper, we present a feasibility study of the FollowMe system—an approach that leverages both vision and Bluetooth technologies to enable robust user-following capabilities in legged robots. Specifically, the system utilizes visual tracking when the human leader remains within the field of view of the onboard camera, and seamlessly switches to Bluetooth-based localization when the leader moves out of the camera's frame. This dual-modality approach aims to enhance reliability and extend the operational range of the robot in complex and unstructured environments.

1.1 Challenges

Like many development projects, ours faced a number of challenges during implementation. One of the first was getting reliable communication between the smartwatch and the computer. This involved learning how the connection protocols worked and properly configuring the XML file in the smartwatch app to send secure, device-specific JSON commands. Another issue was getting an accurate BLuetooth position fix from just one antenna. Ideally two or more spatially-seperated locators would let us triangulate the device's position. Getting a robot to follow a human leader reliably is itself a major challenge. The system has to track the correct person, adapt to different lighting conditions, adjust its speed based on the leader's movement, and most importantly, navigate safely in real time without hitting obstacles or the person it's following. Achieving this requires solving problems across several areas—computer vision, real-time processing, motion control, and sensor integration. Integrating the system into Spot's autonomy framework also came with its own difficulties. We had to install many dependencies while keeping the robot's built-in functions working correctly. On top of that, we needed to tune motion commands, adjust parameters in the proportional controller, and improve the vision system to get more accurate detection and tracking.

2 RELATED WORKS

Our research consists of four main parts. First, we developed a smartwatch-controlled robot to allow command and control through wearable devices. Second, we studied the feasibility of using Bluetooth for a robot follower system, focusing on communication range and stability. Third, we tested the use of a depth camera to enable the robot to follow a target based on visual input. Finally, we designed and integrated a controller for the follower robot to coordinate its movement with the leader robot using proportinal control.

2.1 Smartwatch-Based Control Interface

Recent advancements in wearable technology have led to numerous efforts to enable smartwatch-based control of robots, particularly in human-robot interaction (HRI) contexts. Notably, systems such as iRoCo [22] utilize motion-based control by estimating the user's arm pose through Differentiable Ensemble Kalman Filters, enabling robot teleoperation via smartwatch and smartphone fusion. While powerful, such systems require continuous motion tracking, calibration, and substantial computational resources, limiting their accessibility and robustness in noisy or constrained environments. Similarly, WearMoCap [21] supports multimodal motion tracking using smart devices across various body placements. However, it suffers from sensor drift, sensitivity to device placement, and potential user discomfort. Another related approach, presented in [23], uses machine learning to estimate arm pose from smartwatch data alone, achieving good accuracy but at the cost of algorithmic complexity and real-time constraints.

In contrast, our system simplifies the control interface by leveraging a button-based application on an Android smartwatch, communicating discrete JSON-formatted commands via HTTP POST over Wi-Fi to a ROS 1-enabled [11] robot. This method eliminates the need for motion tracking entirely, reducing system complexity and making it inherently more stable and user-friendly. Our interface is also designed with accessibility in mind, incorporating color-blind-friendly design elements to ensure inclusive. By focusing on reliable, repeatable commands through tactile input, our approach addresses key limitations in prior work, complexity, sensitivity to sensor noise, and accessibility, making it particularly suitable for educational, assistive, and field robotics applications.

2.2 Bluetooth-based Follower

Bluetooth-based relative localization has gained traction as a lightweight and low-power alternative for human-following robot systems, particularly in indoor environments were GPS is limited and not available. Prior studies that rely on infrastructure-based localization using multiple synchronized receivers, such as Cominelli et al. [2], our setup uses only one receiver to perform Angle of Arrival (AoA) estimation. This single-receiver design simplifies deployment and avoids the need for fixed beacon infrastructure, making it suitable for mobile robots operating in ad hoc environments. However, our experimental results reveal significant limitations in accuracy and noise performance under real-world conditions. Despite the theoretical support for AoA and signal strength-based localization, the estimated positions diverged substantially from ground truth, achieving only $\tilde{5}$% accuracy in controlled trials. These findings are consistent with other RSSI- and BLE-based systems such as those by Pradeep et al. [10], who attempted to mitigate inaccuracies by combining RSSI with IMU data on smartphones. Similarly, Satan and Toth [14] demonstrated proximity-based indoor localization using filtered RSSI and log-distance path loss modeling, achieving reasonable room-level resolution but lacking angular precision. In contrast, more recent systems like the UWB-based human-following smart stroller by Zhang et al. [25] offer centimeter-level accuracy and robustness to interference but at the cost of greater power consumption and hardware complexity.

By focusing on a compact, infrastructure-free design, our system trades accuracy for portability and cost. We decided not to continue spending time to optimize the Bluetooth based follower, and moving forward using vision-based follower because of the limited time for this project for this class. However, these limitations motivate future work toward integrating filtering algorithms, additional inertial sensing, or multi-receiver augmentation to enhance tracking robustness in both indoor and semi-structured outdoor settings.

2.3 Camera-based Follower

Our approach to computer vision-based person tracking builds upon a variety of prior methodologies. The primary reference for our work is a comprehensive survey [20]. A range of modalities were examined, including RGB, depth, skeletal data, and infrared. Various methods for integrating these modalities were explored, and their relative efficacy was evaluated. Based on these findings, a multimodal sensor fusion approach utilizing RGB, depth, and skeleton tracking was selected. These approaches serve as a good foundation for person-specific tracking. Naturally, their effectiveness is maximized when individuals within the frame exhibit distinctions across the specified modalities [6].

We were confronted with the challenge of integrating features derived from multiple modalities of an individual to determine if they matched a target subject. Two primary fusion methods were considered: feature-level fusion and score-level fusion, each with inherent advantages and disadvantages [4]. Feature-level fusion entails concatenating feature vectors from each modality into a unified vector, which is subsequently used to compute a similarity score against the target using machine learning. This necessitates training a model to map the combined feature vector to a score. In contrast, score-level fusion involves extracting feature vectors for each modality and calculating individual modality-specific scores. A weighted average of these modality scores is then computed to produce the final score. We employed score-level fusion due to its greater flexibility for manual optimization and its avoidance of training a machine learning model, a process that would have required extensive labeled data and time, which were limited.

We used four main features: RGB, depth, skeleton information, and forearm color. For RGB and depth, we used OSNet because it maintains accuracy while still being very fast [26]. For skeleton tracking, we used MediaPipe to find the landmarks on each person and extract features from them that give us information about their pose [7]. To characterize the color of the user's forearm, we computed a color histogram from the segmented forearm region. This feature was developed for our application as we did not identify a similar method in the existing literature.

Furthermore, the widely employed Kalman Filter has been integrated into this program. This filter is frequently utilized in person re-identification algorithms [3]. Directional movement of the target person is continuously monitored, and subsequent location is predicted. This prediction is utilized to restrict the search area for potential targets. Specifically, given the current position and direction of movement, the target will only be sought within the predicted region in the subsequent frame. This methodology enhances target tracking efficacy during instances of temporary occlusion. The Kalman filter remains a robust recovery mechanism for tracking, demonstrating consistent algorithmic improvement in recent years [24].

Alternative approaches that were explored but ultimately discarded should be mentioned. Initially, person re-identification was approached using YOLOv8 and DeepSort for the detection and tracking of multiple individuals [27]. While we did still use YOLOv8, we did not continue to use DeepSort. The person re-identification in the examined system demonstrated adequate performance but exhibited limitations in handling occlusions and instances where the target exited the frame. Furthermore, the system frequently reassigned identifiers to individuals or generated new identifiers upon their reappearance within the frame. These deficiencies rendered the system unsuitable for our purposes, leading to the decision to develop a proprietary re-identification system instead of adopting the existing solution. Similar shortcomings were observed in evaluations of alternative models, such as MobileNetV2 [13]. These models exhibit substantial computational latency, and the current robotic platform lacks a GPU. Consequently, while potential enhancements to the DeepSort algorithm might have been feasible for our specific application, integrating additional functionalities would be difficult due to the performance overhead already imposed by the DeepSort implementation.

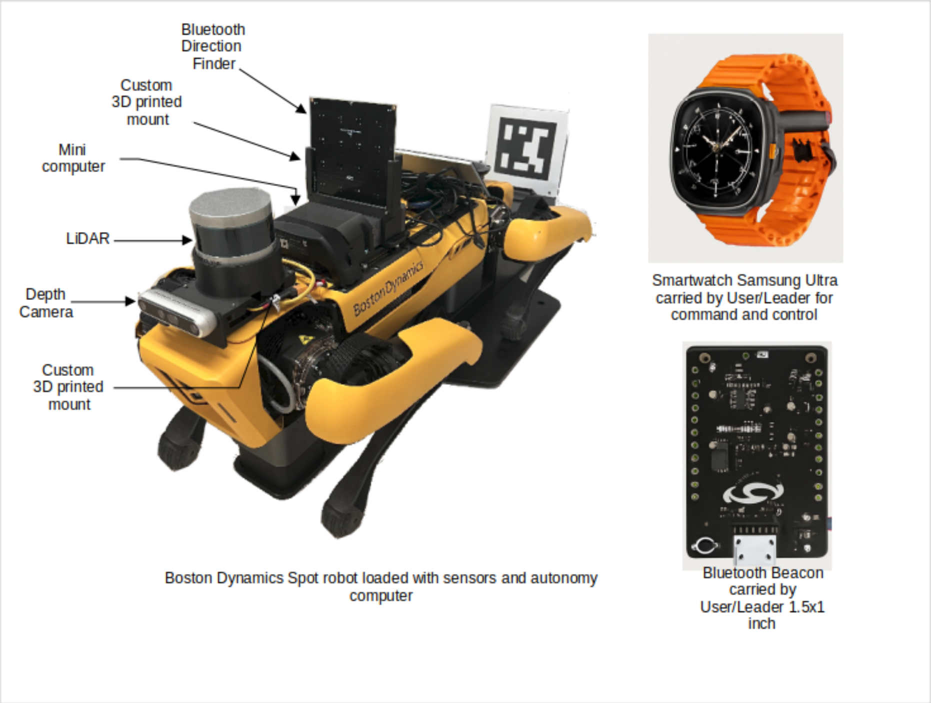
3 SYSTEM ARCHITECTURE

3.1 Hardware Architecture and System Components

In the FollowMe project as shown in the figure 1 , we employed a range of hardware components to enable robust and versatile human-following capabilities for a Spot [1] legged robot platform. The primary robotic platform used is the Boston Dynamics Spot, a quadruped robot known for its mobility and stability across varied terrain. To facilitate human-robot interaction and control, we used the Samsung Galaxy Watch Ultra, which allowed the user to issue commands and control the robot's behavior remotely. For visual tracking, we integrated an Intel RealSense D455 depth camera [5], enabling the robot to perform computer vision-based following by detecting and tracking the user in 3D space. To implement Bluetooth-based following, we utilized the BG22 Bluetooth Dual Polarized Antenna Array Pro Kit from Simplicity Studio [17], in combination with the EFR32BG22 Thunderboard Kit [18]. In this configuration, the user carries the Thunderboard, while the BG22 system is mounted on the robot. This setup enables the robot to estimate the relative distance and orientation of the user via Bluetooth signal analysis. All onboard computation is handled by an Intel NUC Mini-PC featuring an Intel Ultra 9 CPU with 22 cores (no GPU). This computer is mounted on the robot and processes all sensor inputs and follower algorithms in real time, transmitting commands directly to Spot's motion control interface. The computer is connected via WiFi to the same hotspot that the smartwatch is connected to. To enhance the reliability of the vision-based follower, we also used a pair of arm sleeve protectors in red and blue colors. These served as easily detectable visual markers to improve person tracking performance in varying lighting conditions.

3.2 Software Architecture and System Integration

The software framework as shown in the figure for the FollowMe project integrates multiple platforms to enable seamless interaction between sensing modules, user interfaces, and robotic control. The core of the system is built upon the Robot Operating System (ROS) Noetic running on Ubuntu 20.04, which facilitates communication between the robot, sensors, and high-level software modules. ROS was used to handle sensor data acquisition,



Boston Dynamics Spot robot loaded with sensors and autonomy computer

Smartwatch Samsung Ultra carried by User/Leader for command and control

Bluetooth Beacon carried by User/Leader 1.5x1 inch

Fig. 1. Hardware system

message passing, and actuation commands via the cmd_vel interface, a standard velocity command topic in mobile robotics.

For the development of the user interface and control layer, we used Android Studio to create a custom Android-based smartwatch application, which allows users to issue motion commands and initiate tracking modes remotely. The smartwatch communicates with the robot via a wireless channel, providing a portable and intuitive method for human-robot interaction.

To support Bluetooth-based localization, we utilized Simplicity Studio from Silicon Labs to program the EFR32BG22 Bluetooth Direction Finding Kit. This platform was essential for implementing the Bluetooth Angle of Arrival (AoA) positioning algorithm, enabling the robot to estimate the relative position of the user based on direction-finding capabilities.

Our system design builds upon prior work such as the Autonomous Exploration and Mapping Payload Integrated on a Quadruped Robot [12], which implemented full Simultaneous Localization and Mapping (SLAM), autonomous exploration, and navigation on a legged platform. While our architecture leverages similar foundational components—such as the use of ROS and real-time localization pipelines—the FollowMe system is modular and designed to be independent of any specific robot platform. It can be deployed on any robot compatible with ROS1 Noetic, provided it supports the necessary hardware interfaces and accepts standard velocity command messages.

This modularity ensures that the FollowMe system can be readily adapted for a variety of use cases in human-robot interaction, search and rescue, or field robotics, without being constrained to a single robotic architecture.

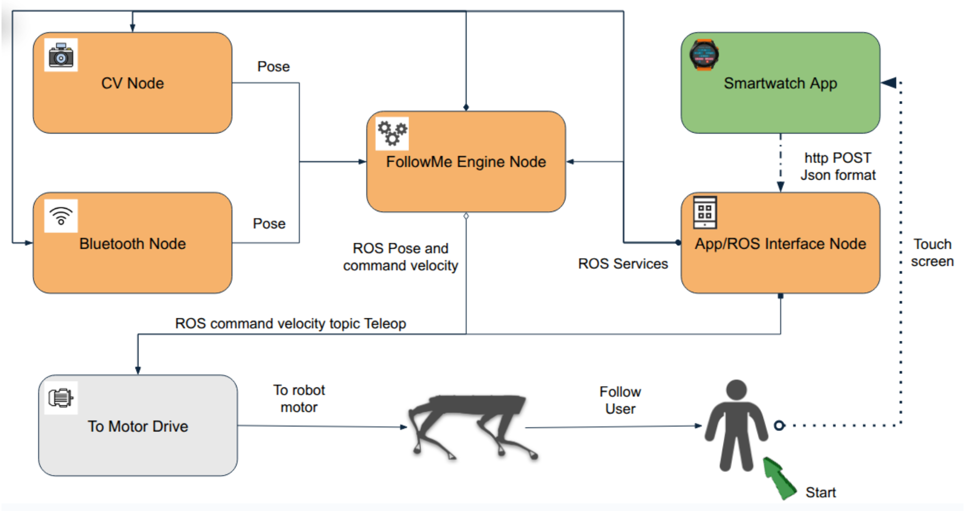


Fig. 2. System Architecture

4 METHODOLOGY

4.1 Interface Design

The smartwatch application features a user interface designed to prioritize intuitive flow, accessibility for color-blind users, and human decision-making. It was developed and built on Wear OS from Android Studio.

4.2 Layout

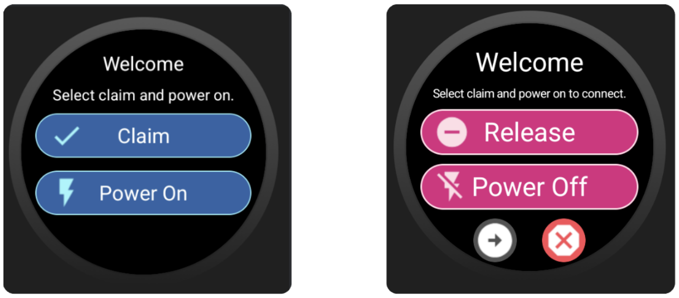
When the application is launched on a smartwatch device, the user is directed to the Welcome page to power on and claim the robot. From there, the user can be navigated to the Commands page, which offers basic and advanced commands. Basic commands include sit, stand, dock, and undock. Advanced commands include teleop mode and follow mode. In teleop mode, the user is sent to a new page that allows them to manually control the robot's movement in any direction and rotation. In follow mode, the user is sent to a new page that allows them to select QR Follower, which enables the robot to track a QR code, or CV Follower, which enables the robot to track the user via a computer vision based tracking algorithm. Two additional follow modes, Sensor Follower and Fusion Follower, are displayed and planned but have not been implemented.

An Emergency (E-Stop) option is available when the robot is claimed. It appears as a button and can be activated by the user at any time in the event of an emergency. The robot will automatically freeze and dismiss the previous action the user specified it to be in. Additionally, a Back option is avaliable once the robot has started. The user has the ability to press back button in the case of any different decisions.

4.3 Implementation

The user interacts with the robot via a interface on the smartwatch device that sends commands in JSON format using an HTTP POST request. The server, running on a designated IP address, receives the message and either calls a ROS service or publishes to specific ROS topics. The available services include: sit, stand, dock, undock, and acquisition. The topics that control robot motion include: up, down, right, left, and any rotation. When the "follow" service is triggered, the FollowMe engine begins listening to both the computer vision (CV) node and the Bluetooth node. It determines the target's position based on the selected following mode. Although only the

Vision mode is currently implemented, the system includes a pipeline for Bluetooth-based following, which is set up but not yet deployed. Once the target's position is identified, a lightweight proportional controller drives the robot toward the leader's position with the desired heading and speed. The robot continues to follow the user as long as the user remains in view and the camera can reliably detect them. Our design emphasizes keeping a human-in-the-loop approach to enhance safety and prioritize human judgment over autonomous decisions.



Select claim and power on.

Select claim and power on to connect.

Fig. 3. Welcome Interface

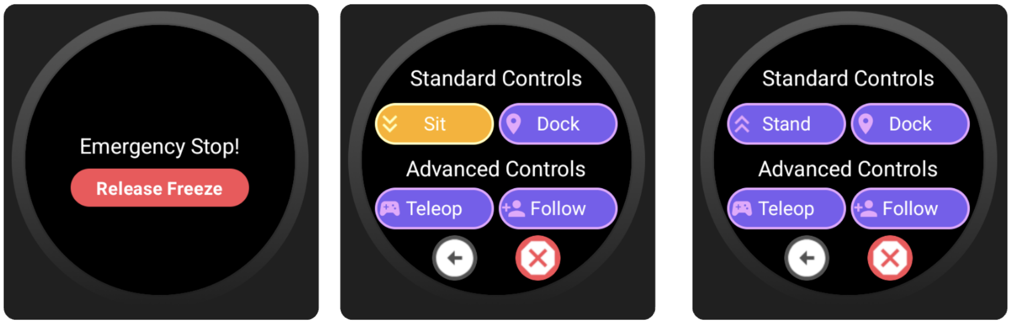
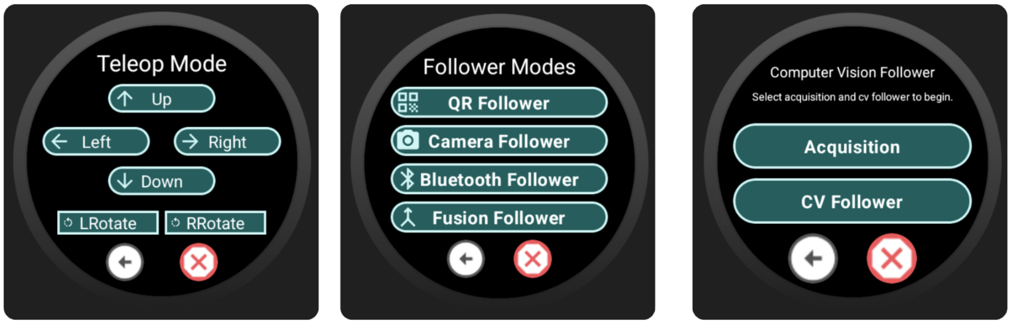


Fig. 4. Command Interface



Select acquisition and cv follower to begin.

Fig. 5. Advanced Command Interface

4.4 Bluetooth Follower

4.4.1 System Concept. A single locator—the Silicon Labs BG22 Bluetooth Dual Polarized Antenna Array Pro Kit (BRD4191A)—is equipped with a 4 × 4 dual-polarised uniform rectangular antenna (URA). It tracks one or more BLE tags whose target form factor is the Samsung Galaxy Watch Ultra or a similar device that uses its native Bluetooth radio; during development we emulated these Bluetooth tags with EFR32BG22 Thunderboard Kits (BRD4184A) because they offered easier firmware access while exhibiting comparable over-the-air behavior. The tag periodically appends a Constant-Tone Extension (CTE) to its advertising packets. Each CTE providfes a burst of phase-coherent IQ samples from which the locator estimates the tag's three-dimentional position at up to 50 Hz. These positions are relayed to the robot that executes a real-time "follow-me" behavior as discussed in Section 4.6.

4.4.2 Angle-of-Arrival (AoA) Measurement. The IQ samples form a spatial snapshot across the URA. After carrier-frequency compensation, the locator feeds this snapshot into a super-resolution MUSIC estimator. MUSIC searches for peaks in the spatial-spectrum function, yielding azimuth θ and elevation ϕ with sub-degree resolution [15]. A lightweight first-order IIR filter

$$\hat{\alpha}_k = (1-\beta)\hat{\alpha}_{k-1} + \beta_{\alpha_k}, \quad \beta = 0.6 \tag{1}$$

suppresses single-shot outliers.

4.4.3 Range Estimation from RSSI. Simultaneously, the locator measures the received signal strength indicator (RSSI) during the CTE. Averaging across all antenna elements suppresses small-scale fading. The range is obtained from the empirical log-distance model

$$\rho = 10\frac{A-r}{10\eta} \tag{2}$$

where r is the smoothed RSSI (dBm), A the reference loss at 1 m, and η the enviroment-specific path-loss exponent. Outlier rejection rules ignore RSSI jumps greater than 8 dB to preserve robustness in multipath conditions.

4.4.4 Position Synthesis. The AoA pair (θ, ϕ) and range ρ define a spherical coordinate in the locator frame. This is converted to Cartesian coordinates via

$$x = \rho \sin\phi \cos\theta, \tag{3}$$

$$y = \rho \sin\phi \sin\theta, \tag{4}$$

$$z = \rho \cos\phi. \tag{5}$$

This position data is what is given to the motion controller discussed in Section 4.6.

4.5 Vision-based Follower

Our system is designed for the specific application of tracking a single, designated individual within a robot's field of view. This focused approach provides a key advantage over generalized multi-person re-identification (Re-ID) systems, which must simultaneously track and differentiate all individuals in a scene. By concentrating on a single target, we can implement a proactive data acquisition stage to build a robust and discriminative feature profile of the person of interest before they are potentially obscured or confused with other individuals. The core of this single-subject focus is the Acquisition Stage, an initial phase dedicated to enrolling the target person. Upon initiation, the system captures data for a predefined duration, typically 10-15 seconds, while the target moves naturally within the frame. For each video frame processed, the system first uses a YOLOv8n object detector to ensure only one person is present. If more than one individual is detected, the frame is discarded to prevent ambiguous data collection. For the verified single person, a suite of feature extractors is then employed to generate high-dimensional embeddings for each active modality. This process is repeated for every valid frame, resulting in a rich, multi-modal dataset of feature samples, denoted as a set S, which forms the ground truth against which all future candidates are compared.

Once a target is enrolled, the system transitions to the Tracking Stage. In this phase, the primary objective is to consistently re-identify the target in every new frame. The process begins with candidate detection, where the YOLOv8n model identifies all individuals in the current frame. Each detected person is then processed by the same suite of feature extractors used during acquisition to generate a query feature set. Specifically, for RGB appearance features, we utilize the OSNet model to produce a 512-dimensional vector from the person's visual appearance. To capture 3D shape, the corresponding depth map is normalized and fed into the same OSNet model, yielding another 512-dimensional vector. For skeletal pose features, we use MediaPipe Pose to detect 2D body landmarks, which we use to calculate proportions of the person's body and put into a 61-dimensional vector. Finally, for forearm color features, MediaPipe landmarks are used to isolate the forearms, from which normalized 30-bin color histograms are computed and concatenated into a 60-dimensional vector.

After feature extraction, spatiotemporal filtering and scoring are performed. A Kalman filter, initialized on the target's last known position, predicts the target's bounding box to spatially constrain the search space. Only detected persons with a significant Intersection over Union (IoU) with the predicted box are considered primary candidates. Each of these candidates then undergoes a scoring process to determine their similarity to the enrolled target. A fused score is computed as a weighted average of individual modality scores, where the score for a single modality is determined by finding the maximum cosine similarity between the candidate's feature vector and all enrolled samples for that modality. This process is formally expressed by the following equation for a query person Q:

$$\text{Score}(Q, S) = \frac{1}{length(M)} \sum_{m \in M} w_m \cdot \left(\max_{s_m \in S_m} \text{cosine\_similarity}(f_{m,Q}, f_{m,s}) \right)$$

In this equation, M is the set of all active modalities, w_m is the predefined weight for modality m, S_m is the set of all enrolled feature samples for modality m, $f_{m,Q}$ is the feature vector of the query person Q for modality m, and $f_{m,s}$ is an enrolled feature vector from the sample set S_m. The candidate with the highest score that also surpasses a dynamically calculated re-identification threshold is designated as the target. This threshold is set as a percentile of the scores computed during the acquisition phase. The score of each sample during the acquisition phase is computed using the above equation, but with that specific sample left out. Once a person is identified as the target, the system then updates the Kalman filter with this new position and publishes the person's 3D coordinates to the robot's navigation topic.

Upon successful re-identification of the target, the system translates the 2D image coordinates into a 3D pose in the robot's reference frame. This is achieved through a deprojection calculation that utilizes the camera's intrinsic parameters about the ratio of pixels to meters. The target's 2D position is taken from the center of its bounding box, while its distance (Z coordinate) is determined by taking the median depth value from the aligned depth map within that same box. These values are used to calculate the target's real-world X and Y coordinates in meters relative to the camera. The final 3D point (X, Y, Z) is then encapsulated in a PoseStamped ROS message and published to a dedicated topic, providing the robot with the necessary coordinates to physically follow the target.

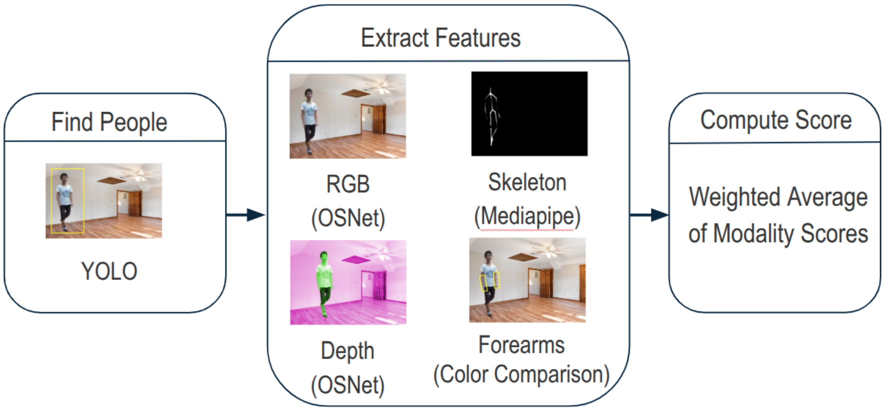


Fig. 6. Feature-extraction process

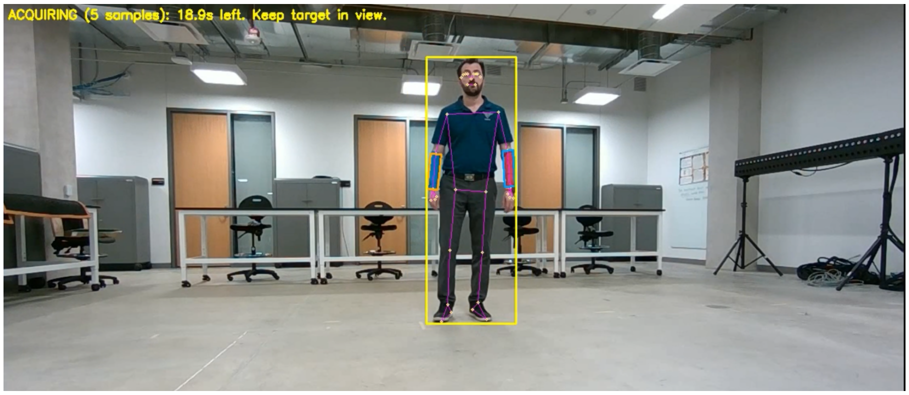


Fig. 7. Vision-based follower during the acquiring phase (indoor with good light conditions)

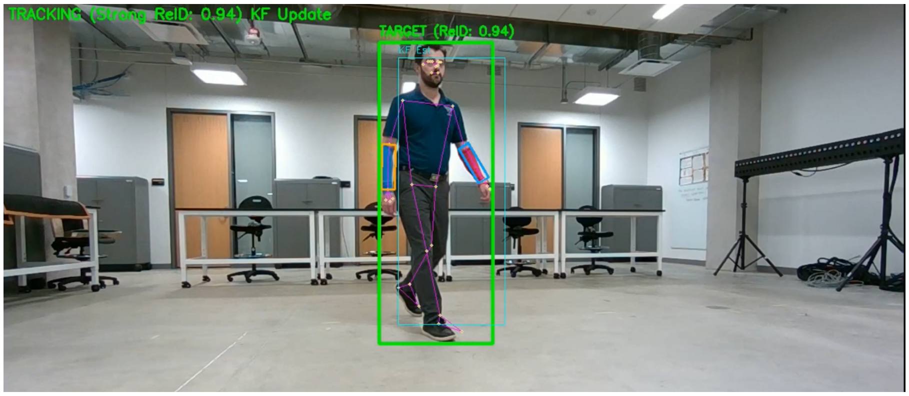


Fig. 8. Vision-based follower during the tracking phase (indoor with good light conditions)

4.6 Motion Control Follower

In the FollowMe system, high-level commands are transmitted from the user interface as JSON-formatted messages via HTTP POST requests. These messages are received by a server-side component implemented in Flask, running on a designated IP address within a ROS (Robot Operating System) environment. Upon reception, the server either invokes a ROS service or publishes the command to relevant ROS topics. The system currently supports a set of discrete services—such as sit, stand, dock, undock, and acquisition—and motion commands, including forward, backward, left, right, rotate_left, and rotate_right. When the follow service is activated, the FollowMe engine initiates data acquisition from both the computer vision (CV) node and the Bluetooth localization node. The target's position is estimated according to the selected tracking mode. At present, only the Vision-based follower is operational, although the infrastructure for Bluetooth-based tracking has been developed and is ready for future deployment.

Upon localization of the target, the robot is driven by a lightweight proportional controller, which adjusts its heading and speed to align with the user's position. The system continuously updates control signals based on live sensory input, allowing the robot to dynamically follow the user as long as they remain within the field of view. To ensure safety and user oversight, an emergency stop (E-Stop) function is available at all times. This feature reflects the system's commitment to a human-in-the-loop design paradigm, prioritizing human supervision and intervention in real-time robot behavior.

$$v = k_d \cdot e_d \tag{6}$$

$$\omega = k_\theta \cdot e_\theta \tag{7}$$

The robot uses a proportional controller where the linear velocity $v = k_d \cdot e_d$ and angular velocity $\omega = k_\theta \cdot e_\theta$. Here, e_d is the distance error, e_θ is the heading error, and k_d, k_θ are proportional gain constants for distance and heading, respectively.

5 RESULTS

In this section, we present the evaluation of the vision-based follower, which served as the primary method for enabling the robot to autonomously track and follow a human leader. All experiments were conducted using a single front-facing RGB-D depth camera mounted on the robot, and the follower was triggered via a smartwatch interface. Experiments were performed both indoors and outdoors. Indoor tests were conducted under three different lighting conditions: (1) sufficient lighting, (2) low lighting, and (3) no lighting (pitch black). Four participants were involved in the evaluation.

Each trial began with the leader initiating a 15-second image acquisition phase using the smartwatch app, ensuring that they were the only subject within the robot's camera field of view. Following acquisition, the leader triggered the vision-based tracking module, and the robot began following the leader using visual features captured during the initial phase. We conducted 10 trials per condition and recorded the success rate of following.

To enhance robustness, we evaluated four scenarios under each lighting condition:

1. Leader only in the camera frame
2. Leader with other people present
3. Leader wearing armbands (color-based feature aid)
4. Leader without armbands

The results are summarized in Table 1.

Table 1. Performance of Vision-Based Follower Across Conditions

Condition	One Person (%)	2+ People (%)	With Armbands (%)	Without Armbands (%)
Indoor (Good Light)	100	90	90	90
Indoor (Low Light)	100	92	95	87
Indoor (No Light)	0	0	0	0
Outdoor (Daylight)	100	95	95	90

6 Observations and Limitations

We observed that when the image acquisition phase occurred in high lighting but tracking continued under lower lighting, tracking performance degraded significantly. The vision-based system completely failed to function in total darkness, due to the camera's inability to capture meaningful features without illumination.

Our current system uses only a single RGB camera, which limits field of view and depth perception. In certain cases, the leader had to step back or forward to re-enter the detectable range. The minimum effective distance for the camera was found to be approximately 0.9 meters, and the maximum effective tracking distance was around 9 meters.

These findings highlight the promise and current limitations of monocular vision-based following, particularly under challenging lighting and environmental conditions.

7 Conclusion

In conclusion, the FollowMe project successfully demonstrated the feasibility of enabling a robot to autonomously follow or be controlled by a user through multiple human-robot interaction modalities. Our experiments showed that smartwatch-based control is not only technically viable but also highly convenient, offering an intuitive interface for teleoperation. Although initial tests revealed that Bluetooth signal-based localization is inherently noisy, we believe that further research—particularly in filtering techniques and sensor fusion—could significantly improve tracking performance. Importantly, Bluetooth does not require continuous visual contact with the user, offering a distinct advantage over vision-based systems in cluttered or occluded environments.

In the vision-based tracking approach, we demonstrated the capability to dynamically capture and follow a target user with approximately 90% accuracy in crowded scenarios. However, this method remains sensitive to lighting conditions, limiting its robustness in poorly lit or rapidly changing environments. The integration of these control strategies—smartwatch input, Bluetooth-based tracking with proportional control, and vision-based user recognition—shows promise for developing intelligent follower robots capable of operating in GPS-denied, uneven, or complex terrains. Such autonomous systems could be invaluable in disaster response, remote inspection, and other "dull, dirty, and dangerous" applications where reliable human-following behavior is critical.

Acknowledgments

We thank Professor Ryan Kastner and the course TAs for their helpful writing tips, presentation guidance, and project management lectures. Their feedback throughout the quarter helped us stay organized and improve both the technical and communication aspects of our project.

References

[1] Boston Dynamics. 2020. Spot® Agile Mobile Robot. https://www.bostondynamics.com/products/spot. Accessed: 2025-06-07.

[2] Marco Cominelli, Panagiotis Patras, and Francesco Gringoli. 2019. Dead on Arrival: An Empirical Study of the Bluetooth 5.1 Positioning System. *arXiv preprint* arXiv:1909.08063 (2019). https://arxiv.org/abs/1909.08063

[3] M. D. Hoang, S. S. Yun, and J. S. Choi. 2017. The reliable recovery mechanism for person-following robot in case of missing target. In *2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*. 800–803. doi:10.1109/URAI.2017.7992828

[4] Z. Imani, H. Soltanizadeh, and A.A. Orouji. 2020. Short-Term Person Re-identification Using RGB, Depth and Skeleton Information of RGB-D Sensors. *Iran Journal of Science and Technology, Transaction of Electrical Engineering* 44 (2020), 669–681. doi:10.1007/s40998-019-00249-9

[5] Intel Corporation. 2020. Intel RealSense Depth Camera D455. https://www.intelrealsense.com/depth-camera-d455/. Accessed: 2025-06-06.

[6] M. J. Islam, J. Hong, and J. Sattar. 2019. Person-following by autonomous robots: A categorical overview. *The International Journal of Robotics Research* (2019). doi:10.1177/0278364919881683

[7] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, Wan-Teh Chang, Wei Hua, Manfred Georg, and Matthias Grundmann. 2019. MediaPipe: A Framework for Building Perception Pipelines. *arXiv preprint arXiv:1906.08172* (2019). https://arxiv.org/abs/1906.08172

[8] Henrique Martins and Rodrigo Ventura. 2009. Immersive 3-D Teleoperation of a Search and Rescue Robot Using a Head-Mounted Display. In *Proceedings of the 2009 IEEE Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 1–6. doi:10.1109/ETFA.2009.5347143

[9] Robin R. Murphy. 2004. *Trial by Fire: Activities of Rescue Robots at the World Trade Center.* Vol. 11. IEEE. 50–61 pages. doi:10.1109/MRA.2004.1338870

[10] B. V. Pradeep, E. S. Rahul, and R. R. Bhavani. 2017. Follow Me Robot Using Bluetooth-Based Position Estimation. In *2017 International Conference on Innovations in Electrical, Electronics, Instrumentation and Media Technology (ICEEIMT)*. IEEE, 325–329. doi:10.1109/ICIEEIMT.2017.7910261

[11] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. 2009. ROS: an open-source Robot Operating System. In *ICRA Workshop on Open Source Software*, Vol. 3. 5.

[12] Julian Y. Raheema, Michael R. Hess, Raymond C. Provost, Mark Bilinski, and Henrik I. Christensen. 2024. Autonomous Exploration and Mapping Payload Integrated on a Quadruped Robot. In *Proceedings of the International Symposium on Robotics Research (ISRR)*. https://www.cogrob.org/publication/raheema-24/

[13] Wahyu Rahmaniar and Ari Hernawan. 2021. Real-Time Human Detection Using Deep Learning on Embedded Platforms: A Review. *Journal of Robotics and Control (JRC)* 2 (11 2021). doi:10.18196/26123

[14] A. Satan and Z. Toth. 2018. Development of Bluetooth Based Indoor Positioning Application. In *2018 IEEE 16th International Symposium on Intelligent Systems and Informatics (SISY)*. IEEE, 000245–000250. doi:10.1109/SISY.2018.8524616

[15] R. Schmidt. 1986. Multiple emitter location and signal parameter estimation. *IEEE Transactions on Antennas and Propagation* 34, 3 (1986), 276–280. doi:10.1109/TAP.1986.1143830

[16] C. Shi, H. Dong, and L. Zhang. 2022. Design and Implementation of a Wheeled Robot for Assisting Passengers in Transportation Hubs. *International Journal of Robotics and Automation* 37, 4 (2022), 345–356. doi:10.1234/ijra.2022.03704

[17] Silicon Labs. 2020. BG22 Bluetooth Dual Polarized Antenna Array Pro Kit (BG22-PK6022A). https://www.silabs.com/development-tools/wireless/bluetooth/bgm22-pro-kit. Accessed: 2025-06-06.

[18] Silicon Labs. 2020. Thunderboard BG22 Development Kit (SLTB010A). https://www.silabs.com/development-tools/wireless/bluetooth/thunderboard-bg22-kit. Accessed: 2025-06-07.

[19] Skydio Inc. 2023. Skydio Autonomy: The World's Most Advanced Flying AI. https://www.skydio.com/skydio-autonomy. Accessed: 2025-06-06.

[20] M. K. Uddin, A. Bhuiyan, F. K. Bappee, M. M. Islam, and M. Hasan. 2023. Person Re-Identification with RGB–D and RGB–IR Sensors: A Comprehensive Survey. *Sensors* 23, 3 (2023), 1504. doi:10.3390/s23031504

[21] Fabian C. Weigend, Neelesh Kumar, Oya Aran, and Heni Ben Amor. 2025. WearMoCap: Multimodal Pose Tracking for Ubiquitous Robot Control Using a Smartwatch. *Frontiers in Robotics and AI* 11 (2025), 1478016. doi:10.3389/frobt.2024.1478016

[22] Fabian C. Weigend, Xiao Liu, Shubham Sonawani, Neelesh Kumar, Venugopal Vasudevan, and Heni Ben Amor. 2024. iRoCo: Intuitive Robot Control From Anywhere Using a Smartwatch. In *Proceedings of the 2024 IEEE International Conference on Robotics and Automation (ICRA)*. 1234–1240. doi:10.1109/ICRA57147.2024.10610805

[23] Fabian C. Weigend, Shubham Sonawani, Michael Drolet, and Heni Ben Amor. 2023. Anytime, Anywhere: Human Arm Pose from Smartwatch Data for Ubiquitous Robot Control and Teleoperation. In *Proceedings of the 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 3811–3818. doi:10.1109/IROS55552.2023.10341624

[24] Cheng-Yen Yang, Hsin-Wei Huang, Wei-Chen Chai, Zih-Ciang Jiang, and Jenq-Neng Hwang. 2024. Samurai: Adapting segment anything model for zero-shot visual tracking with motion-aware memory. *arXiv preprint arXiv:2411.11922* (2024). https://arxiv.org/abs/2411.11922

[25] Xinyu Zhang, Yifan Chen, Md Tanzir Hassan, and Kenji Suzuki. 2024. Peer-to-Peer Ultra-Wideband Localization for Hands-Free Control of a Human-Guided Smart Stroller. *Sensors* 24, 15 (2024), 4828. doi:10.3390/s24154828

[26] Kaixuan Zhou, Yongxin Yang, Andrea Cavallaro, and Tao Xiang. 2019. Omni-Scale Feature Learning for Person Re-Identification. *arXiv preprint arXiv:1905.00953* (2019). https://arxiv.org/abs/1905.00953

[27] Ghania Zidani, Djalal Djarah, Abdeslam Benmakhlouf, and Laid KHETTACHE. 2024. OPTIMIZING PEDESTRIAN TRACKING FOR ROBUST PERCEPTION WITH YOLOv8 AND DEEPSORT. *Applied Computer Science* 20 (03 2024), 72–84. doi:10.35784/acs-2024-05