ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

CVL Computer Vision Lab

Appearance-Based 3D Tracking and Navigation for Robotic Dogs

Designing a Person-Following Robot

Robot Learning Project

Dario Dittli[*] Pablo Soler Garcia[*] Sam Sulaimanov[*,†]

Department of Information Technology and Electrical Engineering

[*]ETH Zurich
[†]Octanis Instruments GmbH

Advisors: Yutong Hu
Supervisor: Prof. Dr. Fisher Yu

July 04, 2024

Chapter 1

Introduction

This project presents the development of an autonomous person-following system for a quadruped robot, specifically the Unitree Go1. The project addresses the growing need for advanced robotic assistance in various fields, including guide dogs for the visually impaired, security applications, and enhanced human-robot interaction. Our approach integrates cutting-edge object detection, person re-identification, and navigation technologies to create a robust tracking and navigation system. We equipped the Unitree Go1 with a custom-designed backpack containing a Jetson Orin Nano computer, ZED2i stereo camera, and LiDAR sensor to enable accurate perception and decision-making.

Quadruped robots have emerged as a promising platform for a wide range of applications due to their versatility and ability to navigate complex terrains. Recent advancements in robotics, computer vision, and artificial intelligence have significantly expanded the potential of these machines. Our project focuses on enhancing the capabilities of quadruped robots, specifically in the domain of autonomous person-following. This functionality is crucial for various real-world applications, including assistive technologies for the visually impaired, advanced security systems, and more intuitive human-robot interaction in both domestic and industrial settings. By developing a robust person-following system, we aim to bridge the gap between current robotic capabilities and the growing demand for more sophisticated, autonomous robotic assistants that can seamlessly integrate into human environments.

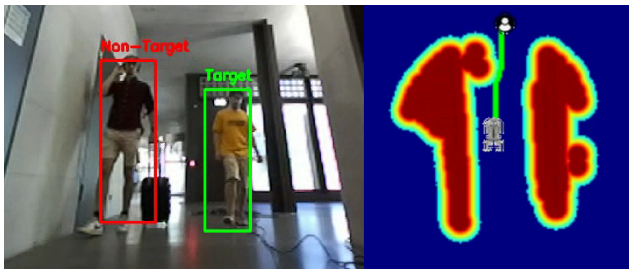

Figure 1.1: **Person Tracking Algorithm**
Right: Vision algorithm marking target and pedestrians. Left: 2D LiDAR map with path to target.

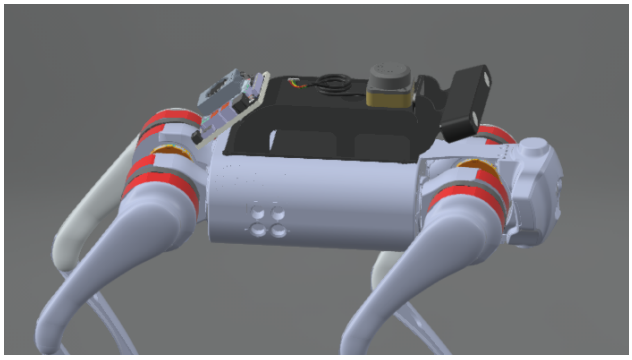

Figure 1.2: **Custom-Designed Tracking Module**
CAD model of Unitree GO1 with our custom backpack for person tracking.

CHAPTER 1. INTRODUCTION

The main goal of this work is to build a system that allows a robotic dog to follow a specific person on its own. This involves creating a reliable person tracking and navigation system that works well in different and changing environments.

We used the Unitree Go1 quadruped robot for this project. We equipped it with a special tracking module that includes a computer, stereo cameras, and a LiDAR sensor. We combined advanced object detection and person re-identification technologies to help the robot accurately follow a person.

This thesis covers the design, building, and testing of the tracking and navigation system for the robotic dog. The chapters are organized as follows: Chapter 2 reviews related research in robotics, navigation, and vision. Chapter 3 explains the methods and technologies we used, including hardware and software details. Chapter 4 describes the experiments we did and the results we found. Finally, Chapter 5 discusses what our findings mean and possible future work.

Our main contributions in this project are:

- Development of a custom, integrated hardware solution for quadruped robots, combining stereo vision, LiDAR, and edge computing capabilities.
- Implementation of an advanced person tracking algorithm that fuses object detection and re-identification techniques for robust target following.
- Design of a navigation system that enables autonomous path planning and obstacle avoidance in dynamic environments.
- Demonstration of the feasibility and effectiveness of autonomous person-following capabilities in quadruped robots.

Chapter 2

Related Work

This chapter reviews key research in robotics, navigation, and vision, providing context for our own work and highlighting the contributions of others in these fields.

2.1 Robotics

Recent advancements in robotics have focused significantly on enhancing autonomous behavior and interaction. For instance, Zhang et al. developed a system that uses RGB-D sensors to enable robots to follow people effectively [19]. This approach has practical applications in making robots more interactive and responsive. Extending this concept to quadruped robots, Yu et al. utilized camera systems to achieve reliable person-following capabilities [18]. Furthermore, the Unitree navigation framework by Moravec offers an open-source solution that supports complex navigation tasks for quadruped robots [12]. Additionally, Hughes' work with ROS 2 integration for Unitree robots facilitates their use in more diverse robotic applications [6].

2.2 Navigation

Effective navigation is crucial for autonomous robots to operate safely and efficiently. Macenski et al. introduced the Marathon 2 navigation system, which integrates various technologies to ensure reliable long-term autonomous navigation [10]. Their work is notable for its comprehensive approach to overcoming the challenges of sustained operation. In a different direction, Chraibi and Wang explored the use of reinforcement learning for crowd navigation, showing how adaptive algorithms can significantly improve a robot's ability to maneuver through dynamic and crowded environments [3].

2.3 Vision

Vision technology plays a critical role in enabling robots to perceive and interact with their surroundings. Mur-Artal and Tardós' ORB-SLAM2 is a notable system that supports monocular, stereo, and RGB-D cameras, forming a foundation for effective visual SLAM [13]. Liao et al. built on this by developing Crowd SLAM, which addresses the unique challenges of navigating crowded spaces [8]. In terms of multi-object tracking, the QD-Track algorithm [14] represents a significant advancement, enhancing detection and re-identification capabilities across various domains by leveraging quasi-dense similarity learning. QDTrack densely samples numerous object regions across image pairs for contrastive learning. This approach enhances the feature space, enabling effective object association through a simple nearest neighbor search during inference. The SOLIDER project [2] also contributes to person re-identification with a training method that uses human prior knowledge to generate pseudo-semantic labels, enhancing the semantic richness of representations tailored for various downstream human-centric visual tasks such as re-identification, attribute recognition, and human parsing. The algorithm demonstrated superior performance on six human-centric visual tasks, setting new benchmarks.

Chapter 3

Methods

To create our person-following robot, we utilized the Unitree Go1, which we equipped with a self designed tracking pack containing a PC, stereo cameras, and a LiDAR sensor. Our system integrated an object detection framework paired with a person re-identification model, enabling the robot to track and follow a specific individual. For navigation, we computed a 2D costmap generated by the LiDAR using the Nav2 framework and used a path planning tool to effectively navigate to the tracked person. The general pipeline can be seen in fig. 3.1

The following sections explain in more detail the hardware setup, the vision algorithm and the navigation tools.

3.1 Hardware Setup[1]

To leverage the state-of-the-art navigation algorithms in the ROS 2 ecosystem, we selected hardware commonly utilized in this field.

Initially, we experimented with the Intel Realsense camera but ultimately chose the ZED 2i camera due to its superior performance, wider field of view, built-in IMU, and robust SDK. The camera was mounted on a 3D-printed angled bracket, optimised for full view of people in front of the robot.

For obstacle detection, we opted for the ”RPLIDAR C1” 2D LiDAR, which operates at 10Hz and has a ranging distance of up to 12m. It includes a ROS node that seamlessly integrates with the Nav2 framework for occupancy map generation.

Given the high data throughput from the high-resolution camera, LiDAR, and multiple data streams in ROS, we selected the high-performance single-board computer ”NVIDIA Jetson Orin Nano.” The SDK for our camera is optimized for this board, which also offers sufficient performance capacity to run our algorithms. This selection was critical for our project, as initial trials with a Raspberry Pi 5 demonstrated the need for a more robust computing solu-

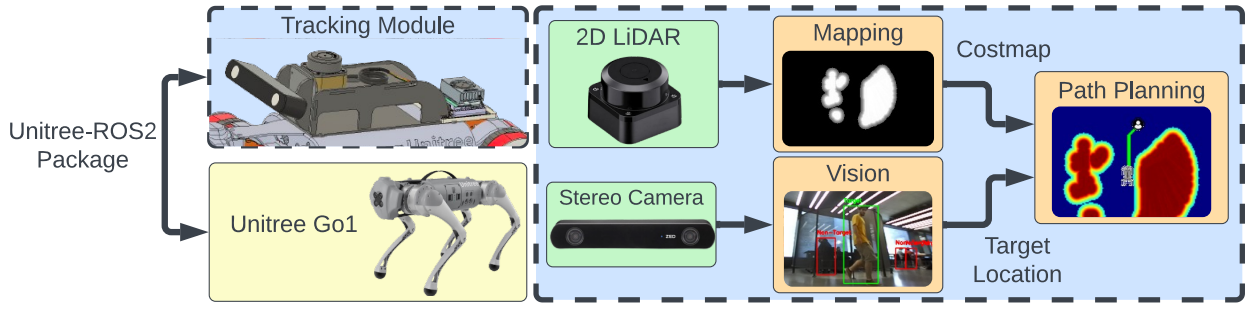

Figure 3.1: **Pipeline Overview:** The mounted tracking module consists of a 2D LiDAR sensor, a stereo camera and a Jetson Orin Nano. The target is detected by the vision pipeline and mapped into the 2D plane using the depth values of the stereo camera. Using the costmap of the surroundings a path is traced to the target.

tion.

The single-board computer was equipped with a 1TB NVMe SSD, which provided fast data recording capabilities, superior to traditional SD card recording methods.

The camera, LIDAR and single board computer were attached onto a custom made laser cut housing that was screwed onto the Unitree Go1. The housing was designed in Autodesk Fusion 360 and all components were checked for good field of view and interferences prior to manufacturing.

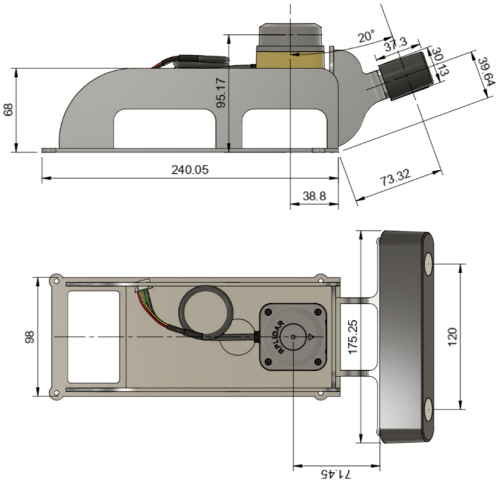

Figure 3.2: **Tracking Module Design:** Model designs of the tracking mount with stereo camera, LiDAR and space for the onboard PC.

Finally, the single board computer was connected via Ethernet cable to the Unitree Go1 which is how high level commands were sent to it.

3.2 Vision

The vision algorithm consists of two main blocks: The person detection and the person re-identification.

3.2.1 Person Detection

The person detection pipeline consisted of a YOLO-v5 [7] with a Simple Online and Realtime Tracking (SORT) algorithm [1]. YOLO-v5 is a convolutional neural network based model that has been very successful in object detection tasks and also has been highly optimized to run on mobile devices, which is perfect for our task. SORT is a multiobject tracking algorithm that uses a Kalman filter coupled with Hungarian matching to assign new detections to tracked objects. We then discovered that Stereolabs already implemented the Yolo + SORT vision model in their SDK [15] and optimized it to run inference in the Jetson. This SDK already uses the depth values of the camera to output a 3D-position of the object in addition to the image bounding boxes and the object ID. However, this ID is not consistent enough for following a person over a long period of time.

3.2.2 Person Re-Identification

Person Re-identification is applied to new objects to decide if the person is the target. For this module we opted for a vision transformer trained for feature matching [16]. The SOLIDER vision transformer currently marks third in the person re-identification benchmark Market-1501 [20] and first in second in MSMT17 [17], another pedestrian re-identification dataset. We use the pre-trained weights of the very successful base-SWIN model architecture [9] trained on Market-1501 using SOLIDER. We prepare a dataset of 5 images of the target, this can happen either at the beginning of the tracking or be done beforehand. The features are then computed and stored. When an untracked detection appears, the cropped image of the person resized to 384 by 128 pixels is given to the transformer and the features are compared using L2 distance to all the 5 reference images. If an image is below a distance threshold to any of the 5 images, the detected object is marked as the target.

To reduce the number of inferences of the re-identification model which slowed down the robot, we came up with a set of tricks for processing new detections. The graph in figure 3.3 shows the complete algorithm running on new detections. If the system already is following the target, all new detected objects are ignored. If there is no target, new

[1] All hardware except for the Unitree Go1 was sponsored by Octanis Instruments GmbH, `https://www.octanis.ch`.

detections will first be put in a tempting container and ran through the person re-identification network. If the transformer matches the person with the target reference images 10 times, the detected person will become the target. After 10 mismatches, the detection will be ignored. This method reduces the computational load while maintaining a robust tracking performance.

We were planning on using QD-Track [14], but we were not able to find the weights for our specific task. We also looked at QD-3D Object Tracking [5] which also seemed suitable for our task, but needed also the camera intrinsics and extrinsics and due to the lack of documentation on how to do inference with the models and our concern that we needed a more lightweight model to run on our Jetson Orin Nano, we opted for a two step vision pipeline.

3.3 Navigation

The following navigation section proposes two methods for controlling the robot to follow the target. First, we implemented a simple "go forward until you reach the goal" method and a more sophisticated path planning algorithm using the LiDAR's data.

3.3.1 "Go straight"

We noticed that if the robot is closely following the target navigation can be simplified by just following the target and letting him navigate. This saves a lot of computational resources and is very successful. Our naive approach simply will go straight if the person is in the central fifth of the image and left and right respectively if the person appears on the left side or the right side of the image. To make it smoother we also added a linear increase in rotation velocity the more to the sides of the image the bounding box of the person is. The robot will start walking forward whenever the person is further than 1.5 meters away and its speed will also linearly increase until a maximum velocity is reached when the person is more than 4 meters away. The decision boundaries of this method can be seen in fig. 3.4.

3.3.2 Path Planning

The main issue with the simple navigation algorithm is that it lacks obstacle avoidance. If between the robot and the target there is a trash can, a bench or other people, it will not be able to avoid colliding with the objects. Furthermore, it cannot be extended to also map the environment along the way or search for the target in its last seen position. For these reasons we made use of the LiDAR to have better perception of the environment. We load the output of the 2D LiDAR with ROS to Nav2 [11]. The Nav2 package is a comprehensive navigation framework for mobile robots that provides planning, control, localization, and behavior management capabilities for autonomous navigation in complex environments. From this package we are only able to use the costmap generation, because the navigation part is to resource intensive to run in parallel to our vision pipeline. The costmap generated is a 2d map of 12m by 12m in size, with a pixel resolution of 5cm. It is oriented with respect to the robot (not to odometry's static orientation) to align the coordinate frame with the coordinates of the camera detections. In figure 3.5, the LiDAR measurements and computed costmaps are shown. Each sensor measurement creates an occupancy circle with a radius of 50cm in which its cost decays with the distance to the center. We are then able to project the coordinates of the target into the 2D plane of the costmap. This can be visualized using foxglove as shown in fig. 3.6.

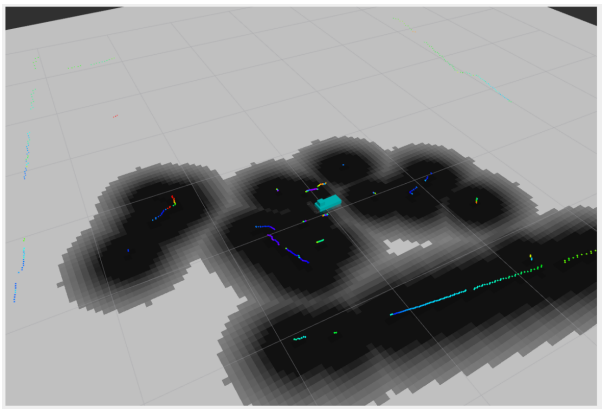

Figure 3.5: **LiDAR and Costmap:** RViz2 visualization of the LiDAR sensor measurements (colored) and the generated Nav2 costmap (black and white).

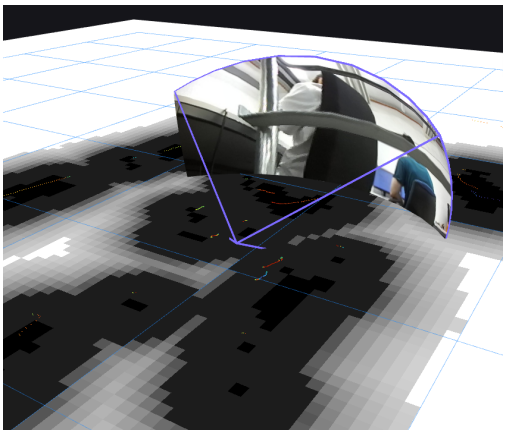

Figure 3.6: **Costmap and Camera:** Foxglove visualization of the generated Nav2 costmap (black and white) and the camera image projection.

Once the costmap is computed, we tested different shortest path algorithms and ended up implementing the Dijkstra algorithm [4]. Compared to other algorithms such as A-star, the Dijkstra algorithm is more compute intensive, but does not create artifacts and that is the reason we chose it over others. With A-star, for example, we noticed it tends to go back sometimes while exploring diagonal paths which made the robot first go backwards. The resulting path planned by our Dijkstra path planning implementation can be seen in figure 3.7.

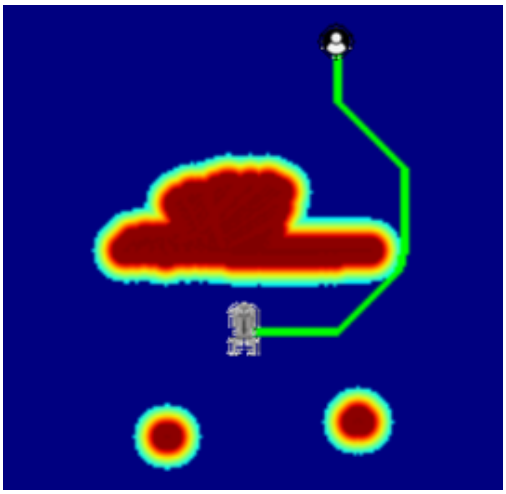

Figure 3.7: **Dijkstra Path Planning:** Path computed by the Dijkstra algorithm from the robot to its target.

We also preprocess the map to reduce the time it takes for path planning by halving the resolution to 10cm per pixel, cropping it to only use the forward half of the map and making the costs binary (either occupied or free). Furthermore, the target point is usually occupied by the person so we set circle with the radius of 0.5m around the target to 0 in the costmap so the algorithm is able to converge to a path. Figure 3.8 shows the comparison between the original costmap and the processed obstacle map showing these changes. Sometimes it is still not able to find a path so we set a timeout of 0.5 seconds to compute the path. Once we have a path we keep refreshing the vision and navigation commands for the next checkpoint in the path with 30Hz but the path is only recalculated once a second. In case that after one second we are not seeing the target we keep following the old path and will recalculate the path when the target reappears. This can be seen in the navigation graph in figure 3.9

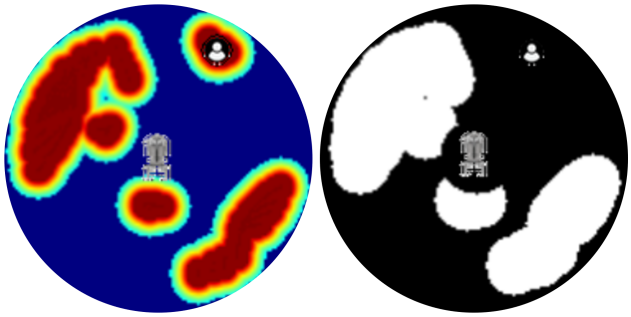

Figure 3.8: **Costmap Processing:** Comparison of the originally computed costmap (left) and the binary, lower resolution map (right) computed for faster path planning. Costs around robot and goal are set to zero.

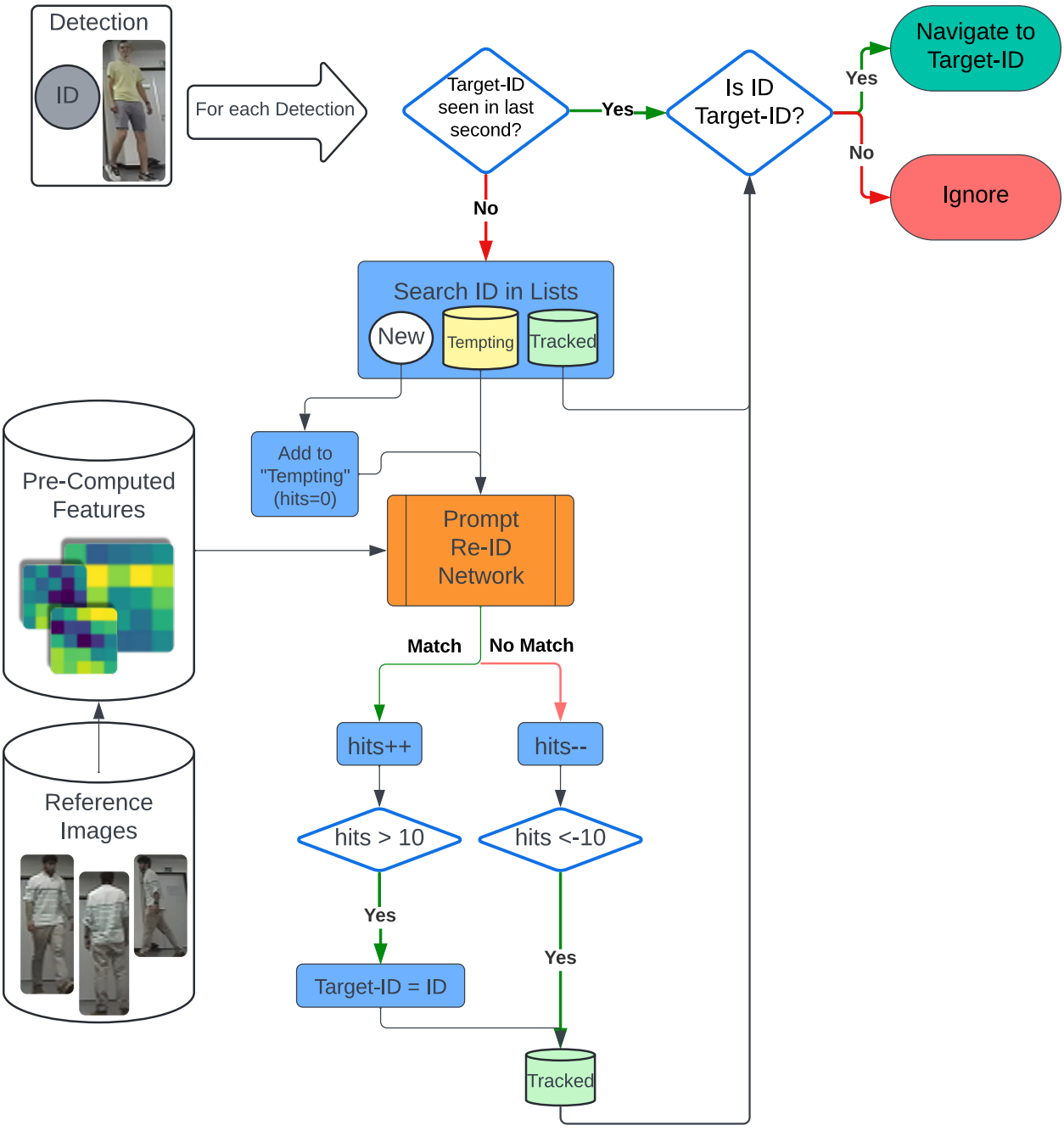

Figure 3.3: **Target Tracking** The graph shows the decisions tree for when a person is detected. In case the target is already being tracked, every other person is ignored. When no target exist, new objects are classified as target or non-target after 10 consecutive classifications by the person re-identification model.

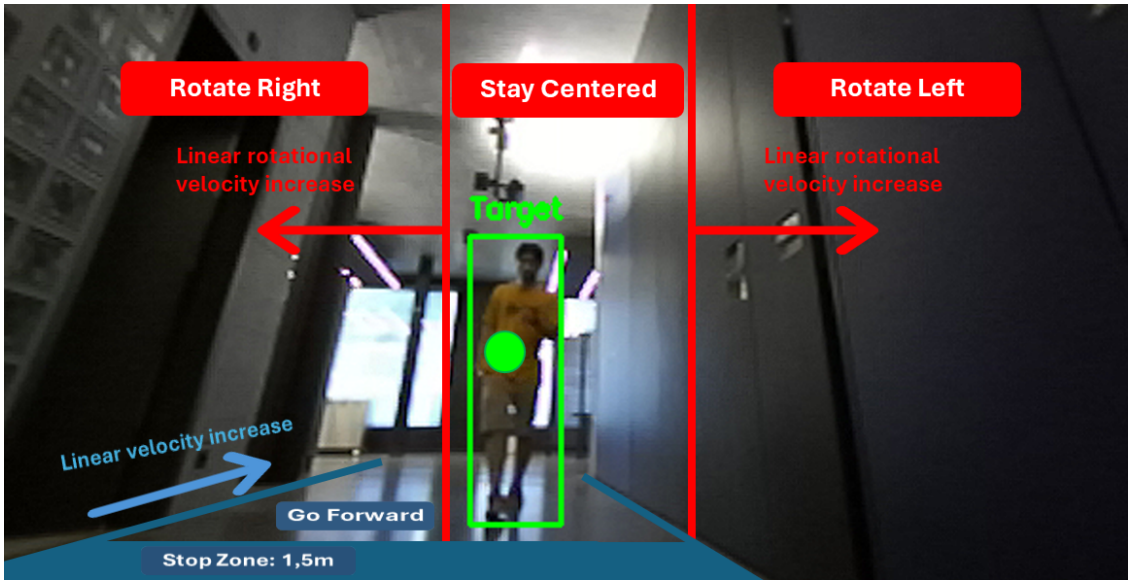

Figure 3.4: **Simple Navigation Algorithm:** In our first, simple navigation approach, the robot will maintain the person centered by rotating left or right depending on the position of the bounding box in the image. Its speed is determined by how far away the person is, stopping when the person is less than 1.5 meters away.

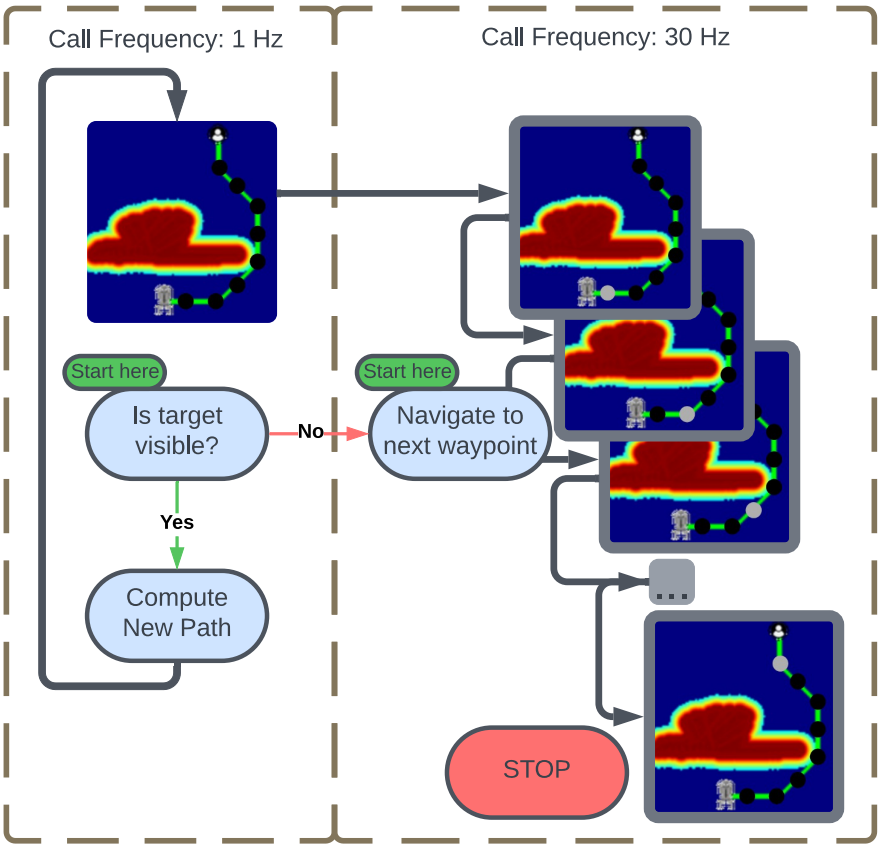

Figure 3.9: **Path Planning based Navigation:** Our proposed method walks the computed path by following the waypoints of the path. Every second if the target is visible, the robot will recalculate the path using the updated LiDAR costmap.

Chapter 4

Experiments and Results

We conducted two key experiments to evaluate the capabilities of our system. The first experiment aimed to test the overall performance of the system in practical scenarios. We focused on functions, including the robot's ability to follow moving targets, track objects accurately, avoid obstacles, and plan efficient paths. This broad assessment was designed to understand how well the system performs in real-world conditions.

In the second experiment, our focus narrowed to the tracking component, a central aspect of the project's "robot learning" objectives. By using a controlled environment, we benchmarked different tracking algorithms to evaluate their effectiveness. This experiment aimed to identify which tracking method offered the best accuracy and reliability under various conditions, comparing established techniques with the advanced algorithms developed as part of our project.

4.1 Live Experiment

We conducted a series of live experiments to evaluate the various capabilities of our implementation. The primary focus was on the following aspects:

- **Dynamics of Following:** We assessed the system's ability to maintain a consistent and appropriate distance while following a target. This involved testing the responsiveness of the robot to changes in the target's speed and direction, ensuring smooth and stable following behavior.
- **Tracking of Target:** This capability was tested to ensure accurate identification and consistent tracking of the correct target without misidentification. We evaluated the system's robustness in different university locations and using different targets.
- **Obstacle Avoidance:** We tested the system's ability to detect and navigate around obstacles effectively. In the experiments stationary and moving objects (like people passing by) challenged the robot's real-time decision-making and path adjustment capabilities.
- **Path Planning:** The efficiency and accuracy of the system's path planning algorithms were evaluated. This involved testing the robot's ability to plan paths to follow the target while avoiding obstacles and efficiently reaching the destination.

4.1.1 Results

The robot demonstrated a high level of responsiveness to the dynamic movements of the target. It swiftly adjusted its orientation whenever the target altered its direction. Thanks to the wide field of view of the stereo camera, the robot was able to maintain the target in the image frame. In the attached videos, one can see even when the target is close and makes an abrupt change of direction, the robot is able to follow.

Regarding the tracking algorithm, it performed well under most conditions. However, two notable flaws were observed. Firstly, there were instances of misidentification, where the robot incorrectly identified and followed a person other than the target. The robot experienced around 1 misidentification every time we performed a test walk which lasted 15 minutes. This issue may stem from the size of

the target dataset; either an insufficient or excessive number of images in the feature dataset could lead to incorrect identification. Secondly, the SORT algorithm occasionally confused the target with another person when they moved in close proximity, causing the robot to follow the wrong individual without even noticing there was a change in the target. This is displayed in the sequence of images of fig. 4.3

Due to time constraints, we were unable to integrate obstacle avoidance and path planning into a single operational code. Path planning when deployed online causes too much delay on the robot navigation to smoothly follow the target. Nonetheless, we did, while following the target with the "Go straight" navigation, compute and save the paths and visualized them in the video attached to this report. A snapshot of this video is visible in figure 4.1. We highly encourage the reader to watch the 3 attached videos, as they are a good overview of the capabilities of our method.

Since obstacle avoidance and route planning are interdependent, we evaluated them together. By designating a virtual target point on the cost map generated by the LiDAR or simulated, we were able to test our path planning navigation pipeline. The main issue we discovered is, that when the map is too crowded it often occurs that the Dijkstra algorithm fails to plan a path. This happens for example when walking through a narrow corridor where other pedestrians are walking by. It is also visible in fig. 4.2 that the LiDAR sporadically senses an obstacle where there is none which results in a suboptimal path.

4.2 Vision Experiment

The vision experiment aimed to benchmark the tracking algorithms by using a recorded ROS bag where the robot was manually controlled by an operator to follow a predefined person wearing an ArUco tag. This setup allowed us to create a controlled environment to compare different tracking methods.

The experiment compared the performance of an ArUco tag detector with a custom tracker that utilizes object detection from the ZED2i camera along with a re-identification pipeline. The goal was to evaluate which method provides more accurate and reliable tracking.

4.2.1 Results

The table below presents key metrics for both the ArUco tag detector and the custom tracking algorithm. These results provide insight into the comparative effectiveness of each method.

The results indicate the effectiveness of each method in maintaining accurate tracking of the target. The ArUco tag detector serves as a reliable baseline, while the custom algorithm showcases advanced capabilities through the integration of object detection and re-identification techniques, potentially offering superior performance in dynamic and complex environments.

Metric	**ArUco Tag Detector**	**Custom Algorithm**
Processing Time per Frame (ms)	10.5	75.3
False Positive Rate (%)	3.6	10.5
False Negative Rate (%)	5.1	4.4
Detection Accuracy (%)	25.3	99.8

Table 4.1: Comparison of various performance metrics between the ArUco tag detector and the custom algorithm. The custom algorithm shows significantly better detection accuracy, though it is slower compared to ArUco.

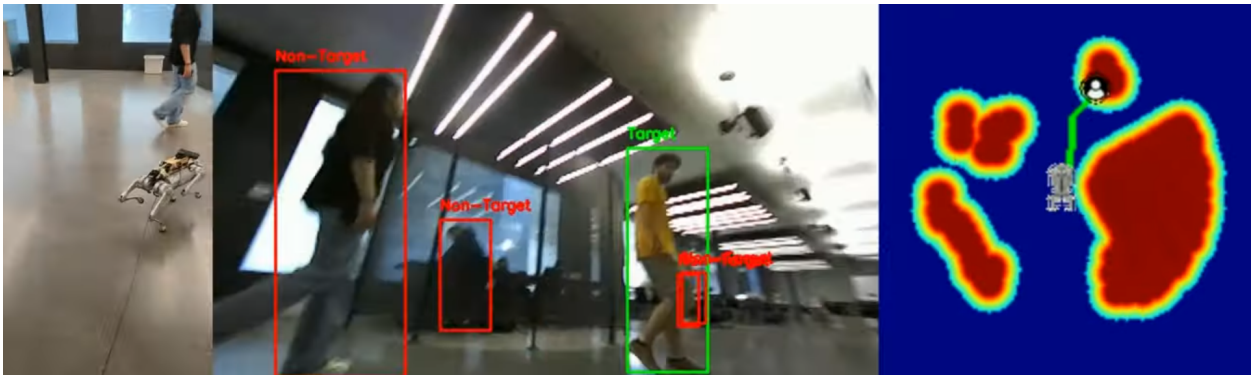

Figure 4.1: **Live Navigation Experiment:** Snapshot of the attached video of a live experiment including target detection and path planning.

Figure 4.2: **Live Navigation Experiment:** Snapshot of the attached video where the LiDAR due to noise creates aberrations in the costmap which distort the path to target.

Figure 4.3: **Tracking Failure:** This sequence of images shows the rare event in which the target ID (tracked by SORT) gets switched between two tracklets causing the robot to follow someone who is not the target. The left image shows the correct target with ID 74. In the second image, a pedestrian walks by very close, resulting in very similar bounding boxes. In the right image the target has been assigned the wrong tracklet and the robot is off on an unexpected adventure with this student that was passing by.

Chapter 5

Discussion

Our implementation of a person-following robot using the Unitree Go1 platform has yielded surprisingly good results, demonstrating the potential of integrating advanced vision algorithms with robotic navigation systems. While the nature of this project was more focused on implementation rather than pure research, the outcomes provide valuable insights into the practical challenges and opportunities in developing autonomous following systems. It's important to note that due to the implementation-oriented nature of this work, we had problems quantifying our results as precisely as one might in a more controlled research environment. Despite this limitation, we try to show the qualitative performance of the system in live experiments in videos and snapshots.

Our navigation experiments revealed an interesting dichotomy between simple following and path planning approaches. The "go straight" method, despite its simplicity, proved remarkably effective for close-range following. This suggests that in many practical scenarios, complex path planning might not always be necessary. However, the integration of LiDAR-based costmap generation and Dijkstra path planning demonstrated the potential for more sophisticated navigation in complex environments. With additional computational resources, such as a dedicated server for processing or a more powerful PC, real-time path planning could be achieved alongside the vision pipeline.

The re-identification component emerged as the main bottleneck in our vision pipeline. While it significantly improved the system's ability to maintain focus on the correct target, it wasn't as good as we want it to be for robust tracking. This, however, is only an issue if the target frequently disappears behind corners or other people.

From a hardware perspective, our setup could benefit from some upgrades. The need to power the PC and sensors forced us to have a long cable to connect to the power outlet. This could have been avoided by incorporating a battery, which would have freed the robot of any cables. Additionally, we found that the Jetson, should ideally be housed within the backpack to create a more compact design, instead of protruding from the back. In figure 5.1 you can see the final look of the robot.

In conclusion, our person-following robot implementation demonstrates the feasibility of creating a responsive and adaptable system using off-the-shelf components and custom algorithms. While there is room for optimization, particularly in the areas of re-identification, efficiency and power management, the overall performance of the system exceeded our initial expectations.

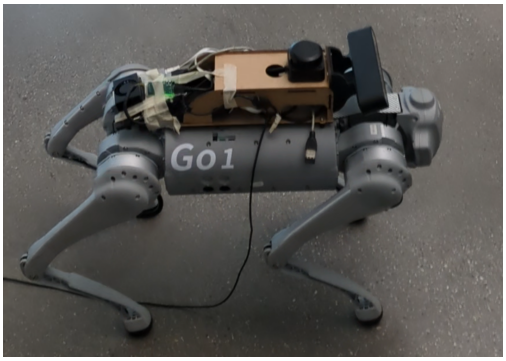

Figure 5.1: **Robot Final Look**

Chapter 6

Conclusion

This project has successfully developed and implemented an advanced person-following system for the Unitree Go1 robotic platform, demonstrating the potential for autonomous robotic assistants in various applications such as personal assistance, security, and human-robot interaction. We equipped the quadruped robot with a computer, stereo cameras and a LiDAR sensor, and combined advanced object detection and person re-identification technologies to create a reliable tracking and navigation system.

Our experiments show that the legged robot could accurately follow a person across various university settings. The system was able to track the person, avoid obstacles, and plan efficient paths. We also found some areas where the system could be improved, such as reducing misidentifications and improving the handling of crowded environments.

Overall, our work contributes to the field of robotics and autonomous navigation by providing a practical solution for person-following robots.

The next design iterations could benefit from a more compact, self-contained design with onboard power solutions. Additionally, future work could focus on further improving the system's accuracy and robustness, as well as exploring additional applications for this technology. Some of the venues to explore include:

- Autonomous map exploration to locate the target person.
- Dynamic obstacle avoidance by predicting the obstacle's movement.
- Interaction capabilities, integrating voice recognition or sign detection to allow for commands and a better human-robot interface.

This project demonstrates the feasibility of creating a reliable person-following robot that can reliably follow a specific target, paving the way for more advanced and helpful robotic assistants in the future.

Bibliography

[1] Alex Bewley, ZongYuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. *CoRR*, abs/1602.00763, 2016.

[2] Weihua Chen, Xianzhe Xu, Jian Jia, Hao luo, Yaohua Wang, Fan Wang, Rong Jin, and Xiuyu Sun. Beyond appearance: a semantic controllable self-supervised learning framework for human-centric visual tasks, 2023.

[3] F. Chraibi and Z. Wang. Crowd navigation with reinforcement learning. *MDPI*, 2023.

[4] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[5] Hou-Ning Hu, Yung-Hsu Yang, Tobias Fischer, Trevor Darrell, Fisher Yu, and Min Sun. Monocular quasi-dense 3d object tracking. *CoRR*, abs/2103.07351, 2021.

[6] Katie Hughes. Unitree ros2, 2023.

[7] Glenn Jocher. Yolov5 by ultralytics, 2020.

[8] X. Liao, H. He, and B. Zhang. Crowd slam: A new paradigm for simultaneous localization and mapping in crowds. *SpringerLink*, 2021.

[9] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[10] S. Macenski, F. Martín, R. White, and J. Clavero. The marathon 2: A navigation system. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[11] Steve Macenski, Francisco Martín, Ruffin White, and Jonatan Ginés Clavero. The marathon 2: A navigation system. *CoRR*, abs/2003.00368, 2020.

[12] Nathan Moravec. Unitree navigation, 2023.

[13] R. Mur-Artal and J. D. Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *arXiv*, 2017.

[14] B. Pang, Y. Qiu, Y. Li, H. Chen, C. Li, C. Lin, and X. Sun. Qd-track: Towards higher quality and diversity in multi-object tracking, 2021. arXiv preprint arXiv:2111.12038.

[15] Stereolabs. Stereolab's vision sdk.

[16] TinyVision. Solider-reid: A person re-identification system, 2023.

[17] Longhui Wei, Shiliang Zhang, Wen Gao, and Qi Tian. Person transfer GAN to bridge domain gap for person re-identification. *CoRR*, abs/1711.08565, 2017.

[18] Y. Yu, H. Wu, and J. Yi. A person-following with camera for quadruped robots. *ScienceDirect*, 2022.

[19] Z. Zhang, X. Zhang, and C. Peng. Person following robot using rgbd. *IEEE Xplore*, 2022.

[20] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.