

Politecnico di Torino

Politecnico di Torino

Master's Degree in Mechatronic Engineering

Multi-camera Perception and Control for Human-aware Robot Navigation

Supervisors:

Marcello Chiaberge
Mauro Martini
Andrea Ostuni
Stefano Trepella

Candidate:

Maria Acciavatti

Academic Year 2025/2026

To those who defend the right to knowledge,
because knowledge is the highest form of freedom.
The one that disarms without wounding.

To the souls on a journey, who, like alchemists, transform
their suffering into love,
their sense of being lost into courage,
and their distraction into gratitude.

To the love that has always surrounded me.

Abstract

Socially aware navigation has become a key requirement for autonomous service robots operating in human-populated environments, as they are increasingly expected to navigate naturally and safely among people. Achieving this objective requires the integration of local motion planning strategies with reliable human perception. Navigation and perception are often investigated independently, although they are intrinsically connected. Indeed, the robot's socially compliant behavior depends on the robustness of the perception system capable of providing accurate and temporally consistent information about surrounding people. In particular, accurate human perception is essential for computing the social costs used by local planners to evaluate robot-human interactions and select the optimal trajectory for human-aware navigation. Motivated by this close interdependence, this thesis jointly addresses, within a robotic platform based on the Robot Operating System 2 (ROS 2), two complementary contributions: the improvement of social navigation capabilities and the development of a global perception system.

The first contribution concerns the enhancement of the Navigation2 Model Predictive Path Integral (MPPI) local controller by integrating an existing social critic based on the Social Force Model (SFM) to evaluate the social cost of sampled trajectories. Specifically, the MPPI controller configuration was optimized through extensive tuning and experimentally validated to improve trajectory smoothness, safety, and social compliance. The second contribution focuses on the development of a multi-camera human tracking framework based on three RGB-D cameras. Following a study of the optimal camera placement, a dedicated fusion node was designed and implemented to fuse the outputs of the cameras, each independently performing local detection through a tracking-by-detection pipeline based on YOLOv8 Pose and StrongSORT. Local detections from multiple viewpoints are combined into global tracks while ensuring long-term identity consistency, despite partial visibility, camera field-of-view overlaps, and temporary occlusions.

Finally, a dedicated validation methodology was developed to quantitatively assess the effectiveness of the proposed framework, evaluating identity stability together with complementary tracking performance metrics. The validation was performed using ROS bags recorded both in the Gazebo simulation environment and in the PIC4SeR laboratory with the Clearpath Jackal robot and a Vicon motion capture system providing ground-truth data. The resulting framework reinforces the reliability of both perception and navigation, providing a robust solution for autonomous service robots operating in dynamic environments.

Acknowledgements

I would like to express my gratitude to the PoliTO Interdepartmental Center for Service Robotics (PIC4SeR) for giving me the opportunity to carry out my thesis in such an interesting and inspiring context. This experience allowed me to explore amazing applications in the field of social robotics, to deepen my knowledge, and also to approach topics that were completely new to me. Throughout these months, I had the opportunity to challenge myself, turn theoretical concepts into practical experience, and face problems that often required perseverance and an even stronger desire to learn. Looking back, I feel that this experience has enriched me both academically and personally. I leave university life with much more knowledge, awareness, and confidence than when I first arrived. Thank you to my supervisors for their guidance and support throughout this work. Their advice and expertise helped me through the different stages of this thesis and contributed significantly to my growth during this experience. Thank you also to all the people I had the pleasure of meeting at PIC4SeR, for welcoming me with a smile from the very beginning, for always being kind and willing to help whenever I needed it.

I would also like to thank all the people I met throughout my Master's degree, and especially the friends with whom I shared this journey. You made these years not only easier, but infinitely more enjoyable. We laughed when everything started to feel too heavy, helped one another, and learned to work as a team to reach our goals. Between exams, projects, endless deadlines, and all the moments in between, we shared much more than a university course.

My deepest thanks go to my family. Thank you, Mum and Dad. You have always been, and will always be, my brightest light. Everything I am today, I owe to you. I know that you struggled with me whenever you saw me going through difficult moments, and I know that you were even happier than I was every time I managed to overcome them, reaching another milestone. You have been by my side through every step. No words will ever be enough to express how grateful I am for everything you have done for me and for the unconditional love you have always given me.

Thank you to my brother Emanuele, half of my heart living outside of me. You have always been my first supporter, even when you chose to remain silent or just to watch over me from far away. I know that you have never stopped believing in me, in my abilities, and in everything I have chosen to pursue. Knowing that you were there, even without needing to say it out loud, has always meant the world to me.

Thank you, Giulio. These university years have tested us in so many ways. It was not always easy to stand beside me when anxiety took over, yet you never allowed me to give up. You encouraged me, pushed me to give my best, and reminded me, especially when I could not see it myself, that beyond all the stress and the difficult moments, a new life was waiting for us; one filled with new experiences, achievements, and happiness to share together. Thank you for your care, for all the repeated hugs when one was simply not enough, and for being my lucky charm throughout this journey. I am glad to have you by my side.

Thank you, Enzo and Antonella, for celebrating every achievement as if it were your own, for never ceasing to support me, and for always reminding me of how much strength I had. Your affection and encouragement always made me feel at home and you have never stopped giving me kindness. I am grateful to have you in my life.

Thank you to my lifelong friends. You have been walking beside me through the different chapters of my life, always surrounding me with your affection and support. Life may sometimes take us far from one another, and we may not speak every day, but some bonds do not need constant words to remain strong. The love stays, despite the distance and the passing of time, and I know that I can always count on you.

Thank you to my angel, who always has my back. I feel you in the air. Sometimes, I recognize your scent and like to think that it is your way of letting me know that you are around me. And somehow, I know that you are still here, watching over me as you always did. Grandma Maria, this achievement is for you too.

Table of Contents

List of Tables VIII

List of Figures IX

1 Introduction 1
- 1.1 Socially Aware Navigation . . . 2
- 1.2 Human Perception for Social Navigation . . . 3
- 1.3 Multi-Camera Human Perception . . . 5
- 1.4 Robotic Software Platform . . . 6
- 1.5 Research Motivation . . . 7
- 1.6 Thesis Organization . . . 8

2 State of the Art 9
- 2.1 Social Navigation . . . 9
  - 2.1.1 Local Motion Planning Strategies . . . 10
  - 2.1.2 Socially Aware Navigation Mechanism . . . 11
  - 2.1.3 Hybrid Navigation Frameworks . . . 12
- 2.2 Human Perception for Service Robotics . . . 13
  - 2.2.1 Human Detection . . . 13
  - 2.2.2 Human Tracking . . . 14
  - 2.2.3 Multi-Camera Human Tracking . . . 15

3 Theoretical Background and Methods 17
- 3.1 ROS 2 Navigation Framework . . . 18
- 3.2 Dynamic Window Based Controller . . . 20
- 3.3 Model Predictive Path Integral Controller . . . 24
- 3.4 LightSFM-based Social Critic . . . 32
- 3.5 Visual Perception and Deep Learning . . . 35
- 3.6 People Detection and Pose Estimation . . . 38
- 3.7 Multi-Object Tracking . . . 40

4 System Design and Implementation 49
4.1 Multi-Camera Payload Design . . . 50
4.1.1 Sensor Configuration Requirements . . . 50
4.1.2 Two-Camera Configuration and Coverage Limitations . . . 51
4.1.3 Three-Camera Configuration Design . . . 54
4.2 Local Perception and Tracking Pipeline . . . 58
4.2.1 Single-Camera Tracking Pipeline . . . 59
4.2.2 Three-Camera Pipeline Deployment . . . 62
4.3 Fusion Strategy for Global Tracking . . . 63
4.3.1 Fusion Architecture and Track Representation . . . 64
4.3.2 Input Preprocessing and Observation Filtering . . . 66
4.3.3 Motion Prediction and Geometric Gating . . . 69
4.3.4 Multi-Cue Global Data Association . . . 74
4.3.5 Local Identity Continuity and Re-ID Memory Management . 80
4.3.6 Track Recovery and Global ID Preservation . . . 81
4.3.7 Track Lifecycle, Reliability, and Output Publication . . . 84
4.4 Socially Aware Navigation Configuration . . . 87
4.4.1 Integration of Human State Information . . . 88
4.4.2 Local Controller Critic Tuning . . . 89

5 Experimental Setup 91
5.1 Perception Models and Inference . . . 91
5.2 Hardware and Sensors . . . 92
5.2.1 Clearpath Jackal Mobile Platform . . . 92
5.2.2 Intel RealSense D435 RGB-D Camera . . . 93
5.2.3 Three-Camera Payload Implementation . . . 102
5.2.4 Vicon Motion Capture System . . . 103
5.3 ROS 2 Experimental Architecture . . . 105
5.3.1 Multi-Camera Data Flow and Topic Structure . . . 105
5.3.2 ROS Bag Data Acquisition . . . 106
5.4 Simulation Environment for Social Navigation Validation . . . 107

6 Experimental Validation and Results 109
6.1 Evaluation Methodology and Metrics . . . 109
6.1.1 Observability and Identity-Continuity Episodes . . . 110
6.1.2 Fusion-Specific Metrics . . . 111
6.1.3 Standard Multi-Object Tracking Metrics . . . 112
6.2 Dataset . . . 114
6.2.1 Simulation Dataset . . . 115
6.2.2 Real-World Dataset . . . 116
6.3 Simulation Results . . . 117

6.4 Real-World Results 126
6.5 Social Navigation Results 135

7 Conclusions 137

Bibliography 139

List of Tables

3.1 Trajectory generation parameters. 22
3.2 Main DWB critic plugins. 23
3.3 Robot kinematic parameters. 24
3.4 Main prediction, sampling, and optimization parameters of the Navigation2 MPPI controller. 29
3.5 Principal critic functions available in the Navigation2 MPPI controller [31]. 30
3.6 Main configuration parameters of the LightSFM-based Social Critic. 34

4.1 Comparison of the investigated two-camera configurations for different yaw orientations. 53
4.2 Geometric comparison of the investigated three-camera configurations for different lateral-camera yaw angles. 57

5.1 Main specifications of the Clearpath Jackal mobile platform. 93
5.2 Technical specifications of the Intel RealSense D435 [48]. 98
5.3 Reference-frame naming structure adopted for the center, left, and right Intel RealSense D435 cameras in the implemented ROS 2 configuration. 101

6.1 Fusion-specific metrics for the first simulation experiment. 118
6.2 Fusion-specific metrics for the second simulation experiment. 121
6.3 Fusion-specific metrics for the third simulation experiment. 125
6.4 Standard MOT metrics for the three simulation experiments. 126
6.5 Fusion-specific metrics for each participant in the first real-world experiment. 128
6.6 Identity-continuity episodes and dominant global IDs in the second real-world experiment. 131
6.7 Fusion-specific evaluation results for the second real-world experiment. 131
6.8 Fusion-specific evaluation results for the third real-world experiment. 133
6.9 Standard MOT metrics for the three real-world experiments. 134

List of Figures

1.1 Alba Robot wheelchair transporting passengers through an airport terminal [1]. 2
1.2 Dabo robot navigating in the Barcelona Robot Lab [5]. 3
1.3 Autonomous ground robot moving its way around people [4]. 3
1.4 Social-type aware navigation for comfort in social spaces [15]. 8

3.1 Overview of the Navigation2 architecture, from Macenski et al. [17]. 19
3.2 Representation of the Dynamic Window Approach search space [30]. 22
3.3 Comparison between image classification, object localization, and object detection [32]. 36
3.4 Comparison between object detection and instance segmentation [33]. 36
3.5 Example of anatomical landmark extraction for human pose estimation [34]. 37
3.6 Example of human pose estimation using YOLOv8-Pose [37]. 39

4.1 Clearpath Jackal platform dimensions [43]. 52
4.2 Top view of the sensor coverage provided by the final three-camera configuration in the Gazebo simulation environment. 58
4.3 Priority hierarchy adopted for survivor selection during duplicate suppression. 68

5.1 Clearpath Jackal mobile platform employed in the experimental activities at the PIC4SeR laboratory. 93
5.2 Example of the RGB and depth streams acquired by the Intel RealSense D435 during the physical experiments: (a) RGB image and (b) corresponding depth image. 94
5.3 Geometric model of binocular stereo vision used for depth estimation through triangulation. Reproduced from [49]. 96

5.4 Examples of images with and without the projected pattern: (a) infrared image without the projected pattern, (b) infrared image with artificial pattern, (c) depth estimation without the projected pattern, and (d) depth estimation using active stereo. Reproduced from [51]. 97
5.5 Optical components of the Intel RealSense D435, including the infrared stereo cameras, infrared projector, and RGB color camera. [52]. 99
5.6 Conventions adopted for ROS 2 and optical coordinate frames. . . . 100
5.7 Reference-frame hierarchy of the left Intel RealSense D435, generated using the ROS 2 `tf2_tools` package. 102
5.8 Physical implementation of the three-camera perception payload on the Clearpath Jackal, showing the 3D-printed mounting structure. . 103
5.9 Vicon Motion Capture setup. 104
5.10 Wearable U-shaped reflective-marker targets (`U_1`, `U_2`, and `U_3`) employed as pedestrian ground-truth references during the physical experiments. 104
5.11 Block diagram of the ROS 2 multi-camera perception architecture. . 105
5.12 Clearpath Jackal equipped with the simulated three-camera perception configuration in Gazebo. 107
5.13 RViz visualization of the simulated social navigation scenario. . . . 108

6.1 Gazebo simulation environment populated with HuNav agents. . . . 115
6.2 Participants wearing the U-shaped reflective-marker targets. 116
6.3 Temporal global-ID assignment for the first simulation experiment. 117
6.4 Standard Multi-Object Tracking performance metrics for the first simulation experiment. 119
6.5 Temporal global-ID assignment across the visibility episodes of the four agents in the second simulation experiment. 120
6.6 Cumulative ground-truth paths and global-ID assignments for the second simulation experiment. 121
6.7 Standard Multi-Object Tracking performance metrics for the second simulation experiment. 122
6.8 Temporal global-ID assignment across the visibility episodes in the third simulation experiment. 124
6.9 Temporal global-ID assignment across the visibility episodes in the first real-world experiment. 127
6.10 Global and ground-truth trajectories in the first temporal window: individual trajectory comparisons and combined representation. . . 129
6.11 Global and ground-truth trajectories in the second temporal window: individual trajectory comparisons and combined representation. . . 130

6.12 Global and ground-truth trajectories in the selected temporal window of the second real-world experiment: individual trajectory comparisons and combined representation. 132
6.13 Representative RViz snapshots of the simulated social navigation experiment. 136

Chapter 1

Introduction

In recent years, robotics has experienced remarkable growth, supported by rapid advances in artificial intelligence (AI), sensing technologies, and computing.
In particular, autonomous service robots have gained increasing importance in various application fields. This expansion is driven by the growing demand for autonomous and intelligent solutions, enabling robots to assist humans in everyday activities while improving efficiency and quality of service.
Indeed, autonomous mobile robots are employed in a wide range of service applications, including hospitals, airports, museums, shopping centers, and public buildings. One representative application is the autonomous mobility assistance for passengers with reduced mobility in airports. Self-driving electric mobility platforms, such as the autonomous robotic wheelchair developed by ALBA Robot, have been designed to transport passengers between different service areas. Airport terminals, as well as hospitals and museums, represent challenging indoor environments, due to large dimensions, dense pedestrian traffic, continuous changing human flows, and strict safety requirements.

Unlike traditional industrial robots, which operate in structured and isolated environments, the main challenge for service robots is to accomplish their tasks in dynamic spaces shared with humans. Consequently, autonomous navigation can no longer be limited to reaching a target position by following a geometrically collision-free planned path. Robots must move in a way to safely coexist and naturally interact with people moving around, being comfortable, predictable, and socially acceptable. These requirements have led to the development of socially aware navigation, which extends conventional autonomous navigation by explicitly considering human presence and social interactions during motion planning. In this context, nearby people are not simply considered as moving obstacles; they change direction, interact with one another, form groups, and react to the robot, whose

behavior is heavily influenced by human dynamics. Socially aware navigation, thus, requires several integrated capabilities: while planning the navigation, the robot should perceive the surrounding environment, estimate the human state in terms of position, velocity, and orientation, and continuously adapt its motion as the scene evolves. Among these capabilities, human perception plays a fundamental role by providing the information required to account for human presence, enabling the integration of human-related constraints and social conventions into the robot motion planning process, and ultimately contributing to reliable social navigation.



Figure 1.1: Alba Robot wheelchair transporting passengers through an airport terminal [1].

1.1 Socially Aware Navigation

Mobile robot navigation represents one of the main research fields in robotics, and numerous solutions with different approaches have been proposed during years to handle problems of increasing complexity. In particular, autonomous mobile robots are robots capable of performing desired tasks in both structured and unstructured environments without continuous human guidance, as stated in [2]. Within this context, autonomous navigation has traditionally focused on guiding mobile robots to move from an initial position to a desired destination while avoiding collisions with static and dynamic obstacles. As discussed by Levitt and Lawton [3], navigation and guidance can be described through three fundamental questions: "Where am I?" "Where are other places relative to me?" "How do I get to other places from here?". Since a robot that moves about the world must be able to compute answers to these questions, navigation systems rely on

environment perception, localization, mapping, and motion planning to generate collision-free trajectories and accomplish the navigation task successfully. However, the increasing interest and deployment of robots operating in crowded public environments has introduced new challenges for mobile robot navigation, leading to "socially aware robot navigation to emerge as a research field of its own", [4]. Conventional navigation strategies, primarily focusing on obstacle avoidance and path efficiency are no longer sufficient when robots operate alongside people. The reason is because achieving social compliance requires accounting for the fact that human's behavior is continuously influenced and might be changed by personal intentions, interactions with other people, environmental context, social rules, and the motion of the robot itself. "Socially appropriate behavior is key for a robot to gain acceptance from humans and prevent causing any discomfort" [4].



Figure 1.2: Dabo robot navigating in the Barcelona Robot Lab [5].



Figure 1.3: Autonomous ground robot moving its way around people [4].

Accordingly, socially aware navigation seeks to generate trajectories that allow the robot to accomplish the navigation task, while minimizing disturbances to nearby people and promoting natural human-robot interaction. The objective is thus to make robots navigate in a manner that guarantees human safety, comfort, and predictable behavior.

1.2 Human Perception for Social Navigation

The availability of continuous knowledge about surrounding people is crucial for socially aware navigation. Indeed, human perception is recognized as one of the essential capabilities of service robots operating in dynamic environments. The focus is perceiving nearby people and estimate their state, collecting the necessary information to produce a socially acceptable behavior.

In the last years, rapid development of computer vision has profoundly transformed autonomous robot's perception. Modern visual algorithms provide robots with the capability of interpreting visual information captured by onboard sensors and estimating the state of the surroundings. According to Szeliski, computer vision aims at extracting meaningful information from images and videos in order to build a representation of the observed scene [6]. In mobile robotics, this capability enables robots to identify relevant objects and continuously monitor human presence and motion, sending useful information to high-level decision-making and navigation modules.

Human perception for service robots consists essentially of two interconnected tasks: human detection and human tracking. The former aims at identifying the presence and instantaneous location of humans within sensor observations, whereas the latter associates consecutive observations belonging to the same individual over time, allowing the continuous estimation of the person's state [7, 8]. As stated by Bellotto and Hu, "One of fundamental issues for service robots is human–robot interaction. In order to perform such a task and provide the desired services, these robots need to detect and track people in the surroundings" [7]. These two processes, thus, are of paramount importance for the final goal of social compliance. Beyond estimating the position of each pedestrian in the explored world, tracking allows the robot to infer dynamic quantities such as velocity, orientation, trajectory evolution, and identity consistency over time [8, 9]. Together, these quantities define the human state that is exploited in navigation algorithms to predict future pedestrian motion and continuously adapt the robot behavior accordingly [4, 8].

Although technology advances gained in recent years are remarkable, achieving reliability in human perception remains a challenging research problem. Numerous factors weaken the process such as frequent occlusions, cluttered scenes, sensor noise, and the simultaneous motion of both humans and the robot itself, that may lead to missed detections, inaccurate localization, and identity inconsistencies during tracking. These limitations directly affect the quality of the estimated state and, consequently, the effectiveness of socially aware navigation [8, 9]. These challenges become even more difficult when perception relies on a single onboard camera. The limited field of view (FOV) results in a restricted observable area, further causing the interruption of continuous observation of a target, when temporary occlusions due to environmental structures or crossing pedestrian trajectories occur. Avoiding fragmented trajectories by providing a robust and continuous human perception through a single viewpoint is still a tough task. For this reason, multi-camera perception has become a popular adopted solution in mobile robotics. By combining complementary observations acquired from multiple viewpoints, these systems enlarge the observable area, reduce the impact of occlusions, and improve the robustness and temporal continuity of human tracking [9, 10].

1.3 Multi-Camera Human Perception

The adoption of multi-camera perception represents the natural evolution of modern robotic perception systems. Rather than relying on a single visual stream, multiple cameras enable the simultaneous observation of the scene from different viewpoints, obtaining a more comprehensive representation of the surrounding environment. As observed by Amosa et al., "While it is preferable for most visual surveillance systems to have a wider view of targets of interest, multiple cameras are often employed" [10]. This principle is applicable not only in the mentioned context of intelligence surveillance, but also in all those applications where reliable human monitoring is essential, including autonomous driving as well as socially aware mobile robotics.

The integration of different observations improves the robot's ability to continuously perceive people in the surrounding, getting also a more complete understanding of the visited places. As highlighted in the Springer Handbook of Robotics, perception "allows the integration of sensor information across space and time to facilitate planning" [11]. Consequently, the availability of complementary visual information provides a richer perception of the environment that can support higher-level robotic functions.

However, if on the one hand, the employment of a multi-camera payload offers advantages by providing wider environmental awareness and richer visual information, on the other hand it also increases the complexity of the overall perception system. Additional sensors imply higher hardware costs associated with camera acquisition, installation, calibration, and maintenance. Moreover, multiple image streams must be processed simultaneously, significantly increasing computational effort, memory consumption, communication bandwidth, and energy demand. One of the major challenges is making the perception framework efficiently manage and integrate information originating from different sensors while preserving real-time performance. Additionally, to further underline the complexity of such solution, the different observations cannot be processed independently, but they must be consistently integrated and treated as a unified representation. As discussed by Rinner and Quaritsch, "it is necessary to correlate the data somehow—that is, to fuse the information from individual sensors" [12]. Consequently, effectively integrating independent observations into a coherent representation of the surrounding environment has become one of the key research challenges in modern multi-camera perception systems.

1.4 Robotic Software Platform

Modern robotic systems require a software platform capable of integrating heterogeneous sensors, communication interfaces, control algorithms, navigation frameworks, perception modules, within a unified software architecture. In particular, robotic applications are no longer developed from scratch as isolated programs; instead, their development relies on dedicated robotic software platforms that provide a common infrastructure for software integration and communication. Such platforms must simplify the development of complex robotic applications by providing standardized communication mechanisms between software components, modular organization, and a rich ecosystem of reusable tools and libraries, thus facilitating software integration, testing, and deployment. Among the available solutions, the Robot Operating System (ROS) has become the de facto standard framework adopted by both the robotics research community and industry [13, 14]. As originally defined by Quigley et al., "ROS is an open-source Robot Operating System" [13]. Its primary objective is "to support code reuse in robotics research and development" [13], providing developers with a modular software framework for building distributed robotic applications.

The continuous robotics evolution has led to the development of ROS 2, specifically designed to address the requirements of modern robots. As stated by Macenski et al., the new generation of the framework was designed to extend the operating system towards applications involving “production engineering, distributed systems, embedded computing, and real-time systems" [14], providing "a modular, scalable, and reliable architecture" [14]. Specifically, compared with the original framework, ROS 2 introduces a fully distributed communication architecture based on the Data Distribution Service (DDS) middleware, eliminating the need for a centralized master node. Furthermore, its support for configurable Quality of Service (QoS) policies enables communication to be tailored according to application requirements, allowing developers to balance reliability, latency, and communication overhead. The operating system is additionally characterized by native cross-platform support and an improved scalability, making it suitable for complex modern applications. As a matter of fact, throughout this thesis work, ROS 2 served as the common software platform integrating the Jackal robot model, the Navigation2 stack, the perception framework, and the experimental tools, such as Gazebo and Rviz, adopted for simulation and visualization during the experimental testing of the proposed solution.

1.5 Research Motivation

As previously discussed in detail, a multi-camera system significantly enhances robot's perception capabilities, though it also entails greater computational challenges. In mobile robotics, the use of a multi-camera system aims to achieve more than just the autonomous detection or tracking of people within each camera's field of view; indeed, observations are processed to meet the need for a coherent and temporally stable representation of surrounding individuals, regardless of which sensor generated the observation. Hence, for autonomous navigation requirements the perception system is expected to provide a unified description of the scene, rather than just a collection of independent camera observations. This objective becomes particularly complicated because the robot continuously moves through dynamic environments while people simultaneously enter, leave, or traverse the fields of view of different cameras. Consequently, observations originated from different viewpoints must be correctly associated, merged, and maintained over time, while solving problems such as duplicated representations, identity instability, crossing human trajectories, and temporary occlusions. Although many multi-camera tracking approaches have been proposed in the literature over the last decades, maintaining a reliable global representation that satisfies the requirements of real-time mobile robot navigation remains an open challenge. This is because the further complexity derives from the additional request of stable identity associations and thus continuous trajectories, beyond maximizing tracking and position estimation accuracy to provide correct information to be directly exploited for navigation. These requirements become even more critical in socially aware navigation, where the robot continuously reasons about the position and motion of nearby pedestrians while planning safe and socially acceptable trajectories.

These considerations motivated the work presented in this thesis. The objective is to develop a perception framework capable of transforming independent local observations into a coherent global representation of the surroundings. Such a representation should provide the navigation system with reliable information, while preserving real-time requirements of modern robotic applications.

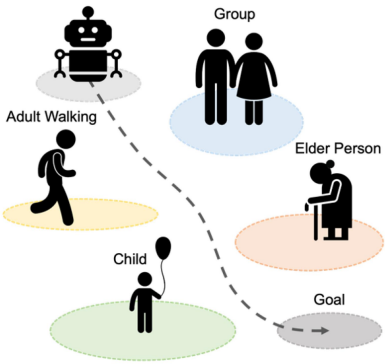


Figure 1.4: Social-type aware navigation for comfort in social spaces [15].

1.6 Thesis Organization

An overview over the thesis organization is given in the following, by briefly introducing the content of each chapter.

Chapter 1 introduces the research context. It discusses the role of socially aware navigation, the importance of human perception, the challenges of multi-camera perception systems, and the motivation behind the thesis work.

Chapter 2 reviews the state of the art in socially aware navigation and human perception for mobile robotics, presenting the principal approaches proposed in the literature and discussing representative research works.

Chapter 3 illustrates the main navigation and tracking algorithms. It is worth to give first the reader a theoretical background about the principal technologies employed, comparing them.

Chapter 4 presents the design choices and the adopted methodology for the development of the proposed multi-camera perception system.

Chapter 5 describes the experimental setup adopted to validate the proposed framework.

Chapter 6 presents the adopted performance metrics and discusses the experimental results obtained during the validation phase through both quantitative and qualitative analyses.

Chapter 7 concludes the thesis by providing some considerations and outlining possible directions for future work.

Chapter 2

State of the Art

This chapter discusses the current state of the art in socially aware robot navigation and human perception for service robotics. The aim is to provide an overview of the most widely adopted approaches by the scientific community, highlighting their operating principles, advantages, and limitations. In particular, the focus is on local navigation strategies, socially aware navigation mechanisms, and human perception techniques, as these constitute the technological basis of the framework proposed in this thesis.

2.1 Social Navigation

Many different approaches exist to address the social navigation task differing in both the local navigation strategy employed to determine the robot motion and the kind of mechanism used to account for human presence during navigation. The literature comprises a broad spectrum of solutions, ranging from conventional reactive planners to predictive optimization-based controllers, as well as approaches based on explicit models of social interaction and adaptive learning techniques. Hence, rather than being based on a single navigation paradigm, recent research investigates possible solutions to combine local planning algorithms with modules for human behavior and social interaction. This basically means that social navigation has evolved into a multidisciplinary research field integrating concepts from mobile robotics, control theory, artificial intelligence, and human-robot interaction. Accordingly, the following discussion is organized into three directions: the first looks at the main local navigation strategies; the second analyzes the methods to integrate social awareness into the navigation process; the third explores hybrid navigation frameworks.

2.1.1 Local Motion Planning Strategies

Local planning is a fundamental operation within the robot navigation system. Unlike the global planner which computes the global path from the robot starting position up to the goal, avoiding collisions through the use of a global representation of the environment, the local planner is instead responsible for generating feasible motion commands that make the robot safely follow the planned path, while continuously adapting to the dynamic evolutions in the environment. Operating over a limited prediction horizon, the local planner relies on real-time sensor observations to promptly react to nearby obstacles, while simultaneously satisfying the robot kinematic and dynamic constraints. This hierarchical navigation architecture, based on the interaction between global and local planning, has become the standard paradigm adopted by modern robotic navigation systems [16, 17].

Besides meeting real-time requirements and ensuring to promptly adjust the behavior in case of sudden changes, the design of an effective local motion planner must also guarantee the generation of smooth and dynamically feasible trajectories. The different proposed methods for local planning can be classified as follows:

- reactive methods
- optimization-based methods
- sampling-based predictive control methods

Each category follows a different strategy for generating robot motion and balancing computational efficiency with trajectory quality.

Reactive methods compute control commands directly from the current robot state and sensor observations, enabling fast responses to dynamic changes with limited computational requirements. Representative algorithms include the Dynamic Window Approach (DWA) and its following implementation, the Dynamic Window Based (DWB) controller, which remain widely adopted in robotic navigation due to their simplicity, lower computational effort, and quite satisfactory real-time performance [17, 18]. Optimization-based methods generate robot trajectories by minimizing an objective function over a finite prediction horizon while satisfying robot kinematic and dynamic constraints and accounting for environmental issues. Representative examples include the Timed Elastic Band (TEB) planner and Model Predictive Control (MPC), both of which explicitly optimize the robot motion aiming at improving the trajectory smoothness, feasibility, and adaptability [19, 20]. More recently, sampling-based predictive control techniques have emerged as

a promising solution for complex and highly dynamic environments. Instead of explicitly solving an optimization problem, these methods evaluate a large number of sampled control sequences to approximate the optimal solution. The Model Predictive Path Integral (MPPI) controller is one of the most representative algorithms of this category and shows itself as an advanced technique since it naturally accommodates nonlinear system dynamics, non-convex optimization problems, and arbitrary cost functions, while offering high computational performance and flexibility [21]. In particular, predictive sampling-based controllers such as MPPI have gained increasing attention for service robots operating alongside humans.

2.1.2 Socially Aware Navigation Mechanism

Regardless of the underlying principle, local planning strategies aim at generating feasible trajectories, but they do not inherently guarantee socially acceptable robot behavior. This means that even though the resulting local planning allows the robot to avoid obstacles, the robot may invade personal spaces, interfere with ongoing human interactions, or produce motions perceived as uncomfortable. For this reason, some studies have focused on introducing specific mechanisms to take care about humans and social conventions during motion planning. Accordingly, based on the mechanism employed to introduce social awareness, the main approaches considered in this discussion can be grouped into:

- cost-based
- interaction-based
- learning-based

Cost-based approaches enrich the planner with additional social cost functions. Rather than modifying the planning algorithm itself, these methods include additional penalties with human-related costs associated with proxemic constraints, pedestrian orientation, walking direction, or predicted occupancy, encouraging trajectories to preserve comfortable interpersonal distances, remaining safe and efficient. This strategy has been successfully integrated into several local planners, including DWA, TEB, MPC and MPPI.
Interaction-based techniques, instead, explicitly model the interactions between pedestrians and robots, thus describing how they influence each other's motion during navigation. Basically, these methods attempt to capture the dynamic nature of human behavior, allowing the robot to anticipate future interactions and generate more natural navigation behaviors. Representative examples include the Social Force Model (SFM) and its subsequent extensions, which have gained,

among interaction-based approaches, considerable attention in social navigation [22]. Finally, learning-based approaches make use of machine learning techniques to learn socially compliant navigation policies directly from interactions with the environment. They often do not generate a trajectory, but compute velocity commands using machine learning algorithms. Examples are given by reinforcement learning and imitation learning, which have become popular thanks to their ability to autonomously capture complex human behaviors without relying on manually designed interaction models.

2.1.3 Hybrid Navigation Frameworks

Recently the trend is to shift towards hybrid navigation frameworks taking advantage by the combination of different methodologies. There exist instances in literature about solutions integrating predictive motion planning, explicit models of human behavior, and adaptive learning techniques to improve navigation efficiency and social compliance.

A possible direction consists of enhancing socially aware planners with learning-based components. Machine learning techniques can be employed to adapt planner parameters or optimize cost function weights, without the need to replace the underlying navigation algorithm. An example is the AdaScore framework proposed by Martini et al. [23], which builds upon the Social Force Window (SFW) planner. SFW combines the local navigation strategy of DWA with a social cost derived from the Social Force Model (SFM). The approach employs a deep reinforcement learning (DRL) agent to dynamically adjust the weights of the planner's cost function according to the current navigation context. In this way, the planning strategy is preserved and enhanced, allowing it to adapt its behavior to different crowd configurations.

Other research works combine local planners with explicit interaction models describing pedestrian behavior. Instead of representing humans only through geometric constraints, human-robot interactions are directly modeled and integrated into the planner objective. A representative example is LightSFM [24], a lightweight implementation of the Social Force Model designed to represent pedestrian interactions through attractive and repulsive social forces. Its lightweight formulation makes it suitable for integration as an additional social interaction component within local controllers, such as MPPI.

Another emerging direction consists of extending predictive controllers with uncertainty estimation and dynamic risk assessment. Rather than considering only the current positions of surrounding pedestrians, these approaches explicitly

account for future motion uncertainty and collision risk during trajectory planning. Recent developments such as Dynamic Risk-Aware MPPI (DRA-MPPI) have shown promising results for navigation in dense crowds [25].

2.2 Human Perception for Service Robotics

Current human perception systems for mobile robots are generally organized as multi-stage pipelines, where human detection and human tracking are the two fundamental processing tasks. In particular, the rapid evolution of deep learning has deeply transformed the technological trend and significantly improved the performance of both tasks, leading to robust perception frameworks that can be employed in challenging environments with satisfactory outcomes.

2.2.1 Human Detection

As a fundamental component of the perception system, human detection provides information about the presence and location of people within the surrounding environment. In this context, different sensing technologies can be employed, each providing different and complementary information about the observed scene. As discussed in the survey by Picello et al. [26], current pedestrian perception approaches for robotic navigation exploit different sensing modalities, including visual, LiDAR, thermal, and multimodal sensing. Accordingly, the main approaches considered in this discussion can be grouped into:

- vision-based
- LiDAR-based
- thermal-based
- multimodal approaches

Vision-based approaches exploit RGB cameras, providing rich appearance information that supports the recognition and characterization of pedestrians. Their relatively low cost, high image resolution, and compatibility with modern deep learning models make them particularly suitable for robotic applications. However, conventional RGB images do not directly provide metric depth information, and their effectiveness can be affected by illumination variations, occlusions, and adverse environmental conditions. For this reason, RGB-D sensors can represent a more

powerful solution, since they combine visual and depth information, allowing the appearance of detected pedestrians to be associated with their spatial position. This characteristic facilitates both visual recognition and three-dimensional localization. LiDAR-based approaches, on the other hand, provide accurate geometric and distance information about the surrounding environment and are less dependent on illumination conditions, although they do not provide the rich visual appearance information available from cameras.
Thermal cameras represent an additional sensing solution, especially useful under poor illumination conditions, although their generally lower image resolution and sensitivity to environmental heat sources can limit their applicability.
Finally, multimodal approaches combine complementary information acquired from different sensing technologies, such as cameras and LiDAR, with the objective of increasing the robustness of pedestrian perception under challenging operating conditions [26].
Among these sensing alternatives, vision-based human detection has experienced remarkable advances due to the development of deep learning techniques, which have considerably improved the capability of perception systems to detect humans in complex environments while progressively enabling real-time operation.

According to the survey presented in [27], current deep learning-based object detectors can be classified into:

- two-stage detectors
- one-stage detectors

The first category, including Faster R-CNN family, prioritizes detection accuracy by first generating candidate object regions, before performing classification and localization. Conversely, one-stage detectors, like YOLO, SSD, RetinaNet, directly predict object locations and classes in a single forward pass, achieving significantly higher inference speed and making them suitable for real-time robotic applications. Among these detectors, the YOLO models have become one of the most preferred solutions in autonomous robotic systems, due to the favorable trade-off between accuracy and computational speed. The latest versions further extend the detection process integrating pose estimation and instance segmentation, while preserving the favorable balance between detection accuracy and inference speed.

2.2.2 Human Tracking

Human tracking seeks to maintain temporal consistency of detected individuals by associating observations across consecutive frames and preserving their identities

over time. In modern literature, multiple object tracking (MOT) systems are primarily based on the tracking-by-detection paradigm, where object detection and temporal data association are treated as two separate stages. This formulation has become the dominant paradigm for multiple object tracking due to its flexibility and compatibility with modern deep learning detectors [27, 8].

Early tracking-by-detection algorithms relied on motion prediction and geometric association, with the SORT algorithm being one of the first real-time adopted solutions for multiple object tracking. To overcome limitations arising from occlusions and identity switches, new approaches have introduced appearance-based information through person re-identification techniques, giving rise to algorithms such as DeepSORT. More recent trackers, including StrongSORT and BoT-SORT, further improve identity consistency by combining robust appearance descriptors with advanced motion estimation and data association strategies.

2.2.3 Multi-Camera Human Tracking

Despite the remarkable progress achieved in single-camera multi-object tracking, extending the objective of identity consistency across multiple cameras remains a tough problem, given the additional requirement about the association of observations from different viewpoints sometimes in adverse operating conditions. The current literature proposes different strategies for cross-camera data association:

- appearance-based
- geometry-based
- hybrid methods

Appearance-based approaches primarily exploit deep person re-identification models to associate targets across different cameras according to their visual descriptors. These methods have become a dominant solution adopted by the scientific community thanks to their high robustness against appearance variations and are employed in many recent tracking systems reported in [10].
Geometry-based methods, instead, exploit geometric consistency and spatial information to establish target correspondences. Hence, these methods leverage camera calibration, three-dimensional localization, and motion consistency to improve cross-camera identity preservation. More recently, rather than relying on a single source of information, the proposed systems further combine geometric constraints with appearance cues, leading to hybrid frameworks capable of enhancing cross-camera association.

Beyond the type of information employed to establish cross-camera correspondences, different strategies can be adopted to organize and formulate the association process. While some methods directly associate individual detections, other approaches first group consecutive observations of the same target within each camera into short trajectories, commonly referred to as tracklets. Cross-camera association is then performed between these tracklets, allowing the temporal information accumulated over multiple frames to be exploited for identity matching. These approaches are commonly known as tracklet-based.
Moreover, the association problem can also be formulated through graph-based approaches, where detections or tracklets are represented as nodes and possible correspondences between them as edges. An optimization procedure can then be employed to determine a globally consistent set of associations across different observations, time instants, and camera views [10].

This reflects a broader trend towards hybrid multi-camera tracking approaches integrating appearance descriptors, geometric reasoning, and temporal motion information within a unified association strategy, reflecting the stronger and stronger need for robust long-term identity consistency in realistic scenarios characterized by frequent occlusions, crowded environments and camera transitions. Representative examples include the online multi-camera tracking framework proposed by Xie et al. [28], which combines person Re-ID with geometric consistency. A similar multi-cue strategy can be observed in the more recent RockTrack framework [29], where geometric and appearance information is jointly exploited for data association in multi-camera 3D object tracking.

Chapter 3

Theoretical Background and Methods

It is worth providing the reader with some theoretical and methodological background to have a deeper knowledge and to better understand the navigation and perception frameworks over which this thesis work has been developed. With this aim, the chapter introduces the technical working principles and the implementation details of the algorithms of interest.

The first part addresses autonomous navigation, introducing the ROS 2 Navigation2 framework and subsequently analysing the DWB and MPPI local controllers, with particular attention to their trajectory generation and evaluation mechanisms, highlighting the main differences between their control strategies. The social component integrated into the local navigation logic is then introduced through the Social Force Model and its lightweight implementation, LightSFM. Interaction forces are employed in the local planning process through a dedicated social critic, allowing human–robot interactions to contribute to the evaluation of candidate trajectories.

The second part focuses on human perception, describing the principles underlying YOLOv8–Pose for human detection and pose estimation, followed by the analysis of the tracking-by-detection paradigm adopted by modern multiple-object tracking algorithms. In this context, DeepSORT, StrongSORT and BoT–SORT are examined to provide the required theoretical basis for the multi-camera tracking framework and preparing to the global filtering system developed in this work.

3.1 ROS 2 Navigation Framework

Robot navigation is implemented through Navigation2 (Nav2), the autonomous navigation framework developed for ROS 2 as the successor to the original ROS Navigation Stack. It was introduced by Macenski in 2020 [17] with the objective of providing a modular and extensible architecture capable of supporting different robotic platforms, sensing technologies and navigation algorithms. Its modular architecture enables different planning, control, perception, and recovery algorithms to be integrated within the same software infrastructure. Hence, Nav2 decomposes the overall navigation task into specialized functionalities.

The organization of Nav2 is based on two main architectural concepts: a Behavior Tree Navigator and a set of task-specific servers. The Behavior Tree (BT) constitutes the high-level navigation logic orchestrating the individual operations and coordinating the whole navigation task. It defines the sequence of actions to be performed, such as computing a path towards a goal, following the generated path, monitoring the navigation progress, and activating recovery behaviors when normal execution cannot proceed. Rather than directly implementing these operations, BT action nodes communicate with the corresponding Nav2 servers through ROS 2 interfaces, requesting the required functionalities. This separation allows the high-level navigation logic and the algorithms used to accomplish individual navigation tasks to evolve independently.

A fundamental characteristic of Nav2 is its plugin-based architecture. At the execution level, each task-specific server is implemented as a ROS 2 node hosting one or more algorithm plugins, which can be loaded and configured through standardized interfaces. This allows different implementations of the same navigation functionality that can be selected, without modifying the overall system architecture. This also means that the navigation pipeline is not bound to a specific planning or control algorithm, but the strategy can be changed according to the needs and the operating characteristics of the environment, by replacing or reconfiguring the corresponding plugins. This modular organization is particularly relevant for experimental robotics, since different algorithms can be evaluated within the same navigation framework without performing a complete redesign of the navigation stack.

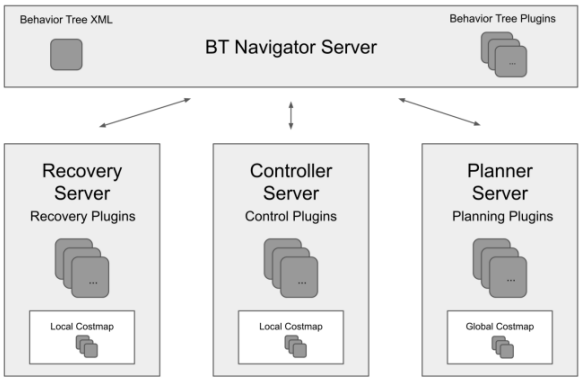


Figure 3.1: Overview of the Navigation2 architecture, from Macenski et al. [17].

The navigation servers are implemented according to the ROS 2 managed–node, or lifecycle node, model. Lifecycle nodes are characterized by a predefined state machine that regulates their initialization and execution through controlled transitions between states such as *unconfigured*, *inactive*, *active*, and *finalized*. Unlike conventional ROS 2 nodes, which typically begin execution immediately after being started, lifecycle management separates resource initialization and configuration from active task execution. This mechanism allows resources and communication interfaces to be initialized before the navigation algorithms become operational. It provides deterministic control over the bring-up and shutdown of the navigation system and ensures that the interconnected components are correctly configured before navigation begins.

Among the main components of this architecture are the Planner Server and the Controller Server. The former is responsible for global path computation. It invokes the selected planner plugin to generate a feasible path from the current robot pose up to a specified navigation goal. The resulting path provides a reference for the subsequent local motion control performed by the Controller Server. Specifically, once the global path is received, the selected controller plugin is executed at a higher frequency, continuously computing appropriate robot commands according to the current robot state and the locally perceived environment. The separation between these two functionalities allows global path planning and local motion control to be independently configured according to different algorithmic principles. Both global planning and local control require an explicit representation of the navigable environment. Nav2 provides this information through two-dimensional costmaps, in which the workspace is discretized into cells and each cell is assigned a cost representing the difficulty or risk associated with traversing that region. High costs are associated with occupied or undesirable areas, whereas low-cost regions correspond to free navigable space. This representation allows planning and control

algorithms to reason about environmental constraints through a common numerical representation. The two costmap instances provided are:

- global costmap
- local costmap

The global costmap represents the environment over the spatial extent required for global path planning and it is what the Planner Server uses when computing a global path. The local costmap, instead, represents a limited region surrounding the robot and is continuously updated using current sensor observations. It thus provides the Controller Server with an updated representation of nearby obstacles and environmental changes, supporting reactive local motion generation. Costmaps follow a layered architecture, in which different components independently contribute to the final cost representation. This architecture includes the Static Layer, giving occupancy information from an available map; the Obstacle Layer, which introduces obstacles detected by the robot sensors; and the Inflation Layer, which propagates costs around occupied cells to discourage robot motion too close to obstacles. The same plugin-based mechanism also enables additional application-specific layers to be introduced, allowing the environmental representation to be extended with information required by particular navigation tasks.

According to what has been discussed so far, it can be stated that the resulting Nav2 architecture separates three fundamental aspects of autonomous navigation: task coordination, handled through Behavior Tree; environmental representation, provided by costmaps and perception components; and execution of algorithms, performed by the plugins hosted within the navigation servers.

3.2 Dynamic Window Based Controller

The Dynamic Window Based (DWB) controller is the default local controller available in Navigation2 and constitutes the ROS 2 implementation of the Dynamic Window Approach (DWA), originally proposed by Fox, Burgard and Thrun. It preserves the fundamental principle of velocity-space search strategy, meaning looking for feasible robot motions in the allowed velocity space, while extending the controller through a modular architecture based on configurable trajectory generators and critic plugins. This extension of the original formulation enables the controller to be adapted to different robotic platforms and scenarios. The DWB controller, therefore, searches directly in the velocity space rather than in the entire robot configuration space. The set of admissible translational and

rotational velocity commands is first constrained by the robot kinematic limits and subsequently restricted according to the current velocity and the admissible acceleration and deceleration limits. The resulting subset of dynamically reachable velocity commands defines the so-called dynamic window, as explained in the following.

The robot control input can be expressed as

$$\mathbf{u} = \begin{bmatrix} v \\ \omega \end{bmatrix}, \tag{3.1}$$

where v and ω denote the translational and angular velocities, respectively.

First, the admissible velocity space is bounded by the robot kinematic limits

$$\begin{aligned} v_{\min} \leq v \leq v_{\max}, \\ \omega_{\min} \leq \omega \leq \omega_{\max}. \end{aligned} \tag{3.2}$$

Considering the current robot velocity (v_t, ω_t), the admissible velocity space is further restricted by the acceleration and deceleration limits. Over a control interval Δt, the dynamically reachable velocities are bounded by both the robot velocity limits and the velocities that can be reached from the current state:

$$\begin{aligned} v \in \left[\max\left(v_{\min},\, v_t - a_{v,\mathrm{dec}}\Delta t\right), \min\left(v_{\max},\, v_t + a_{v,\mathrm{acc}}\Delta t\right)\right], \\ \omega \in \left[\max\left(\omega_{\min},\, \omega_t - a_{\omega,\mathrm{dec}}\Delta t\right), \min\left(\omega_{\max},\, \omega_t + a_{\omega,\mathrm{acc}}\Delta t\right)\right]. \end{aligned} \tag{3.3}$$

Here, $a_{v,\mathrm{acc}}$ and $a_{v,\mathrm{dec}}$ denote the admissible linear acceleration and deceleration magnitudes, respectively, while $a_{\omega,\mathrm{acc}}$ and $a_{\omega,\mathrm{dec}}$ denote the corresponding angular acceleration and deceleration magnitudes. The resulting intervals therefore define the dynamically reachable portion of the admissible velocity space considered during the current control cycle.

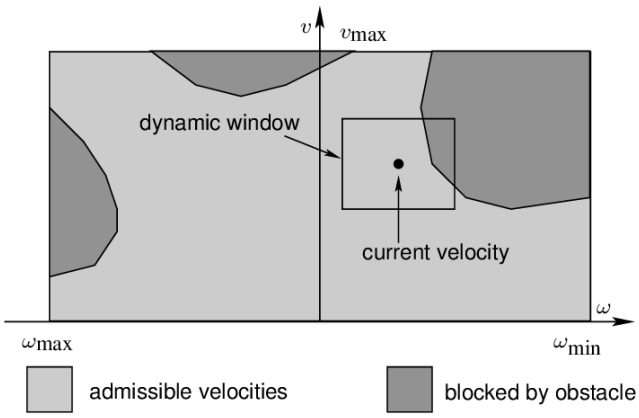


Figure 3.2: Representation of the Dynamic Window Approach search space [30].

At each control cycle, the controller uniformly samples the admissible velocity space, generating a set of candidate velocity commands. Each sampled command, represented by the admissible velocity pair (v, ω), is forward simulated according to the robot kinematic model over a finite prediction horizon, resulting in a collection of feasible local trajectories.

The trajectory generation process is, thus, controlled by a set of configurable parameters (illustrated in Table 3.1) defining the prediction horizon and the sampling density of the velocity space.

Parameter	**Description**
`sim_time`	Prediction horizon
`vx_samples`	Number of sampled linear velocities
`vtheta_samples`	Number of sampled angular velocities
`linear_granularity`	Linear trajectory discretization
`angular_granularity`	Angular trajectory discretization

Table 3.1: Trajectory generation parameters.

Since every produced trajectory satisfies the robot motion constraints, the controller only needs to determine which trajectory best satisfies the desired navigation objectives. Rather than selecting trajectories according to a single optimization criterion, DWB evaluates them through a collection of independent critic plugins. Each critic measures a specific aspect of the navigation behavior, such as obstacle avoidance, adherence to the global path, goal convergence, or trajectory smoothness.

The overall trajectory score is computed as the weighted combination of the critic evaluations

$$J(\tau) = \sum_{i=1}^{N} w_i J_i(\tau), \tag{3.4}$$

where $J_i(\tau)$ denotes the score assigned by the i-th critic to the candidate trajectory τ, while w_i represents its corresponding weighting factor. The trajectory associated with the minimum overall cost is finally selected and converted into the velocity command to be sent to the robot.
The most commonly adopted critics are summarized in Table 3.2.

Critic	**Navigation objective**
`BaseObstacle`	Obstacle avoidance
`PathAlign`	Path heading alignment
`PathDist`	Distance from global path
`GoalAlign`	Goal heading alignment
`GoalDist`	Distance to goal
`PreferForward`	Forward motion preference
`RotateToGoal`	Final orientation
`Oscillation`	Oscillation suppression

Table 3.2: Main DWB critic plugins.

This described formulation represents one of the main differences between DWB and the original DWA implementation. The reason is that, differently from DWA, which employs a fixed objective function combining a number of navigation criteria, DWB introduces a fully modular evaluation architecture in which individual critics can be independently enabled, disabled, or reconfigured without modifying the underlying planning algorithm. Consequently, different navigation behaviors can be achieved simply by changing the active critics or adjusting their relative weights.

Within the Navigation2 framework, the DWB controller is implemented as the `DWBLocalPlanner` plugin hosted by the Controller Server. The controller receives the global path together with the current robot state and the local costmap. Candidate trajectories are generated by the selected trajectory generator and subsequently evaluated by the enabled and properly configured critic plugins, before the best motion command is published on the robot velocity interface. Furthermore, the controller behavior can also be configured through robot kinematic constraints defining the admissible velocity space introduced by the Dynamic

Window formulation. The main configuring parameters are illustrated in the Table 3.3.

Parameter	Description
`max_vel_x`	Maximum forward velocity
`min_vel_x`	Minimum linear velocity
`max_vel_theta`	Maximum angular velocity
`acc_lim_x`	Maximum linear acceleration
`decel_lim_x`	Maximum linear deceleration
`acc_lim_theta`	Maximum angular acceleration
`decel_lim_theta`	Maximum angular deceleration

Table 3.3: Robot kinematic parameters.

3.3 Model Predictive Path Integral Controller

The Model Predictive Path Integral (MPPI) controller is a sampling-based stochastic optimal control method that combines predictive motion simulation with trajectory optimization. In particular, it determines the robot motion by evaluating the predicted future evolution of the system over a finite time interval, analysing the consequences of possible control actions before applying them to the robot [21]. The finite time interval over which the system evolution is predicted is referred to as the prediction horizon.

Unlike the DWB controller, which samples velocity commands within the admissible velocity space and forward simulates the corresponding local trajectories, MPPI operates on sequences of control inputs defined over a prediction horizon. This allows the effects of a sequence of future control actions to be evaluated over the predicted evolution of the robot. At each control cycle, the sequence of control inputs is optimized over the prediction horizon, but only the first control action of the resulting optimized sequence is applied to the robot, according to the receding-horizon principle. At the following iteration, the horizon is shifted forward, and the optimization process is repeated using the updated robot state and the available environmental information. This continuous re-optimization strategy, combining future motion prediction with feedback from the current robot state and environmental conditions, enables the controller to adapt its control action as the navigation scenario evolves.

Considering a differential-drive mobile robot, the state of the system at discrete time step k can be represented as

$$\mathbf{x}_k = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix}, \tag{3.5}$$

where x_k, y_k, and θ_k define the robot position and orientation, respectively. The corresponding control input is expressed as

$$\mathbf{u}_k = \begin{bmatrix} v_k \\ \omega_k \end{bmatrix}, \tag{3.6}$$

where v_k and ω_k denote the translational and angular velocities, respectively. The discrete evolution of the robot state can be generally expressed as

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k), \tag{3.7}$$

where $f(\cdot)$ represents the motion model used by the controller to predict the future robot state.

Rather than optimizing a single instantaneous control input, MPPI considers a nominal sequence of future control actions

$$\mathbf{U} = [\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_{T-1}], \tag{3.8}$$

where T represents the number of discrete time steps composing the prediction horizon.

Considering a constant model time step Δt, the overall temporal extent of the prediction horizon can be expressed as

$$T_h = T\Delta t \tag{3.9}$$

The optimization of the control sequence is performed through stochastic sampling. At each optimization iteration, MPPI generates a set of K alternative control sequences by introducing random perturbations into the current nominal solution. For the j-th sampled control sequence, the perturbed control input at prediction step k can be expressed as

$$\mathbf{u}_{j,k} = \mathbf{u}_k + \boldsymbol{\epsilon}_{j,k}, \tag{3.10}$$

where the perturbation term is sampled from a zero-mean Gaussian distribution

$$\boldsymbol{\epsilon}_{j,k} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}), \tag{3.11}$$

The covariance matrix $\boldsymbol{\Sigma}$ determines the dispersion of the stochastic perturbations and therefore regulates the extent of the exploration around the nominal control sequence. The resulting batch, obtained by iteratively repeating the sampling process, contains K alternative control sequences representing different possible control solutions over the same prediction horizon [21]. Each sampled control sequence is subsequently propagated through the robot motion model, generating a corresponding predicted state trajectory

$$\boldsymbol{\tau}_j = [\mathbf{x}_{j,0}, \mathbf{x}_{j,1}, \ldots, \mathbf{x}_{j,T}]\,. \tag{3.12}$$

The resulting batch of trajectories therefore represents different possible future evolutions of the robot over the considered prediction horizon. The number of sampled trajectories, their temporal extension, and the magnitude of the stochastic perturbations determine the extent to which the controller explores the space of possible control solutions.

Once the batch of candidate trajectories has been generated, each predicted motion is evaluated according to the navigation objectives defined for the controller. As previously described, this evaluation is performed through a number of independent critic functions constituting the modular architecture implemented in Navigation2. Each critic examines a specific property of the predicted robot motion, such as compliance with the robot motion constraints, obstacle avoidance, adherence to the global path, progress towards the goal, or orientation with respect to the desired path and goal pose. For each candidate trajectory $\boldsymbol{\tau}_j$, the contributions generated by the active critics are accumulated into an overall trajectory cost. In general terms, this evaluation can be represented as

$$S_j = \sum_{i=1}^{N_c} J_i(\boldsymbol{\tau}_j), \tag{3.13}$$

where S_j denotes the overall cost associated with the j-th candidate trajectory, N_c represents the number of active critics, and $J_i(\boldsymbol{\tau}_j)$ denotes the cost contribution generated by the i-th critic for that trajectory. Each critic can be independently configured and typically provides a weighting parameter, `cost_weight`, determining its relative influence on the trajectory evaluation, together with a `cost_power` parameter defining the power applied to the corresponding cost term. Therefore, $J_i(\boldsymbol{\tau}_j)$ represents the resulting configured contribution of the i-th critic to the overall trajectory cost.

Similarly to the DWB controller, this modular formulation allows the navigation behavior to be modified through critic configuration by introducing additional critics or by enabling, disabling, and reconfiguring existing ones, without changing the underlying MPPI optimization mechanism.

After evaluating the complete trajectory batch, the controller obtains the corresponding set of trajectory costs

$$\mathbf{S} = [S_1, S_2, \ldots, S_K]\,. \tag{3.14}$$

In particular, lower costs identify predicted motions that better satisfy the navigation objectives evaluated by the active critics, whereas trajectories presenting undesirable characteristics are assigned higher costs. However, unlike the trajectory-selection strategy used by the DWB controller, which selects the valid candidate trajectory with the best overall critic score, MPPI uses the cost distribution of the complete sampled batch to update the nominal control sequence. Candidate trajectories with lower costs have a greater influence on this update, while high-cost trajectories contribute progressively less to the resulting solution. This operation constitutes the importance-weighting stage of the MPPI optimization [21].

In the Navigation2 implementation, this importance-weighting stage is performed through a soft-max operation over the trajectory costs. The normalized importance weight associated with the j-th sampled trajectory can be expressed as

$$w_j = \frac{\exp\left(-\frac{1}{\lambda}\left(S_j - S_{\min}\right)\right)}{\sum_{\ell=1}^{K} \exp\left(-\frac{1}{\lambda}\left(S_\ell - S_{\min}\right)\right)}, \tag{3.15}$$

where

$$S_{\min} = \min_{j=1,\dots,K} S_j, \tag{3.16}$$

and $\lambda > 0$ represents the temperature parameter of the MPPI optimization. The subtraction of the minimum trajectory cost improves the numerical stability of the exponential computation without changing the relative importance of the candidate solutions. The temperature parameter therefore regulates the selectivity of the optimization with respect to the trajectory costs. Lower values of λ concentrate the importance distribution around the lowest-cost samples, making the optimization more selective towards the best-performing control sequences. Conversely, higher values of λ produce a more distributed weighting, allowing a larger portion of the sampled trajectory set to contribute to the control update.

The resulting importance weights are subsequently used to update the nominal control sequence according to the stochastic perturbations associated with the sampled trajectories. Conceptually, the updated nominal control input at prediction step k can be represented as

$$\mathbf{u}_k^{\text{new}} = \mathbf{u}_k + \sum_{j=1}^{K} w_j \boldsymbol{\epsilon}_{j,k}. \tag{3.17}$$

Consequently, perturbations associated with low-cost predicted motions contribute more strongly to the update, whereas perturbations associated with high-cost trajectories have a smaller influence on the resulting control sequence.

The optimized control sequence is obtained by applying this operation to each control input along the entire prediction horizon:

$$\mathbf{U}^* = \left[\mathbf{u}_0^*, \mathbf{u}_1^*, \dots, \mathbf{u}_{T-1}^*\right]. \tag{3.18}$$

The resulting sequence represents the optimized future control evolution computed at the current controller iteration. As previously introduced, only the first control input of the optimized sequence, $\mathbf{u}_0^*$, is applied as the current robot command. At the following control cycle, the computed control solution provides the starting point for the subsequent optimization.

The prediction and stochastic optimization process described above is governed by a set of configurable parameters that determine the characteristics of the

prediction horizon, stochastic sampling, and optimization procedure. The principal parameters of the Navigation2 MPPI controller are summarized in Table 3.4 [31].

Parameter	**Related quantity**	**Description**
`time_steps`	T	Number of discrete prediction steps composing the control sequence and trajectory rollout.
`model_dt`	Δt	Time interval between two consecutive prediction steps.
`batch_size`	K	Number of sampled candidate control sequences and corresponding trajectories generated during an optimization iteration.
`iteration_count`		Number of optimization iterations performed within each controller cycle.
`vx_std`	σ_{v_x}	Standard deviation of the Gaussian perturbations applied to longitudinal velocity.
`vy_std`	σ_{v_y}	Standard deviation of the Gaussian perturbations applied to lateral velocity for holonomic motion models.
`wz_std`	σ_{ω}	Standard deviation of the Gaussian perturbations applied to angular velocity.
`temperature`	λ	Controls the selectivity of the importance weighting with respect to trajectory costs.
`gamma`	γ	Controls the contribution associated with control effort within the MPPI optimization formulation.
`motion_model`	$f(\cdot)$	Motion model employed to propagate the robot state according to the sampled control sequences.

Table 3.4: Main prediction, sampling, and optimization parameters of the Navigation2 MPPI controller.

Among the parameters listed in Table 3.4, `time_steps` and `model_dt` correspond respectively to the quantities T and Δt introduced in the theoretical formulation. Therefore, they jointly determine the temporal extent of the prediction horizon, also influencing the computational cost and temporal resolution of the predicted trajectories. The stochastic exploration itself is mainly controlled through the standard deviations of the Gaussian perturbations. In the case of a differential-drive platform, `vx_std` and `wz_std` determine the dispersion of the perturbations applied to the longitudinal and angular velocity components, respectively. Higher standard deviations increase the exploration around the nominal control sequence, while

lower values concentrate the candidate solutions closer to the current nominal solution. The `vy_std` parameter performs the corresponding function for lateral velocity and is therefore particularly relevant to holonomic motion models. Unlike the `temperature` parameter which influences the importance-weighting mechanism, the `gamma` parameter is, instead, associated with the control-cost component of the MPPI formulation and regulates the contribution of control effort. In addition to these optimization parameters, velocity and acceleration limits constrain the admissible robot motion. Parameters such as `vx_max`, `vx_min`, `wz_max`, `ax_max`, `ax_min`, and `az_max` define the translational and rotational motion limits within which the controller operates. These constraints are combined with the selected motion model so that the predicted trajectories remain compatible with the kinematic and dynamic capabilities assumed for the robotic platform.

While the parameters described above determine how the candidate motions are generated, propagated, and explored, the resulting navigation behavior also strongly depends on how these motions are evaluated by the employed critic functions, whose configuration makes the behavior of the controller adaptive according to the specific requirements of the navigation task. The principal critics are summarized in Table 3.5.

Critic	**Main objective**
ConstraintCritic	Penalizes motions that violate the constraints associated with the selected robot motion model.
CostCritic	Evaluates trajectories according to local costmap costs and penalizes potentially colliding motions.
ObstaclesCritic	Penalizes proximity to obstacles and critical near-collision configurations.
PathAlignCritic	Encourages the predicted robot trajectory to remain aligned with the global path.
PathFollowCritic	Encourages progress along the global path.
PathAngleCritic	Penalizes undesirable angular disagreement between predicted motion and the global path.
GoalCritic	Encourages convergence of the robot position towards the goal.
GoalAngleCritic	Encourages convergence towards the desired orientation at the goal.
PreferForwardCritic	Penalizes unnecessary reverse motion and promotes forward navigation.

Table 3.5: Principal critic functions available in the Navigation2 MPPI controller [31].

The enabled critic plugins cooperate in shaping the trajectory cost according to navigation objectives. Specifically, path-related critics regulate the relationship between the predicted robot motion and the global path, encouraging progression along the reference path and evaluating the alignment with the path geometry (*PathFollowCritic*, *PathAlignCritic*). Additionally, the *PathAngleCritic* penalizes motions characterized by inappropriate orientations with respect to the desired path direction. As the robot approaches the destination, goal-related critics contribute to the final convergence behavior, evaluating both the positional relationship and the final orientation (*GoalCritic*, *GoalAngleCritic*).

Obstacle avoidance is instead represented through costmap- and obstacle-related terms. The *CostCritic* evaluates the costs encountered by the predicted trajectories in the local costmap, penalizing motions entering high-cost or potentially colliding regions. The *ObstaclesCritic* provides a dedicated formulation for obstacle interaction, distinguishing the general preference for maintaining clearance from obstacles from critical near-collision conditions. The exact obstacle-related critic adopted therefore depends on the controller configuration. The *ConstraintCritic* complements these objectives by evaluating consistency with the constraints imposed by the selected motion model. Additional critics can further shape specific aspects of the resulting motion, such as the *PreferForwardCritic* which discourages unnecessary backward motion.

It is important to remark that critics do not directly prescribe the trajectory that the robot must follow. Instead, they modify the cost associated with the set of sampled trajectories. Increasing the weight assigned to a specific critic increases the relative penalty associated with violations of the corresponding navigation objective. Consequently, candidate trajectories that poorly satisfy a highly weighted navigation property receive larger costs and exert a smaller influence on the importance-weighted update of the nominal control sequence. For this reason, the critic weights represent one of the principal parameters in the controller configuration. They must therefore be properly configured to better adapt the behavior to the application of interest. Depending on the relative configurations, indeed, different navigation objectives can be emphasized, while preserving the same underlying stochastic optimization procedure. Moreover, the modular architecture enables additional application-specific objectives to be incorporated directly into the MPPI trajectory evaluation. In socially aware navigation, for example, the evaluation can specifically account for robot-human interactions by integrating dedicated social critics designed for this purpose, such as the LightSFM-based Social Critic, which represents the social component considered in this work.

3.4 LightSFM-based Social Critic

To achieve social compliance during robot navigation, this additional objective is introduced into the trajectory optimization process by integrating an existing Social Critic based on a lightweight implementation of the Social Force Model (LightSFM) [24].

Generally, the Social Force Model describes pedestrian motion through the combination of different virtual forces representing the tendency of an agent to move towards its desired destination while simultaneously reacting to surrounding people and obstacles [22].

In general terms, the resulting force acting on an agent i can be expressed as

$$\mathbf{F}_i = \mathbf{F}_i^{goal} + \mathbf{F}_i^{people} + \mathbf{F}_i^{obstacle}, \tag{3.19}$$

where $\mathbf{F}_i^{goal}$ represents the attractive contribution driving the agent towards its desired goal, while $\mathbf{F}_i^{people}$ and $\mathbf{F}_i^{obstacle}$ represent the interaction contributions generated by surrounding agents and environmental obstacles, respectively. Hence, the local navigation behavior of pedestrians is described by the combination of attractive and repulsive forces.

In the considered Social Critic, pedestrians are not simply geometric obstacles, but their states are explicitly introduced into the social interaction model. The critic subscribes to a topic providing people messages and constructs a LightSFM agent for each received person using the available position, velocity, and orientation information. This representation enables the explicit evaluation of the robot-human interaction along the MPPI prediction horizon.

Since the Social Force Model requires a desired destination for the simulated agents, a temporary local goal is assigned to each pedestrian according to its current motion. Considering the position $\mathbf{p}_i$ and velocity $\mathbf{v}_i$ of pedestrian i, this goal can be represented as

$$\mathbf{g}_i = \mathbf{p}_i + \mathbf{v}_i T_h, \tag{3.20}$$

where T_h corresponds to the temporal extension of the MPPI prediction horizon. Therefore, rather than requiring knowledge of the actual long-term pedestrian destination, the critic assumes that the current pedestrian motion provides a short-term indication of its intended direction. This is consistent with the implementation,

where the temporary goal information is obtained by projecting the current velocity over the MPPI horizon. The interaction between the robot and the surrounding people is quantified through the concept of social work. In particular, the formulation accounts both for the social interaction experienced by the robot and for the one induced by the robot itself on the nearby pedestrians. The total social work can be expressed as

$$W_{social} = W_r + W_p, \tag{3.21}$$

where W_r indicates the contribution associated with the social force acting on the robot, while W_p represents the cumulative interaction produced by the robot on persons.

More specifically,

$$W_r = \left\| \mathbf{F}_r^{social} \right\|, \qquad W_p = \sum_{i=1}^{N_p} \left\| \mathbf{F}_{i \leftarrow r}^{social} \right\|, \tag{3.22}$$

where N_p denotes the number of considered pedestrians and $\mathbf{F}_{i \leftarrow r}^{social}$ is the social interaction force induced by the robot on the i-th pedestrian. This bidirectional formulation is explicitly implemented in the Social Critic: the social force acting on the robot is first evaluated, after which the interaction forces acting on the pedestrians are recomputed by considering the robot as an interacting agent.

The resulting social interaction measure is directly integrated into the MPPI trajectory evaluation described in the previous section. For each candidate trajectory $\boldsymbol{\tau}_j$, the Social Critic simulates the evolution of the robot and the considered pedestrians along the prediction horizon. At each evaluated prediction instant, LightSFM updates the pedestrian dynamics according to the interaction forces, while the robot state is given by the corresponding state of the candidate MPPI trajectory. The social work associated with the specific robot-human configuration is then computed and accumulated along the rollout. However, performing a complete interaction computation at every time step of the prediction horizon for every candidate trajectory would considerably increase the computational effort of the controller, particularly considering that the controller may evaluate thousands of trajectories during each optimization cycle. For this reason, the Social Critic adopts a temporal evaluation strategy with a given granularity that is controlled by the `step_grouping` parameter. This basically means that rather than evaluating the social interaction at every point of the original MPPI rollout, consecutive prediction

steps are grouped and LightSFM is evaluated at a lower temporal resolution. Moreover, it is worth noticing that before performing the social evaluation, the critic additionally restricts the set of pedestrians considered in the computation. Agents are selected according to configurable spatial bounds around the robot, limiting the model only to those pedestrians that are sufficiently close and whose influence is relevant to the current robot motion. In this way the computational effort is further reduced, avoiding unnecessary computations involving distant agents. Environmental obstacles are also incorporated into the LightSFM simulation. The critic collects obstacle information from the local costmap, performs a spatial reduction through max pooling, and extracts cells corresponding to dangerous obstacles. However, obstacle points located within a configurable `person_radius` of a selected pedestrian are removed before the LightSFM evaluation. In this way, the pedestrian representation used by the interaction model is separated from conventional obstacle points extracted from the navigation costmap. The main parameters controlling the Social Critic evaluation are summarized in Table 3.6.

Parameter	**Description**
`social_weight`	Scales the contribution of the social interaction cost to the overall MPPI trajectory evaluation.
`cost_power`	Exponent applied to the Social Critic contribution before its addition to the trajectory cost.
`step_grouping`	Number of MPPI prediction steps grouped between successive LightSFM evaluations.
`field_of_view`	Angular region around the robot within which pedestrians are considered.
`max_distance_robo_agent_x`	Maximum admissible distance along the x direction for including a pedestrian in the social computation.
`max_distance_robo_agent_y`	Maximum admissible distance along the y direction for including a pedestrian in the social computation.
`person_radius`	Radius used to distinguish pedestrian-associated costmap points from conventional environmental obstacles.
`pooling_size`	Spatial downsampling factor used when extracting obstacle information from the local costmap.
`laser_distance_cut_off`	Distance threshold associated with the obstacle information considered by the critic.

Table 3.6: Main configuration parameters of the LightSFM-based Social Critic.

Among these parameters, `social_weight` establishes the relative importance assigned to social interactions with respect to the other conventional navigation objectives. A higher weight value increases the penalty assigned to trajectories generating stronger predicted interactions with pedestrians, whereas a lower value reduces the influence of the social objective relative to path tracking, obstacle avoidance, goal convergence, and the other enabled cost terms. The resulting socially aware behavior is therefore not simply determined by the Social Critic operating as a standalone function, but derives from the balance between the Social Critic and the standard MPPI critics.

3.5 Visual Perception and Deep Learning

This section starts the second part of the chapter, providing an overview of the fundamental tasks in computer vision, such as object detection, instance segmentation, pose estimation, and re-identification, that constitute the basis for the methods employed in the perception system considered in this thesis.

Visual perception enables the extraction of meaningful information about the surrounding environment from images and video sequences. Depending on the information that has to be recovered from the visual input, different computer vision tasks can be distinguished. Image classification aims at determining what is represented in an image by assigning it a semantic class, whereas object localization, as the name suggests, determines where an object is located. Typically, localization is performed through the identification of a bounding box, represented through rectangular coordinates (x_c, y_c, w, h), which define the spatial location of the detected object in the image plane. The coordinates (x_c, y_c) usually correspond to the pixel location of the bounding box center, while w and h represent its width and height. Object detection combines these two objectives, simultaneously identifying and localizing one or more object instances within the observed scene. The output of an object detector therefore generally consists of a set of bounding boxes, each associated with a predicted class and a confidence score expressing the reliability of the corresponding detection.

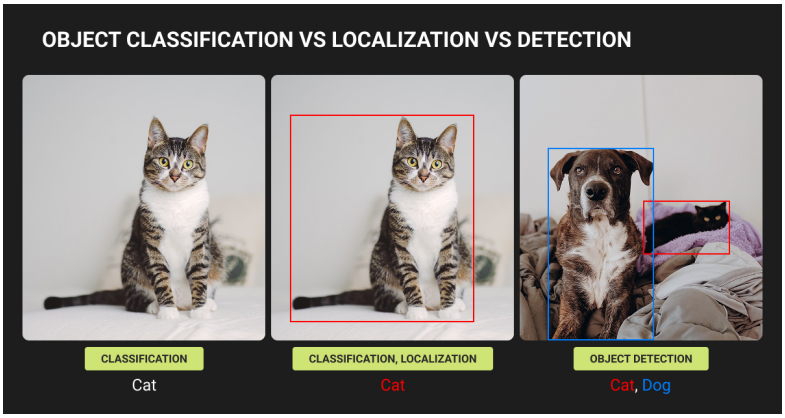


Figure 3.3: Comparison between image classification, object localization, and object detection [32].

Instance segmentation extends object detection by providing a more detailed spatial representation of each detected instance. The detected object is not described only through a rectangular bounding box; instead, instance segmentation assigns a pixel-wise mask identifying the image pixels belonging to that specific object. Consequently, it provides a more precise representation of the actual visible shape and spatial extent of the detected object, as shown in Figure 3.4.

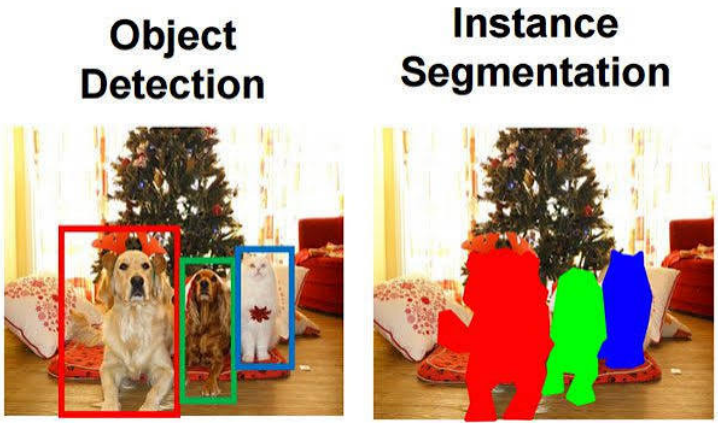


Figure 3.4: Comparison between object detection and instance segmentation [33].

More specialized computer vision tasks can provide additional information about the detected targets, such as pose estimation. Rather than describing only the spatial extent or the visible shape of the object of interest, pose estimation aims to recover information about its internal configuration by extracting a set of characteristic landmarks. The spatial arrangement of these landmarks can then be exploited to reason about the pose and orientation of the observed target.

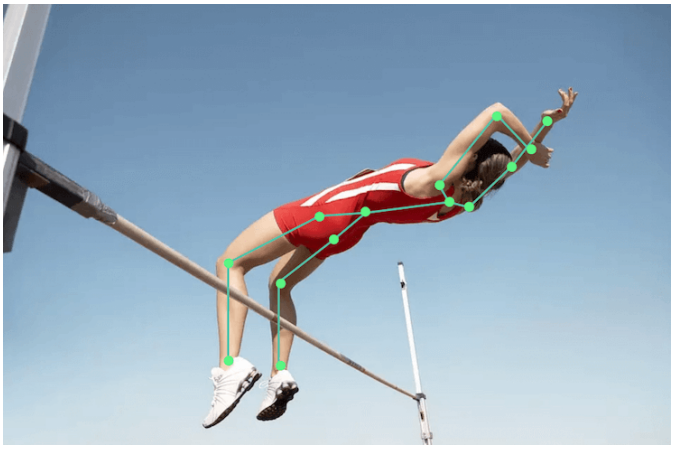


Figure 3.5: Example of anatomical landmark extraction for human pose estimation [34].

Re-identification (Re-ID), on the other hand, addresses a different visual perception problem, whose objective is to determine whether observations acquired at different time instants, and potentially from different cameras, correspond to the same physical target. Hence, it focuses on visual appearance information, extracting features that can be exploited to recognize the same target across different observations over time. This requires the extraction of visual characteristics that are sufficiently discriminative, while remaining robust to changes in pose, viewpoint, illumination, and partial appearance variation. Modern Re-ID methods extensively rely on neural networks to extract such discriminative visual representations. Specifically, given an image region containing a target, a Re-ID network transforms the visual information into a compact numerical representation, commonly referred to as a feature vector or appearance embedding. The network is trained to construct an embedding space in which observations belonging to the same identity are represented by similar feature vectors, while observations corresponding to different identities are mapped farther apart. These visual perception tasks have been increasingly addressed through deep learning approaches. In particular, Convolutional Neural Networks (CNNs) are able to automatically learn relevant visual representations directly from image data, rather than relying on manually designed features. The feature extraction process is hierarchical: the first network layers generally capture simpler visual patterns, such as edges, textures, and corners, while deeper layers combine these simple patterns into more complex representations related to shapes, object parts, and higher-level visual characteristics.

The learned feature representations can then be exploited by task-specific components of the network to generate the desired output, such as object classes and bounding boxes for object detection, pixel-wise masks for instance segmentation, characteristic landmarks for pose estimation, or discriminative appearance embeddings for re-identification. Therefore, these tasks provide complementary information about the observed targets and constitute fundamental components of modern people perception and tracking systems.

3.6 People Detection and Pose Estimation

Visual perception has gained the stage in recent years, showing itself as a fundamental aspect for autonomous mobile robotics, especially when operating in environments shared with humans. In the context of social navigation, detecting the presence of people is not sufficient on its own, since information concerning their spatial position and body configuration can provide additional knowledge about the surrounding human environment. For this reason, human pose estimation extends conventional person detection by identifying a set of anatomical landmarks, typically referred to as keypoints, associated with the human body.

In this work, the visual perception pipeline is based on YOLOv8 Pose, a pose-estimation variant of the YOLO (You Only Look Once) family developed by Ultralytics [35]. As anticipated in the previous section, YOLO models belong to the class of one-stage deep learning architectures, in which object localization and classification are performed within a single inference process. The original YOLO formulation introduced this approach by treating object detection as a unified regression problem, obtaining detections directly from the input image while maintaining high computational efficiency and speed [36]. In particular, in the original YOLO formulation, the input image is divided into an $S \times S$ grid, and each grid cell is responsible for predicting bounding boxes and class probabilities for objects whose center falls within that cell. This architecture presented limitations, particularly in detecting small objects or objects located close to one another, partly due to the spatial constraints imposed by the grid-based prediction mechanism. The YOLO family has subsequently evolved through successive generations. In particular, YOLOv8 provides dedicated model variants for different computer vision tasks, including object detection, instance segmentation, and human pose estimation, with YOLOv8 Pose representing the variant dedicated to the latter task [35]. In particular, it additionally identifies and maps characteristic anatomical points (keypoints) belonging to each detected person in images or video frames.

Therefore, the output associated with a person detection can be represented as the combination of a bounding box

$$\mathbf{b}_i = [x_{1,i}, y_{1,i}, x_{2,i}, y_{2,i}, s_i] \,, \tag{3.23}$$

and a set of body keypoints

$$\mathcal{K}_i = \{\mathbf{k}_{i,1}, \mathbf{k}_{i,2}, \ldots, \mathbf{k}_{i,17}\} \,, \tag{3.24}$$

where $x_{1,i}$, $y_{1,i}$, $x_{2,i}$, and $y_{2,i}$ define the coordinates of the bounding box in the image plane, s_i represents the confidence associated with the person detection, and $\mathbf{k}_{i,j}$ denotes the j-th body keypoint associated with the i-th detected person. The YOLOv8 Pose models trained on the COCO human-pose dataset represent each detected person through 17 anatomical keypoints, corresponding to the nose, left and right eyes, left and right ears, left and right shoulders, elbows, wrists, hips, knees, and ankles [35]. Each keypoint is characterized by its two-dimensional location in the image plane and by the confidence associated with its prediction. The j-th keypoint of a detected person can therefore be represented as

$$\mathbf{k}_{i,j} = [u_{i,j}, v_{i,j}, c_{i,j}]^T \,, \tag{3.25}$$

where $u_{i,j}$ and $v_{i,j}$ are the horizontal and vertical pixel coordinates of the keypoint and $c_{i,j}$ is its prediction confidence. The resulting outcome therefore provides the human representation with considerably richer information than the bounding box alone. This is because instead of only identifying the region of image occupied by the person, the keypoints also allow the spatial body configuration description and make it possible to reason about individual anatomical regions independently.



Figure 3.6: Example of human pose estimation using YOLOv8-Pose [37].

This property is particularly relevant for RGB-D-based people perception. Since every predicted keypoint is associated with a specific pixel location in the RGB image, when the depth information is spatially aligned with the color image, the corresponding depth measurement can be retrieved from the depth map. Once the RGB-D information is available, the 2D body landmarks can subsequently be deprojected into 3D coordinates by exploiting the intrinsic parameters of the camera. Multiple valid body keypoints can thus contribute to the estimation of the person's position, while the relative arrangement of specific anatomical points can additionally provide information about body orientation. For these reasons, YOLOv8-Pose constitutes the first stage of the perception architecture considered in this work: from each RGB frame, it simultaneously supplies the person detections required by the tracking system and the body features required for the subsequent estimation of the three-dimensional position and orientation of the detected person.

3.7 Multi-Object Tracking

The detection stage makes it possible to collect information about people observed in each individual frame, but for autonomous robotic perception an additional step is required: maintaining this information consistently over time. Since the application field is highly dynamic, it may happen that the same person is observed across several consecutive frames, temporarily disappears, or moves in close proximity to other individuals. Consequently, relying only on people detection and independently processing each frame does not provide comprehensive knowledge of the observed scene, since no temporal relationship is established between successive observations. Multi-Object Tracking (MOT) addresses this problem by associating detections belonging to the same physical target over time, generating trajectories and maintaining a persistent identity for each tracked object.

A popular strategy for Multi-Object Tracking is the tracking-by-detection paradigm [8]. In this approach, object detection and temporal tracking are treated as two consecutive stages. Visible targets in the current frame are first identified by an object detector, and then the tracking algorithm associates the newly obtained detections with the tracks maintained from previous frames. The MOT problem can therefore be essentially formulated as a data association problem, in which motion information, appearance information, or a combination of both is used to determine the correspondence between existing tracks and incoming observations. This paradigm represents the basis of several widely adopted trackers, including SORT, DeepSORT, StrongSORT, and BoT-SORT.

The association process can be represented by considering a set of M tracks

maintained from the previous observations,

$$\mathcal{T} = \{T_1, T_2, \ldots, T_M\}\,, \tag{3.26}$$

and the set of N detections acquired from the current frame,

$$\mathcal{D} = \{D_1, D_2, \ldots, D_N\}\,. \tag{3.27}$$

For each possible track-detection pair (T_i, D_j), the tracker computes an association cost $C_{i,j}$, representing the dissimilarity between the predicted state of the track and the incoming observation. The resulting cost matrix can therefore be expressed as

$$\mathbf{C} = [C_{i,j}]_{M \times N}\,. \tag{3.28}$$

Data association consists in finding the correspondence between tracks and detections that minimizes the overall association cost, while respecting the one-to-one assignment constraint. The Hungarian algorithm is commonly employed in this family of trackers to efficiently solve this linear assignment problem.

SORT

Among the approaches belonging to the tracking-by-detection paradigm, SORT (Simple Online and Realtime Tracking) combines a Kalman filter for motion estimation with frame-by-frame data association [38]. For each active track, the Kalman filter is employed to predict the expected state of the corresponding target in the new frame, while newly available detections provide the observations used to correct the predicted state. Association between predicted tracks and incoming detections is mainly based on their spatial compatibility, expressed in the SORT formulation through the Intersection over Union (IoU) between the corresponding bounding boxes. The resulting assignment is then solved using the Hungarian algorithm.

Given two bounding boxes A and B, their spatial overlap can be expressed as

$$\mathrm{IoU}(A, B) = \frac{|A \cap B|}{|A \cup B|}, \tag{3.29}$$

where $|A \cap B|$ represents the area of intersection between the two bounding boxes, while $|A \cup B|$ represents their union area.

The IoU assumes values between 0 and 1. A larger IoU value indicates a stronger spatial correspondence between the predicted location of the target and the incoming detection, with a value of 1 corresponding to complete overlap, while non-overlapping bounding boxes result in a value equal to zero. Hence, the smaller the IoU, the lower the spatial compatibility between the predicted track and the incoming detection. This formulation gives a simple and computationally efficient tracking architecture, but there are some drawbacks. The exclusive or prevalent reliance on geometric and motion information, indeed, creates problems when trying to distinguish different targets when they move close to one another or undergo temporary occlusions, since such ability is limited with the risk of tracking fragmentation.

DeepSORT

The limitations associated with purely motion-based association are addressed by DeepSORT (Deep Simple Online and Realtime Tracking), which extends the SORT architecture by introducing visual appearance information into the data-association process [39]. The underlying principle is that the motion information about where a previously tracked target is expected to appear in the current frame is complemented with appearance information, indicating whether the currently observed target is visually similar to that track. The DeepSORT architecture is consequently characterized by the combination of two complementary branches: a motion branch and an appearance branch.

The motion branch continues to rely on a Kalman filter to recursively estimate the state of each active track. Given the previously estimated state $\hat{\mathbf{x}}_{k-1}$, the prediction step can generally be represented as

$$\hat{\mathbf{x}}_k^- = \mathbf{F}_k \hat{\mathbf{x}}_{k-1}, \tag{3.30}$$

while the corresponding uncertainty is propagated according to

$$\mathbf{P}_k^- = \mathbf{F}_k \mathbf{P}_{k-1} \mathbf{F}_k^T + \mathbf{Q}_k, \tag{3.31}$$

where $\mathbf{F}_k$ represents the state-transition model, $\mathbf{P}_k$ the state covariance, and $\mathbf{Q}_k$ the process-noise covariance.

Once a corresponding detection becomes available, the predicted state can be

corrected through the Kalman update, combining the prediction and the newly acquired observation. The uncertainty associated with the predicted track is additionally exploited during data association through the squared Mahalanobis distance. In particular, considering a detection $\mathbf{z}_j$, the distance from the predicted measurement $\hat{\mathbf{y}}_i$ associated with track i can be expressed as

$$d^{(m)}(i,j) = (\mathbf{z}_j - \hat{\mathbf{y}}_i)^T \mathbf{S}_i^{-1} (\mathbf{z}_j - \hat{\mathbf{y}}_i), \tag{3.32}$$

where $\mathbf{S}_i$ represents the innovation covariance associated with the predicted track. Unlike a simple Euclidean distance, the squared Mahalanobis distance $d^{(m)}(i,j)$ takes into account the uncertainty associated with the state estimate. In DeepSORT, this information is employed as a gating mechanism to reject associations that are inconsistent with the predicted target motion before performing the final assignment [39].

The main extension introduced by DeepSORT, however, concerns the appearance branch. For each detected person, a deep neural network trained for person re-identification (Re-ID) extracts an appearance descriptor, commonly referred to as an appearance embedding:

$$\mathbf{f}_j \in \mathbb{R}^d, \tag{3.33}$$

where d denotes the dimensionality of the feature space. The purpose is to map visually similar observations of the same person to nearby points in the embedding space, while maintaining greater separation between visually different individuals. In this context, DeepSORT maintains a gallery of appearance descriptors associated with each active track. Assuming normalized feature vectors, the appearance compatibility between track i and detection j can be evaluated through the minimum cosine distance between the descriptor extracted from the incoming detection and the descriptors stored in the track gallery:

$$d^{(a)}(i,j) = \min_{\mathbf{f}_k^{(i)} \in B_i} \left(1 - \mathbf{f}_j^T \mathbf{f}_k^{(i)}\right), \tag{3.34}$$

where B_i denotes the feature gallery associated with the i-th track. A lower appearance distance therefore indicates greater visual similarity between the current detection and the previous observations associated with that same identity. The original DeepSORT formulation stores multiple descriptors for each track, allowing the current observation to be compared with its recent appearance history [39].

Hence, the additional Re-ID representation helps discriminate between different targets that remain spatially compatible, making the association process more reliable. DeepSORT further organizes the association process through a matching cascade, which gives priority to tracks according to their recency, while the resulting assignment problems are still solved through the Hungarian algorithm [39].

The introduction of appearance information in DeepSORT notably improves identity preservation compared with purely geometry-based tracking, especially in crowded environments with targets crossing trajectories. Nevertheless, several components of the original DeepSORT architecture were progressively replaced by improvements in object detection, Re-ID, and motion estimation. StrongSORT was therefore introduced to strengthen the DeepSORT formulation while preserving its fundamental tracking-by-detection philosophy.

StrongSORT

StrongSORT makes DeepSORT great again as stated in [40], it represents therefore an evolution of DeepSORT rather than a completely different tracking approach. As anticipated, the method retains the fundamental combination of motion and appearance information, but improves the processes involved in feature extraction, state estimation and data association. Its development was strongly motivated by the consideration that the DeepSORT framework could remain highly competitive when equipped with more recent and advanced techniques. Specifically, one of the principal modifications concerns the appearance branch, where StrongSORT replaces the relatively simple appearance model employed by the original DeepSORT implementation with a stronger person Re-ID network, providing more discriminative feature representations. At the same time, the feature-bank mechanism analyzed in DeepSORT is replaced by an Exponential Moving Average (EMA) representation, enabling the progressive update of a single appearance state as new observations are associated with the track.

In general terms, for the i-th track, the appearance representation at time t is updated according to the following expression:

$$\mathbf{e}_i^t = \alpha \mathbf{e}_i^{t-1} + (1-\alpha)\mathbf{f}_i^t, \tag{3.35}$$

where $\mathbf{e}_i^{t-1}$ represents the previously maintained appearance state, $\mathbf{f}_i^t$ is the embedding extracted from the newly associated detection, and α regulates the contribution of previous and current appearance information. In this way, the target representation evolves progressively over time without requiring the complete gallery of previous descriptors used by DeepSORT.

The StrongSORT formulation also introduces improvements in the motion branch. In particular, Camera Motion Compensation (CMC) is introduced through Enhanced Correlation Coefficient (ECC) image alignment to estimate the global transformation occurring between consecutive frames. This compensation was introduced to reduce the apparent displacement of tracked objects generated by camera motion rather than by the actual motion of the targets. Such a distinction becomes particularly relevant when tracking is performed from a moving sensing platform.

Moreover, the conventional Kalman-filter configuration is further improved through the Noise Scale Adaptive (NSA) Kalman filter. The introduced change adapts the measurement-noise covariance according to the confidence of the corresponding detection, rather than assuming the same measurement uncertainty for every observation.

This adaptation can be represented as

$$\widetilde{\mathbf{R}}_k = (1 - c_k)\mathbf{R}_k, \tag{3.36}$$

where c_k is the detection confidence, $\mathbf{R}_k$ is the original measurement-noise covariance, and $\widetilde{\mathbf{R}}_k$ represents its adapted value. Consequently, a high-confidence detection is associated with lower measurement uncertainty and can have a stronger effect on the state correction. Observations considered less reliable because of lower confidence values, instead, are assigned greater uncertainty and thus their contribution is weaker.

Another fundamental difference concerns the way motion and appearance information are employed during data association. Unlike DeepSORT, which mainly exploits appearance distance as the association cost while using motion information for gating, StrongSORT explicitly combines the two contributions within the matching cost [40]. The resulting cost matrix can be expressed as

$$\mathbf{C} = \lambda \mathbf{A}_a + (1 - \lambda)\mathbf{A}_m, \tag{3.37}$$

where $\mathbf{A}_a$ denotes the appearance-cost matrix, $\mathbf{A}_m$ represents the motion-cost matrix, and the weighting factor λ regulates their relative contribution. The final association therefore considers both the visual similarity of the targets and their consistency with the predicted motion.

Finally, StrongSORT employs a global linear assignment, replacing the matching cascade in the original DeepSORT formulation.

Hence, by combining all the introduced modifications and improvements, StrongSORT provides a more robust association framework while maintaining the general architecture inherited from DeepSORT.

BoT-SORT

A further development of the tracking-by-detection paradigm is represented by BoT-SORT (Bag of Tricks for SORT), proposed by Aharon, Orfaig and Bobrovsky as a robust multi-pedestrian tracking method combining motion and appearance information within a unified association framework [41]. Similarly to DeepSORT and StrongSORT, BoT-SORT manages a set of temporally evolving tracks and associates the newly available detections with the existing target hypotheses.

The method focuses on improving the accuracy of motion prediction and on integrating geometric and appearance information during data association. The original authors identify the combination of motion and appearance information, camera-motion compensation and an improved Kalman-filter state vector as the principal components of the method [41].

BoT-SORT motion model relies on the Kalman-filter formulation, which commonly represents a building block for SORT-based trackers. In conventional Kalman-filter-based tracking approaches such as SORT and DeepSORT, the bounding-box state is typically represented using its center coordinates, aspect ratio, height, and the corresponding temporal derivatives. BoT-SORT modifies this representation by directly modeling the width and height of the target bounding box rather than using an aspect-ratio formulation. The state vector can therefore be represented as

$$\mathbf{x} = \left[x_c, y_c, w, h, \dot{x}_c, \dot{y}_c, \dot{w}, \dot{h}\right]^T, \tag{3.38}$$

where x_c and y_c denote the coordinates of the bounding-box center, w and h represent its width and height, while the corresponding dotted terms describe their temporal rates of change. This representation allows changes in the dimensions of the tracked pedestrian to be explicitly included in the predicted state.

As in the previously described SORT-based approaches, the Kalman filter propagates each track's state and the comparison with the new detections is based on bounding-box geometric overlap. Specifically, in BoT-SORT the geometric compatibility is represented through an IoU-based distance as

$$d_{\mathrm{IoU}}(i, j) = 1 - \mathrm{IoU}(T_i, D_j), \tag{3.39}$$

where T_i represents the predicted bounding box of track i and D_j the bounding box associated with detection j. According to the formula, spatial consistency between overlapping boxes results in lower values of the association distance. The higher the distance, the greater the inconsistency.

However, as already observed when introducing the SORT architecture, geometric compatibility alone may remain ambiguous. For this reason, additional appearance information is also used in the complete BoT-SORT formulation, and visual compatibility is addressed through the cosine distance. In this case, the corresponding appearance distance can be expressed as

$$d_{\text{emb}}(i,j) = 1 - \mathbf{e}_i^T \mathbf{f}_j, \tag{3.40}$$

where $\mathbf{e}_i^T$ denotes the transpose of the appearance embedding associated with track i, while $\mathbf{f}_j$ represents the embedding extracted from detection j. Since both vectors are normalized, the scalar product $\mathbf{e}_i^T \mathbf{f}_j$ corresponds to their cosine similarity. Hence, like DeepSORT and StrongSORT, the approach is characterized by two association cues providing complementary information: the IoU distance evaluates the geometric consistency between the new detection and the predicted track position, while the embedding distance evaluates the compatibility of visual appearance with the identity maintained by the tracks.

It is worth highlighting that, in BoT-SORT, appearance and geometric information are not considered independently. Indeed, before appearance similarity is allowed to influence the assignment, candidate associations are constrained according to a geometric proximity threshold. This means that track-detection pairs having insufficient spatial correspondence are also not reliable for Re-ID association. This prevents visual similarity alone from producing a match between inconsistent observations. Similarly, appearance associations whose embedding distance exceeds a predefined threshold are considered unreliable. This gating mechanism, therefore, restricts the association space by requiring candidate matches to satisfy suitable spatial and appearance consistency conditions before they can contribute to the final assignment.

In the original BoT-SORT implementation, the gated Re-ID distance and IoU distance are compared and the lower admissible distance is kept for assignment.

Denoting the gated appearance distance as $\widetilde{d}_{\text{emb}}(i,j)$, the resulting association cost can therefore be represented as

$$d(i,j) = \min\left(d_{\text{IoU}}(i,j), \widetilde{d}_{\text{emb}}(i,j)\right), \tag{3.41}$$

Consequently, the Re-ID contribution can improve an association when appearance provides stronger evidence than geometric overlap, but it cannot override the spatial-proximity constraint. This represents an important difference from the weighted linear combination previously described for StrongSORT [41]. The resulting cost matrix is finally processed through linear assignment to determine the correspondences between predicted tracks and current detections.

BoT-SORT additionally inherits the two-stage association strategy introduced by ByteTrack [42, 41]. In the first stage, predicted tracks are matched with high-confidence detections using the previously described association mechanism. Tracks that remain unmatched are then considered in a second association stage, where lower-confidence detections are recovered. During this second phase, the IoU information is used, considering the geometric consistency with the predicted tracks. This strategy allows potentially useful detections to be retained rather than immediately discarded because of their lower confidence, improving track continuity when pedestrians are partially occluded or temporarily difficult to detect. After both association stages, the remaining unmatched observations are then handled according to the track-management strategy.

Overall, the evolution of algorithms, starting from the original SORT architecture, shows how modern tracking-by-detection approaches progressively integrate different sources of information to improve temporal association. While motion prediction and geometric consistency provide the basic mechanism for relating observations across consecutive frames, appearance information introduces an additional identity-related cue that becomes particularly relevant in more adverse situations. The improvement of tracking performance by combining multiple pieces of information is the basic strategy considered in this work.

Chapter 4

System Design and Implementation

The theoretical background provided in the previous chapter represents the starting point for the design and implementation of the system developed in this thesis. In particular, the proposed framework aims at addressing the problem of people perception and tracking, which represents an essential component for the final objective of generating robot navigation behaviors that explicitly account for human presence.

From the perception perspective, the system is designed to overcome the limitations of a single sensing viewpoint by employing multiple RGB-D cameras mounted on the mobile robotic platform. In particular, each camera provides an independent observation of the environment and is associated with a local perception pipeline responsible for human detection, three-dimensional localization, and tracking. This means that the developed system is tasked with the management of multiple independent sensing pipelines. Operating with a multi-camera payload, therefore, introduces an additional challenge: observations of the same person may originate from different cameras, either simultaneously in overlapping fields of view or sequentially as the person moves across the overall observable area. Consequently, local tracking information alone does not provide a unique and temporally consistent representation of the surrounding pedestrians.

To address this problem, a global multi-camera tracking and fusion strategy was developed. The objective was to combine the local observations derived from each camera expressed in a common reference frame, to obtain a unified representation of the people for social navigation.

The design of the perception architecture itself represents an additional aspect addressed in this work. The number and placement of the cameras directly influence the observable area around the robot, the extent of blind areas and of the overlaps, as well as the computational resources required to execute the perception pipelines. Hence, the final sensor configuration was defined through a preliminary dedicated design analysis aimed at identifying a suitable arrangement for the considered application.

The resulting global representation of surrounding pedestrians represents the needed connection to the socially aware navigation framework. Once the humans are correctly perceived, on the navigation side, the work focused on the configuration of the MPPI local controller and on the tuning of the critic functions, especially with the integration of the social critic, used to make the robot move safely in crowded indoor spaces.

This chapter, therefore, aims to describe the main design choices and implementation decisions underlying the developed system. The multi-camera perception architecture and the study leading to the final camera arrangement are presented first. The local perception and tracking pipeline is characterized, followed by the detailed description of the proposed fusion strategy for global multi-camera tracking and its implementation within ROS 2. Finally, the integration of the resulting pedestrian information into the socially aware navigation framework is discussed.

4.1 Multi-Camera Payload Design

The design of the multi-camera architecture was a necessary preliminary step toward the development of the global tracking system. In fact, the first requisite for the capability of observing and continuously tracking people around a mobile robot concerns the optimal spatial arrangement of the sensors from which the data is acquired. For this reason, a study was carried out, considering both the geometric characteristics of the selected RGB-D sensors and the physical constraints imposed by their installation on the robot. Different camera placements were evaluated according to the perception requirements defined below, progressively leading to the final configuration adopted in the developed system.

4.1.1 Sensor Configuration Requirements

Three main geometric requirements drove the definition of the camera configuration: maximizing the overall horizontal coverage, minimizing blind regions between

adjacent fields of view, and limiting excessive overlap. The achievement of these objectives requires a suitable trade-off to be identified. This is because increasing the angular separation between adjacent cameras extends the overall observable region, but it may also introduce larger blind areas between their fields of view. On the other hand, reducing this separation improves the continuity of the coverage, but the overall horizontal field of view becomes smaller since a greater overlap is produced. Having shared regions allows the same person to be observed by more than one camera and, if this is correctly managed, facilitates the continuity of perception when people move from one sensing region to another. For this reason, a certain degree of overlap is desirable in a multi-camera application. Nevertheless, it must be limited to avoid losing information because of the reduction of the effective coverage if the cameras share too much of the same perception area.

The geometric analysis was based on the characteristics of the Intel RealSense D435 cameras selected for the perception payload. At the considered resolution of 640×480 pixels, each RGB camera provides a horizontal field of view (HFOV) of $69°$. Defining α as the corresponding half-angle, it gives

$$\alpha = \frac{\text{HFOV}}{2} = 34.5°. \tag{4.1}$$

The operating range considered in the analysis extends up to 300 cm. These parameters define the individual observable region associated with each sensor and constitute the geometric basis for evaluating the full coverage, blind regions, and overlap resulting from the different camera arrangements. In addition to the geometric requirements, the sensor configuration must satisfy the physical constraints imposed by the available mounting area on the robotic platform. Camera dimensions, available mounting space, mechanical clearances, and possible interference between sensors must therefore be considered when defining their positions and orientations. The number of cameras also affects the hardware and computational complexity of the perception architecture, since each additional sensor requires the execution of an independent perception and tracking pipeline, while also increasing the hardware cost. Accordingly, the configurations investigated in the following sections are evaluated by considering the trade-off among all these aspects.

4.1.2 Two-Camera Configuration and Coverage Limitations

As a first design solution, a two-camera RGB-D configuration was tried since it represented a preferable solution in terms of both hardware cost and computational effort. Indeed, this first attempt aimed to extend the sensing capability beyond

a single viewpoint while keeping the amount of data to be processed manageable. The objective was therefore to determine whether an appropriate placement and orientation of two cameras could satisfy the geometric requirements previously defined and provide a satisfactory horizontal coverage of at least 150°, while maintaining a reduced sensing architecture.

The physical placement of the cameras was first determined according to the dimensions of the available mounting surface on the Clearpath Jackal platform. The external mounting area measures 50.8×43 cm, while the internal usable area is 42.8×32.3 cm. Considering the front side of the robot for camera installation, the available half-width is 16.15 cm with respect to the longitudinal symmetry axis of the platform. A reference frame located at the geometric center of the robot was considered according to the ROS convention, with the x-axis directed forward, the y-axis pointing to the left, and the z-axis pointing upward.

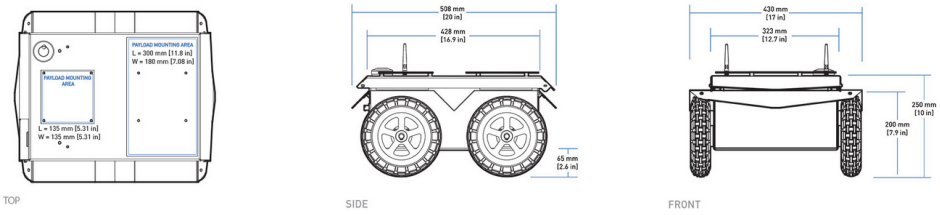


Figure 4.1: Clearpath Jackal platform dimensions [43].

To obtain an equivalent perception capability on the two sides of the robot, a symmetric arrangement was adopted. The two cameras were therefore positioned on opposite sides of the longitudinal axis and rotated outward by the same yaw angle. Their lateral displacement was determined by considering the camera dimensions and a mechanical tolerance of 2 cm from the boundaries of the mounting area.

Once the physically feasible positions were defined, the subsequent analysis focused on the fields of view to identify the optimal camera orientation. This was done by investigating the effect of the camera orientations on the resulting observable region. In particular, for each camera, the horizontal field of view can be represented in the xy-plane through two boundary rays originating from the camera optical center. Considering a generic operating distance x, each boundary ray is characterized by an angle determined by the camera yaw and by the HFOV half-angle α. The boundary of the field of view can therefore be expressed as

$$y(x) = y_c + (x - x_c)\tan(\theta), \qquad \theta = \psi \pm \alpha, \tag{4.2}$$

where (x_c, y_c) represents the camera position, ψ is its yaw orientation, and $\alpha = 34.5°$ is the HFOV half-angle.

The two inner rays are the relevant boundaries to analyze the region between the two cameras, since their relative position determines whether a central blind region or an overlap exists at a given distance from the robot. The difference between the lateral coordinates of the two inner rays can therefore be used to characterize the relation between their fields of view. A positive separation indicates the presence of a blind region, whereas after the two rays intersect, the fields of view overlap. The intersection distance consequently represents the transition between these two conditions.

Different yaw rotations were evaluated, considering the corresponding variations in the three main design objectives. A forward-facing configuration (0°) was first evaluated, at which the cameras' fields of view strongly overlap throughout the operating region, resulting in limited overall coverage. Progressively increasing the yaw angle rotates the cameras toward the lateral regions, extending the total horizontal coverage and reducing the overlap at larger distances. However, this improvement comes at the cost of an increasing central blind region close to the robot.

The main results obtained for the various investigated configurations are summarized in Table 4.1.

Camera yaw	**Blind region**	**Overlap**	**Total coverage**	**Observation**
20°	8.95 cm	137 cm	109°	Large overlap
25°	12.2 cm	82 cm	119°	Improved coverage
30°	**15.2** cm	**29.3** cm	**129°**	**Best trade-off**
35°	Persistent blind region	No overlap	139°	FoVs do not intersect

Table 4.1: Comparison of the investigated two-camera configurations for different yaw orientations.

As shown in Table 4.1, the configuration with a yaw angle of 30° corresponds to the most suitable compromise among the considered requirements, providing an overall horizontal coverage of 129°, while maintaining both the blind region and the overlap within acceptable limits. Conversely, for yaw angles greater than the HFOV half-angle $\alpha = 34.5°$, the inner field-of-view boundaries no longer intersect, resulting in a persistent central blind region.

The conducted study and the corresponding results show that even the optimal arrangement identified for the two-camera architecture cannot be considered sufficient for the intended application. In fact, the achievable horizontal coverage of 129° remains too limited with respect to the desired observation of the area surrounding the robot. Consequently, satisfying the coverage requirement necessarily involves increasing the number of sensing units, motivating the transition to the three-camera configuration.

4.1.3 Three-Camera Configuration Design

The new configuration with the introduction of a third RGB-D camera was designed by retaining a symmetric arrangement with respect to the longitudinal axis of the robot and introducing a central forward-facing camera between the two lateral sensing units. In this way, the central camera is used to provide a direct observation of the frontal region, while rotating the two lateral cameras makes it possible to extend the overall horizontal coverage beyond that achievable with the two-camera configuration.

The study was repeated with the additional constraint of accommodating a third sensing device on the available mounting surface. Hence, the first step concerned the definition of a physically feasible placement of the three sensors. According to the robotic platform dimensions, and thus the available width, the simplest arrangement with the three cameras positioned side by side along the front side of the robot was not feasible and was therefore immediately discarded. In fact, locating the three sensing units at the same longitudinal coordinate did not satisfy the required mechanical clearances from the platform boundaries and between adjacent cameras. In particular, considering a camera width w_c and the mechanical tolerance m, the central camera itself requires a minimum lateral clearance from each lateral camera. The minimum absolute lateral coordinate of each lateral camera would have to satisfy the following relation:

$$|y_{\text{lat}}| \geq w_c + m, \tag{4.3}$$

which is incompatible with the maximum lateral position allowed by the mounting surface when the three cameras are placed side by side. The lateral cameras were therefore moved backward with respect to the central sensor, which was maintained in the frontal position. The longitudinal position of the lateral cameras had to account for their rotation to avoid collisions with the central camera, since the projected footprint along the longitudinal direction of the robot changes with yaw. Considering the camera width w_c and depth d_c and the camera yaw ψ, the

projected half-extent along the longitudinal direction can be expressed as

$$h_x(\psi) = \frac{w_c}{2}|\sin\psi| + \frac{d_c}{2}|\cos\psi|, \tag{4.4}$$

where $h_x(\psi)$ represents the projected half-extent of the camera along the longitudinal direction.

During the mechanical placement analysis, a yaw angle of 60° was adopted as a reference orientation for evaluating the footprint of the lateral cameras under a large outward rotation. This angle value was used to determine a sufficiently conservative clearance to accommodate their rotation without mechanical interference with the central sensor. At this stage, the 60° orientation therefore represents a mechanical design assumption combined with the mechanical tolerance previously introduced, rather than the final orientation of the cameras, which is subsequently determined through the field-of-view analysis. Based on this constraint, the lateral cameras were sufficiently displaced backward, resulting in a mechanically feasible and symmetric arrangement. The corresponding positions, expressed according to the previously introduced reference frame, are:

$$\begin{aligned} \mathbf{p}_C &= (17.75,\ 0)\ \text{cm}, \\ \mathbf{p}_L &= (9,\ 9)\ \text{cm}, \\ \mathbf{p}_R &= (9,\ -9)\ \text{cm}, \end{aligned} \tag{4.5}$$

corresponding to the central, left, and right cameras, respectively.

These coordinates define the physical arrangement adopted as the starting point for the field-of-view analysis, aimed at determining the optimal orientation of the lateral sensors. Specifically, the central camera was maintained at a yaw angle of 0°, aligned with the forward direction of the robot, while equal and opposite yaw rotations were assigned to the left and right cameras to preserve the symmetry of the sensing configuration. Consequently, the geometric analysis was performed by considering the interaction between the central camera and one of the two lateral cameras, since the same reasoning applies symmetrically to the opposite side. Following the field-of-view model introduced for the two-camera configuration, the relevant boundaries in this case are the external boundary ray of the central camera and the internal boundary ray of the corresponding lateral camera. Considering the left side, the external boundary of the central camera is characterized by the angle $+\alpha$ with respect to the longitudinal axis. Since the central camera is located at (x_C,0), its boundary ray can be expressed as

$$y_C(x) = (x - x_C)\tan(\alpha), \tag{4.6}$$

For the left lateral camera, located at (x_L, y_L) and rotated outward by a yaw angle ψ, the inner boundary of its field of view is characterized by the angle $\psi - \alpha$, and can therefore be expressed as

$$y_L(x) = y_L + (x - x_L)\tan(\psi - \alpha), \tag{4.7}$$

The relative position of these two rays determines whether a blind region or an overlap exists between the central and lateral fields of view. Defining their lateral separation as

$$B(x) = y_L(x) - y_C(x), \tag{4.8}$$

a positive value indicates the presence of a blind region, while a negative value indicates an overlap between the two fields of view. The transition between these two conditions occurs at the intersection of the two boundary rays, where:

$$y_L(x_{\text{int}}) = y_C(x_{\text{int}}), \tag{4.9}$$

and the corresponding intersection distance can be obtained as

$$x_{\text{int}} = \frac{y_L + x_L\tan(\psi - \alpha) - x_C\tan(\alpha)}{\tan(\alpha) - \tan(\psi - \alpha)}. \tag{4.10}$$

The intersection distance thus indicates how far from the robot the blind area between the adjacent sensing regions extends before the overlapping region begins. Increasing the yaw angle of the lateral cameras progressively extends the overall horizontal coverage and reduces the overlap at larger distances. However, this improvement comes at the cost of increasing the near-field blind region and moving the intersection between adjacent fields of view farther from the robot. Since the central camera is aligned with the longitudinal axis and the two lateral cameras are symmetrically oriented by $\pm\psi$, the total angular coverage of the three-camera configuration can be expressed as

$$C_{\text{tot}} = 2(\psi + \alpha) = 2\psi + \text{HFOV}. \tag{4.11}$$

Configurations with different yaw angles were investigated for the lateral cameras, while keeping the alignment of the central camera with the forward direction unchanged. The resulting configurations were evaluated according to the same geometric criteria adopted for the two-camera study. The main results are summarized in Table 4.2.

Lateral yaw	Blind region	Intersection	Overlap at 3 m	Coverage
0°	No blind region	Before operating range	Large	69°
40°	1.42 cm	16.65 cm	165.24 cm	149°
50°	5.58 cm	27.88 cm	110 cm	169°
58°	**9.21** cm	**50.72** cm	**62** cm	**185°**
59°	9.69 cm	56 cm	55.6 cm	187°
60°	10.18 cm	62.64 cm	49 cm	189°
67°	13.85 cm	≈ 290 cm	Almost absent	203°

Table 4.2: Geometric comparison of the investigated three-camera configurations for different lateral-camera yaw angles.

The table shows the described effect on the observable area when increasing the yaw rotation. However, the maximum angular coverage alone does not provide the most suitable sensing arrangement, since increasing too much the rotation also leads to larger and farther blind regions up to the point of losing dangerously observability. In particular, although a yaw angle of 67° provides a total coverage of 203°, the intersection occurs only at approximately 290 cm, resulting in an almost persistent blind region throughout the whole considered operating range.

According to the obtained results, a lateral-camera yaw angle of 58° was selected as the optimal compromise among the considered requirements, since it provided the most balanced solution among the investigated ones. For this configuration, the total horizontal coverage is

$$C_{\text{tot}} = 2(58^\circ) + 69^\circ = 185^\circ, \tag{4.12}$$

thus overcoming the minimum coverage requirement of 150°. At the same time, the maximum blind region is maintained at approximately 9.21 cm, while an overlap of approximately 62 cm is preserved at the maximum operating distance.

Due to the symmetric arrangement of the sensors, the same geometric behavior occurs between the central camera and each lateral camera. Consequently, two separate blind regions are present, one between the central and left camera and the other between the central and right camera. The same reasoning can be repeated

for the overlapping areas. Moreover, the additional sensor in the three-camera configuration allows the forward direction to be continuously and directly observed by the central camera. The resulting arrangement, consisting of one forward-facing central camera and two lateral cameras oriented by ±58°, was therefore adopted as the final configuration of the multi-camera perception payload.

The resulting fields of view of the final three-camera configuration are shown in Fig. 4.2, where the overall sensor coverage can be observed from a top-view perspective in the Gazebo simulation environment.

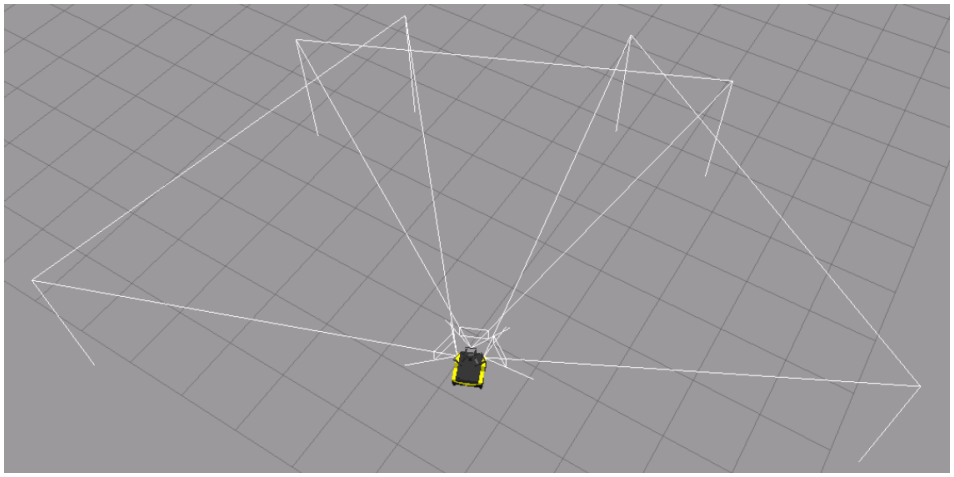


Figure 4.2: Top view of the sensor coverage provided by the final three-camera configuration in the Gazebo simulation environment.

4.2 Local Perception and Tracking Pipeline

The multi-camera perception architecture employed in this work relies on a local perception and tracking pipeline independently operating on the data acquired by each RGB-D camera. Indeed, before addressing the global multi-camera tracking problem, each sensing stream is first independently processed to detect individuals in the observable scene, estimate their spatial state and maintain their identities over consecutive observations. The local tracking layer therefore represents the first level of temporal processing of the perception architecture, converting the frame-based information provided by the individual sensors into local pedestrian tracks.

The implemented local tracking pipeline follows the tracking-by-detection approach introduced in Chapter 3, combining YOLOv8 Pose for person detection

and body-keypoint estimation with StrongSORT for temporal tracking. In particular, the RGB information is additionally integrated with the aligned depth measurements, allowing the estimation of the three-dimensional people position from the image-space detections. The body keypoints estimated by YOLOv8 Pose are also exploited to obtain information about the orientation of detected pedestrians. Hence, the local tracker operates on observations enriched with spatial information rather than relying only on the two-dimensional bounding boxes.

Starting from this baseline architecture, several modifications were introduced in this work to improve the robustness of the local estimates and adapt the pipeline to the requirements of the subsequent multi-camera processing stage. In particular, the three-dimensional position estimation was improved by introducing a median-based combination of the valid 3D body-keypoint measurements, reducing the influence of outliers, such as isolated depth errors. A controlled depth fallback mechanism was additionally implemented to recover a pedestrian position when the available reliable keypoints are insufficient, while rejecting estimates that do not satisfy suitable depth-consistency conditions.

The management of coordinate transformations was also adapted to the mobile nature of the sensing platform. Since the cameras move together with the robot, the transformation of a pedestrian observation from the camera reference frame to the common fixed frame, the map frame, must be consistent with the instant at which the corresponding RGB-D data were acquired. The implementation therefore performs the required TF lookup according to the observation timestamp. Finally, the results produced by the local tracking layer were extended in view of the global multi-camera association. In particular, the visual Re-Identification representation maintained by StrongSORT is explicitly propagated together with the local tracking information. Each resulting pedestrian track can therefore provide the upper processing layer with its local identity, three-dimensional position, velocity, body orientation, reliability, and visual appearance representation.

The following sections first describe how these operations are combined within the processing pipeline associated with a single RGB-D camera and then how the same pipeline is independently instantiated for the three sensors of the developed perception architecture.

4.2.1 Single-Camera Tracking Pipeline

The single local tracker is tasked with converting the acquired sensor data into temporally consistent pedestrian tracks. Specifically, the node receives the RGB

image, the aligned depth image, and the corresponding camera calibration information. Since the estimation of people position requires combining the body keypoints detected in the color image with the corresponding depth measurements, these input streams are processed through a time synchronization mechanism. Additionally, the camera intrinsic parameters needed for the three-dimensional reconstruction are extracted from the `CameraInfo` message.

In particular, the RGB image is processed by the YOLOv8 Pose inference module. As described in Chapter 3, the network provides a bounding box, a detection confidence, and a set of two-dimensional body keypoints for each detected target. For each sufficiently confident body keypoint, the depth value corresponding to its pixel location is retrieved from the aligned depth image. Measurements associated with invalid or unavailable depth values are discarded. Each valid pixel-depth measurement is then deprojected into the three-dimensional camera reference frame according to

$$Z = D(u, v), \qquad X = \frac{(u - c_x)Z}{f_x}, \qquad Y = \frac{(v - c_y)Z}{f_y}. \tag{4.13}$$

where u and v represent the pixel coordinates of the considered keypoint, $D(u, v)$ is the corresponding depth measurement, and f_x, f_y, c_x, and c_y are the focal lengths and principal-point coordinates obtained from the camera intrinsic calibration.

In the implementation developed in this work, keypoints with insufficient prediction confidence or invalid depth are first rejected, while the remaining 3D points are combined through their component-wise median. In particular, considering the set $\mathcal{K}_i$ of valid three-dimensional keypoints associated with pedestrian i, the resulting position is evaluated as

$$\mathbf{p}_i = \begin{bmatrix} \text{median}_{k \in \mathcal{K}_i}(X_k) \\ \text{median}_{k \in \mathcal{K}_i}(Y_k) \\ \text{median}_{k \in \mathcal{K}_i}(Z_k) \end{bmatrix}, \tag{4.14}$$

where the median is independently evaluated along the three spatial coordinates. This strategy replaces the direct averaging of the available keypoint measurements and reduces the influence of isolated keypoints affected by inaccurate depth values. In the implemented configuration, only keypoints having a prediction confidence of at least 0.45 are considered for the 3D estimation.

The controlled fallback mechanism additionally accounts for those cases in which the number of reliable keypoints is insufficient to directly obtain the pedestrian

position. Rather than assigning the depth measured at a single pixel in the bounding-box center, a local region around the center is inspected. The fallback is activated only when a minimum support from valid body keypoints is still available, and the depth samples extracted from the considered region are subjected to additional consistency checks. In particular, the algorithm verifies the availability of a sufficient number of valid depth measurements, limits their admissible depth dispersion, and checks their compatibility with the depth information provided by the available body keypoints. If these conditions are not satisfied, the corresponding detection is discarded rather than propagating an unreliable spatial measurement to the tracking stage.

In parallel with the three-dimensional position estimation, some of the body landmarks obtained from the YOLOv8 Pose inference, including facial, shoulder, and hip keypoints, are exploited to determine the pedestrian orientation, since they provide geometric information about the body configuration from which a yaw angle ψ_i can be estimated for the current pedestrian observation. Hence, the observation supplied to the local StrongSORT tracker is organized as

$$\mathbf{d}_i = [x_{1,i}, y_{1,i}, x_{2,i}, y_{2,i}, X_i, Y_i, Z_i, \psi_i, s_i] \,, \tag{4.15}$$

where $(x_{1,i}, y_{1,i}, x_{2,i}, y_{2,i})$ represents the bounding box returned by YOLOv8 Pose, (X_i, Y_i, Z_i) is the estimated pedestrian position in the camera reference frame, ψ_i is the estimated body yaw, and s_i is the corresponding detection confidence.

Within the StrongSORT instance, for each detection, the appearance information is extracted through the Omni-Scale Network (OSNet), a convolutional neural network specifically designed for person re-identification. The network maps each pedestrian crop according to the corresponding bounding box into a numerical feature vector, referred to as an appearance embedding. The network is trained so that observations with compatible visual appearance are represented by similar feature vectors, thus providing an additional identity-related cue for temporal association. The adopted appearance model is OSNet x0.25, a lightweight width-scaled variant of the original architecture, suitable for limiting the computational requirements of the local perception pipeline. The resulting feature vector is L2-normalized before being used by the tracking algorithm. StrongSORT exploits the incoming observations together with the state of the tracks maintained from previous frames to perform prediction, data association, and track update according to the tracking procedure described in Chapter 3. From the implementation perspective, the frame-level observations are converted into local trajectories, assigning a local tracking identity to each pedestrian and recursively updating its state as new observations become available.

Before publication, the estimated state must be represented in a reference frame common to all sensing pipelines. The pedestrian position is therefore transformed from the corresponding camera optical frame to the fixed map frame through the ROS 2 TF system. Since the cameras are rigidly mounted on a moving robotic platform, it is essential that the transformation used for this operation is temporally consistent with the observation being processed. As anticipated, the timestamp-aware handling prevents measurements acquired at a given instant from being transformed according to a robot pose corresponding to a different instant, which becomes particularly relevant while the platform is moving. The velocity and body orientation are also expressed in the common reference frame. In particular, the velocity initially estimated from the temporal evolution of the pedestrian with respect to the sensing platform is compensated by accounting for the robot motion. At the end of the complete processing chain, each confirmed pedestrian track is therefore characterized by its local tracking ID, three-dimensional position, velocity, body yaw, tracking reliability, and Re-ID appearance representation. These quantities constitute the output of the single-camera pipeline.

4.2.2 Three-Camera Pipeline Deployment

To address the multi-camera tracking objective, each of the three installed sensor devices must contribute by providing its own data stream. To this end, the local tracking pipeline described above was independently instantiated for each of the three RGB-D cameras composing the perception payload. A dedicated ROS 2 launch configuration was defined to simultaneously execute three tracker nodes, associated with the central, left, and right cameras. Each instance receives the field data from its corresponding sensor and independently performs the complete processing chain previously discussed.

As a consequence, the three StrongSORT instances maintain independent local identity spaces. The same physical pedestrian can therefore be associated with different local IDs when observed by different cameras, while identical numerical IDs generated by separate trackers do not necessarily identify the same person. The outputs of the central, left, and right pipelines are respectively published through the ROS 2 topics `/people_center`, `/people_left`, and `/people_right`. Since the resulting pedestrian states are expressed in the common map reference frame, observations originating from different cameras can be directly processed by a common global layer. These three local tracking streams constitute the inputs of the developed global fusion node.

4.3 Fusion Strategy for Global Tracking

Although the outputs provided locally are temporally consistent pedestrian tracks within the individual camera streams, they result from independent tracking processes characterized by separate local identity spaces; overlapping regions may lead to simultaneous observations of the same pedestrian from multiple sensors, while transitions between adjacent camera regions may lead to fragmentation of the local tracking history. Thus, the association of observations across different cameras and the maintenance of a global identity remain unresolved, since the information collected from the local tracking results cannot be directly exploited as a unique representation of detected targets.

To address this problem, a global fusion strategy was developed. In particular, the fusion node is designed to maintain a set of global tracks, each associated with a unique global identifier, generated starting from the incoming observations published by the camera-specific StrongSORT instances. The objective of this additional global tracking level is thus to establish and preserve the correspondence between the local observations and the global tracks over time, independently of the camera currently providing the measurement. In this way, the global identity is decoupled from the local IDs generated by the individual trackers. The final outcome of the global tracking framework therefore consists of a unified set of temporally stable pedestrian tracks.

The design of the global association strategy was inspired by two tracking perspectives. On the one hand, it follows the tracking-by-detection philosophy of modern SORT-based approaches and, in particular, the general BoT-SORT principle of combining motion and geometric consistency with appearance information during data association. On the other hand, the global identity-management problem is closely related to Multi-Camera Multi-Object Tracking approaches traditionally adopted in surveillance systems, where independent tracks produced by different cameras must be associated across different sensing viewpoints to recover persistent identities. In the proposed architecture, these concepts are adapted to a mobile robotic platform equipped with the multi-camera payload, considering that the local pedestrian states are already transformed into the same metric map reference frame before arriving at the upper processing layer and can therefore be directly compared in the physical environment. Hence, the resulting method can be regarded as a BoT-SORT-inspired multi-cue global association strategy, integrating different sources of information from both local observations and the state of tracks maintained at global level. Overall, the strategy is characterized by layered processing, including dedicated mechanisms for handling uncertain or unmatched observations, track recovery, identity-protection, and lifecycle management.

4.3.1 Fusion Architecture and Track Representation

The global fusion system is implemented as a ROS 2 node operating downstream of the three local tracking pipelines. It subscribes to the topics on which the processing results are published and collects the corresponding incoming observations through the respective subscriber callbacks. Rather than processing observations immediately upon reception, the algorithm stores them in an internal shared buffer. Each buffered observation contains the information associated with the local track together with the source camera producing the observation and the acquisition timestamp. The global tracking procedure is executed periodically through the `process()` routine, triggered by a ROS 2 timer at a predefined fusion rate. At each processing cycle, the observations accumulated in the buffer are retrieved and handled as the set of input measurements currently available to the global tracking layer. This organization addresses the asynchronous reception of the camera-specific tracking results and thus allows observations coming from different channels to be jointly processed. To support the information required by the developed association strategy, the communication interface between the local and global tracking levels was extended through the definition of a custom ROS 2 message. In particular, the message allows the Re-ID appearance descriptor extracted during the local tracking to be transported together with the corresponding pedestrian information. Therefore, the visual representation remains explicitly associated with the corresponding detection and becomes directly available to the fusion node without requiring an independent feature stream and an additional synchronization mechanism.

Each incoming local observation can consequently be represented at the global level as

$$\mathbf{d}_j = [x_j,\, y_j,\, \theta_j,\, t_j,\, c_j,\, l_j,\, \mathbf{e}_j]\,, \tag{4.16}$$

where x_j and y_j denote the estimated pedestrian position expressed in the common map reference frame, θ_j represents the estimated pedestrian orientation, t_j is the timestamp associated with the observation, c_j identifies the source camera, l_j is the identity assigned by the corresponding local StrongSORT instance, and $\mathbf{e}_j$ denotes the associated Re-ID appearance embedding.
It is worth noticing that the camera source and local identity are maintained as distinct information because of the independence of the identity spaces within which the individual trackers operate, and the possibility of equal local IDs from the different pipelines not necessarily implying the correspondence to the same physical person.

Each global track represents instead the current state associated with a pedestrian

identified at the global level and is assigned a unique global ID generated by the fusion system. The state contains the pedestrian planar position, estimated velocity and orientation, together with temporal and lifecycle information required to propagate and manage the track over time. In addition, the global representation preserves information derived from previous successful associations, including the local IDs and a visual appearance representation obtained from the Re-ID embeddings.

Accordingly, the state maintained for the i-th global track can be represented as

$$\mathcal{T}_i = \left\{ ID_i^G,\, x_i,\, y_i,\, v_{x,i},\, v_{y,i},\, \theta_i,\, t_i^{\text{last}},\, h_i,\, m_i,\, \mathcal{L}_i,\, \mathbf{e}_i^G \right\}. \tag{4.17}$$

where ID_i^G denotes the unique global identifier, x_i and y_i represent the current pedestrian position, $v_{x,i}$ and $v_{y,i}$ are the estimated velocity components, θ_i is the orientation, and t_i^{last} denotes the timestamp of the most recent successful update. The variables h_i and m_i represent the number of successful updates and missed associations, respectively, and are subsequently exploited for track lifecycle management. Finally, $\mathcal{L}_i$ represents the memory of camera-specific local identities associated with the global track, while $\mathbf{e}_i^G$ denotes its maintained appearance representation.

The distinction between local and global identities represents a fundamental aspect of this architecture. Local IDs are not propagated directly as final identities, but each global track maintains a camera-indexed memory of the local identities that have previously been successfully associated with it, as additional support for the association stage. Considering the three sensing pipelines, this information can be represented as

$$\mathcal{L}_i = \left\{ c \mapsto l_i^{(c)} \mid c \in \mathcal{C} \right\}, \qquad \mathcal{C} = \{\text{center, left, right}\}. \tag{4.18}$$

where $l_i^{(c)}$ denotes the local identity associated with global track i for camera c, when such a correspondence has previously been established. In this way, the same person can progressively be related to the different local identities through which it is represented by the three independent trackers, while preserving its global identity. In fact, this information constitutes a camera-specific association history between the local and global tracking levels.

A temporal representation is also maintained for the visual appearance of each global track. When a newly associated observation provides a valid Re-ID embedding, the existing appearance descriptor is not simply replaced by the most

recent feature vector. Instead, it is progressively updated through an Exponential Moving Average (EMA), according to the following relation:

$$\mathbf{e}_{i,t}^{G} = \text{norm}\left(\alpha_e \mathbf{e}_{i,t-1}^{G} + (1-\alpha_e)\mathbf{e}_{j,t}\right), \qquad \alpha_e = 0.85. \tag{4.19}$$

where $\mathbf{e}_{i,t-1}^{G}$ represents the appearance descriptor previously maintained for global track i, $\mathbf{e}_{j,t}$ is the Re-ID embedding carried by the newly associated local observation, and α_e determines the relative contribution of previous and current appearance information. In the implemented configuration, $\alpha_e = 0.85$, so that the accumulated representation evolves gradually as new observations are associated with the pedestrian. The resulting feature vector is normalized before being stored as the updated global appearance representation. Maintaining the appearance information at the global level is particularly relevant in the multi-camera architecture. It enriches the fusion strategy, since the descriptor associated with a global track is not related exclusively to the most recent local observation or to a single sensing pipeline, but progressively integrates visual information from the observations successfully matched.

4.3.2 Input Preprocessing and Observation Filtering

The observations collected during each fusion cycle are subjected to a preliminary filtering stage to get ready for the next association phase. The objective of this processing is to provide the association mechanism with a temporally consistent and non-redundant set of measurements. In fact, since the local tracking pipelines operate asynchronously and the fusion routine executes periodically, the internal buffer may contain observations belonging to different messages from the same camera. Additionally, the overlapping sensing regions may cause the simultaneous generation of multiple local detections corresponding to the same person.

The first filtering operation addresses the temporal consistency problem. As explained, there is the chance of having more than one generated message transmitted by the same local tracker accumulated between two consecutive executions of the global `process()` routine. Filtering the incoming observations is useful to avoid considering them jointly, with the risk of making older and more recent local states coexist and be processed in the same fusion cycle, affecting and artificially increasing the ambiguity of the following association problem. For this reason, only the observations belonging to the most recent message received from each camera are retained. For a given camera c, let $\mathcal{D}_c$ denote the set of observations currently available in the fusion buffer. The timestamp of the latest message associated with that camera is evaluated as

$$t^c_{\text{latest}} = \max_{\mathbf{d}_j \in \mathcal{D}_c} t_j, \tag{4.20}$$

where t_j denotes the acquisition timestamp of observation $\mathbf{d}_j$. An observation is consequently retained only if its timestamp satisfies

$$t_j \simeq t^c_{\text{latest}}, \tag{4.21}$$

where the equality is evaluated within a small numerical tolerance.

In this way, just target states contained in the most recent message from each camera are preserved. This operation does not force the three camera streams to share exactly the same timestamp, but instead it provides the fusion layer with the freshest available local information from each independent tracker. The residual temporal differences between camera observations are subsequently handled through timestamp-aware global track prediction, where each global track is predicted at the timestamp of the observation involved in the comparison.

The second preprocessing mechanism addresses duplicated observations. The primary source of observation redundancy arises from the partial overlap between the fields of view of the installed cameras. The cross-camera case thus represents the natural consequence of the multi-camera configuration under analysis and consequently constitutes the main reason behind the duplicate suppression strategy. Moreover, to improve the robustness of the fusion process, the strategy was extended to account for occasional same-camera duplicates. Indeed, even though these are not expected during the regular operation of the local StrongSORT tracker, temporary association failures may occasionally cause an old track and a newly initialized track corresponding to the same pedestrian to coexist for a short interval. The duplicate suppression criterion adopted is deliberately conservative and depends on whether the two observations originate from the same or from different cameras.

Considering two observations $\mathbf{d}_i$ and $\mathbf{d}_j$, their Euclidean distance is given by

$$\delta_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}. \tag{4.22}$$

For what concerns the observations produced by the same camera, the algorithm does not rely on geometric proximity alone. Instead, two observations are classified as duplicates only when they are sufficiently close in space and their Re-ID descriptors also exhibit strong appearance compatibility. A strict tolerance is imposed on

the appearance distance, which reduces the risk of merging two distinct pedestrians in case they are simply spatially close within the same sensing stream.

For observations arriving from different cameras, the duplicate test similarly exploits both geometry and visual appearance, whenever valid Re-ID descriptors are available. In this case, the observations are classified as duplicates when their spatial distance is below the *cross-camera duplicate threshold*, and their appearance distance is sufficiently small. If the Re-ID information is unavailable, duplicate suppression can still be performed, but only by imposing a more severe geometric threshold. This design choice intentionally restricts the geometry-only condition, since spatial proximity alone provides weaker evidence that two measurements acquired from different viewpoints correspond to the same person. The appearance information is thus exploited to strengthen the confidence in stating two spatially close local observations as redundant measurements of the same pedestrian. Only after multiple observations have been identified as duplicates, the algorithm applies a ranking procedure to determine which observation should be the one to be kept. In particular, the survivor is selected according to the amount of identity-related information associated with the observation itself, rather than simply according to its arrival order. An observation carrying a valid Re-ID embedding is preferred over one without appearance information. If both candidate observations have the same embedding availability, the presence of a valid local tracker identity is used as an additional preference criterion. Only when both observations remain equivalent according to these conditions, then it is sufficient to preserve the first retained one. The survivor-selection criteria are therefore evaluated according to the following priority order:



Figure 4.3: Priority hierarchy adopted for survivor selection during duplicate suppression.

After these prefiltering operations, the algorithm determines which of the global tracks currently stored in memory are eligible for the standard (or normal) association stage. This operation introduces the first use of track temporal age in the fusion procedure, being a first important aspect of the track's lifecycle management mechanism.

For the i-th global track, its age at the current fusion instant is defined as

$$a_i = t_{\text{now}} - t_i^{\text{last}}, \tag{4.23}$$

where t_i^{last} is the timestamp of its most recent successful update. A track is considered normally matchable only if

$$a_i \leq T_{\text{match}}, \tag{4.24}$$

where T_{match} corresponds to the configurable `match_max_age` parameter.

Tracks whose age overcomes this threshold are excluded from the standard Hungarian association stage, since they start to be considered less reliable with their predicted state becoming less representative of the current pedestrian position. However, the logic does not simply remove those tracks. As no longer normally matchable tracks, they are retained in the memory and can still be considered by a dedicated recovery procedure, provided that they remain alive within the larger reactivation horizon.

Consequently, the preprocessing stage produces two sets of inputs: the filtered observation set $\mathcal{D}_{\text{filtered}}$ and the set of currently matchable global tracks

$$\mathcal{T}_{\text{matchable}} = \{\mathcal{T}_i \mid a_i \leq T_{\text{match}}\}. \tag{4.25}$$

4.3.3 Motion Prediction and Geometric Gating

This section concerns the evaluation of spatial compatibility once the matchable global tracks have been identified. At this stage the filtered observations arriving from independent processing routes are not necessarily associated with exactly the same timestamp, and consequently the current position stored within a global track cannot be directly compared with every incoming measurement. In particular, the motion state maintained by each global track is exploited to estimate its expected position at the specific timestamp of the observation under consideration. The motion information associated with each track is the result of an estimation process based on successive successful updates. When a new observation is associated with the i-th global track, an instantaneous velocity measurement is obtained from the displacement between the previous track position and the newly associated position. Considering an update occurring at time t_j, the elapsed time since the previous track update is

$$\Delta t_i = t_j - t_i^{\text{last}}, \tag{4.26}$$

and the corresponding measured velocity components are computed as

$$v_{x,i}^{\text{measured}} = \frac{x_j - x_i}{\Delta t_i}, \qquad v_{y,i}^{\text{measured}} = \frac{y_j - y_i}{\Delta t_i}. \tag{4.27}$$

When the track is initially created, the velocity components are set to zero, since a single observation does not provide sufficient information to estimate the pedestrian motion. The first valid velocity measurement, obtained at the next successful update, initializes the motion estimate of the track. During the following updates, abrupt variations caused by measurement noise are reduced by applying an exponential smoothing mechanism to the velocity components. The updated velocity is therefore obtained as

$$v_{x,i}^{+} = \alpha_v v_{x,i} + (1 - \alpha_v) v_{x,i}^{\text{measured}}, \tag{4.28}$$

$$v_{y,i}^{+} = \alpha_v v_{y,i} + (1 - \alpha_v) v_{y,i}^{\text{measured}}, \tag{4.29}$$

where α_v controls the contribution of the previous motion estimate with respect to the newly measured velocity. In the implemented configuration, $\alpha_v = 0.7$, thus preserving temporal smoothness while still allowing the global track to adapt to changes in pedestrian motion.

This velocity estimate is then used to perform timestamp-aware motion prediction, through which the same global track position may be propagated to slightly different temporal instants when it is compared with measurements from different cameras. This directly accounts, during the pairwise geometric comparison, for the residual temporal differences possibly still present after the input preprocessing stage, without requiring an explicit synchronization of the three local tracking streams. Therefore, the prediction is performed independently for each candidate track-observation pair. When a global track $\mathcal{T}_i$, last updated at t_i^{last}, is compared with an observation $\mathbf{d}_j$ acquired at t_j, the corresponding prediction interval is defined as

$$\Delta t_{ij} = t_j - t_i^{\text{last}}. \tag{4.30}$$

A possible approach to this propagation would consist of adopting a constant-velocity model, according to which the velocity estimated at the last successful update is assumed to remain unchanged throughout the entire prediction interval. Such an assumption provides a reasonable approximation over short temporal gaps but becomes increasingly uncertain when a global track remains without new

observations. The uncertainty is strictly related to the unknown behavior the pedestrian could assume during an unobserved interval. Indeed, the pedestrian may continue moving with approximately the same velocity, but may also slow down, stop, or change direction. Especially in these latter cases, working with a constant-velocity model could be significantly misleading, since indefinitely propagating the last estimated velocity would progressively move the predicted track position far away from the actual pedestrian location. For this reason, a more conservative prediction strategy was adopted in this work. Instead of maintaining the estimated velocity constant throughout the entire prediction interval, its contribution to the predicted displacement is progressively reduced as the time elapsed from the last successful observation increases. This behavior is implemented through an exponential velocity-decay model,

$$\mathbf{v}_i(\tau) = \mathbf{v}_i e^{-k_v \tau}, \tag{4.31}$$

where $\mathbf{v}_i = [v_{x,i}, v_{y,i}]^T$ is the velocity estimated at the last successful track update, k_v is the velocity-decay gain, and τ represents the elapsed prediction time. The exponential decay should not be interpreted as a physical model according to which the pedestrian is assumed to decelerate exponentially. Instead, it represents a conservative mechanism for progressively reducing the influence of a velocity estimate that becomes less reliable.

Since position prediction depends on the displacement accumulated over the considered temporal interval, the exponentially decaying velocity is integrated from the last track update to the timestamp of observation $\mathbf{d}_j$:

$$\Delta\mathbf{p}_i(\Delta t_{ij}) = \int_0^{\Delta t_{ij}} \mathbf{v}_i e^{-k_v \tau}\, d\tau = \mathbf{v}_i \frac{1 - e^{-k_v \Delta t_{ij}}}{k_v}. \tag{4.32}$$

where $\Delta\mathbf{p}_i(\Delta t_{ij})$ represents the predicted displacement of the i-th global track accumulated over the prediction interval Δt_{ij}.

The predicted position of the global track at the observation timestamp is therefore obtained as

$$\hat{\mathbf{p}}_i(t_j) = \mathbf{p}_i(t_i^{\text{last}}) + \mathbf{v}_i \frac{1 - e^{-k_v \Delta t_{ij}}}{k_v}. \tag{4.33}$$

where $\mathbf{p}_i(t_i^{\text{last}})$ denotes the position stored at the last successful update of the i-th global track, while $\hat{\mathbf{p}}_i(t_j)$ represents its predicted position at the timestamp t_j

of the considered observation.

Compared with a conventional constant-velocity extrapolation, for which the predicted displacement would be $\mathbf{v}_i \Delta t_{ij}$, the adopted model progressively limits the displacement produced by a velocity estimate starting to become stale. The two approaches are comparable, showing a similar behavior, for short prediction intervals, whereas the effect of the employed exponential decay becomes more relevant with the increment of the temporal gap. The more conservative strategy selected induces an inherent trade-off: when the unobserved person keeps moving approximately according to the previously estimated velocity, this can underestimate the actual displacement; conversely, it enables limiting the prediction drift resulting from maintaining an outdated velocity estimate when the person changes behavior during the unseen interval.

Motion prediction is therefore not intended to provide an independently sufficient estimate of pedestrian identity. Rather, it provides a temporally aligned geometric reference for the association process, which is next complemented by adaptive geometric constraints and identity-related information.

Once the global track has been propagated to the timestamp of the considered observation, its geometric compatibility with the measurement can be evaluated directly in the common map reference frame. For a candidate association between global track $\mathcal{T}_i$ and observation $\mathbf{d}_j$, the prediction-to-observation distance is defined as

$$d_{ij}^{\text{geom}} = \|\mathbf{p}_j - \hat{\mathbf{p}}_i(t_j)\|_2 = \sqrt{(x_j - \hat{x}_i(t_j))^2 + (y_j - \hat{y}_i(t_j))^2}. \tag{4.34}$$

where d_{ij}^{geom} represents the Euclidean distance between the predicted position of the i-th global track and the position of the j-th observation at timestamp t_j.

This distance provides the basic geometric information used to determine whether an incoming observation represents a spatially acceptable candidate for a given global track. However, the admissible spatial discrepancy cannot be appropriately represented by a single fixed threshold for every tracking condition. In particular, the uncertainty associated with the expected pedestrian position generally becomes more relevant for moving tracks, especially for those which are expected to move faster, and for tracks considered "older", due to the increase of the time elapsed since their last successful update. For this reason, the fusion strategy employs a dynamic geometric threshold whose value depends on both the estimated velocity of the global track and its temporal age. The track speed is computed as

$$s_i = \sqrt{v_{x,i}^2 + v_{y,i}^2}, \tag{4.35}$$

while its age is computed as in the previous section.

Starting from a nominal distance threshold d_{base}, the admissible geometric region is adapted according to

$$d_i^{\text{dyn}} = d_{\text{base}} + g_v s_i a_i, \tag{4.36}$$

where the velocity gain g_v controls the contribution of track motion and temporal age to the enlargement of the association region.

The threshold is bounded between predefined minimum and maximum values, preventing the geometric constraint from becoming either excessively restrictive or excessively permissive. This preliminary geometric threshold is thus made adaptive by introducing context-dependent corrections: recently updated and slowly moving tracks are associated with a relatively compact admissible region, whereas a larger tolerance can be provided when the track characteristics lead to a greater positional uncertainty. The resulting dynamic threshold is not used as the final hard acceptance boundary in this strategy; rather, it constitutes a reference describing the expected spatial compatibility of a track-observation pair. Starting from this value, a second geometric quantity is constructed to define the effective candidate region used before the global assignment stage.

The effective geometric gate can be expressed as

$$d_{ij}^{\text{allow}} = \min\left(d_{\max}^{\text{candidate}}, \gamma_g d_i^{\text{dyn}} + \delta_g + \delta_l I_{\text{match}}\right), \tag{4.37}$$

where γ_g is a multiplicative gate-expansion factor to proportionally enlarge this reference region when defining the actual candidate gate, δ_g is an additional fixed tolerance term, δ_l provides a further limited relaxation when short-term camera-local identity continuity supports the considered correspondence, I_{match} is the value activating the relaxation, being equal to one when the incoming observation is associated with a valid recent camera-local ID match for the considered global track or zero otherwise, and $d_{\max}^{\text{candidate}}$ defines the upper bound on the admissible candidate distance.

This distinction between the dynamic threshold d_i^{dyn} and the effective candidate distance d_{ij}^{allow} is important because, beyond the reference estimate introduced by

the former, the latter defines the broader region within which a track-local detection pair is still admitted in the global association process. In this way, potentially correct correspondences are not prematurely discarded solely because of moderate prediction errors, but the process can be supported also by the available short-term identity continuity information. This allows maintaining a balance between robustness and selectivity. Nevertheless, the geometric information is not treated as sufficient evidence of identity correspondence alone, but as one component of a broader global association decision to be further interpreted together with the available identity information. This prevents the geometric prediction from becoming the unique association factor, which is particularly important in cases where pedestrian motion is uncertain or multiple individuals are spatially close. Therefore, with the motion-prediction and geometric-gating stage the implemented logic provides, for each candidate track-observation pair, a temporally aligned predicted position, the corresponding geometric discrepancy, and an adaptive indication of its compatibility. The resulting geometric quantity represents just the first building block to determine the final correspondence of a pair.

4.3.4 Multi-Cue Global Data Association

The node implemented in this work jointly exploits multiple sources of information. In particular, it combines four cues: geometric consistency, the similarity between Re-ID appearance representations, the short-term consistency of the camera-local track identifiers, and the temporal freshness of the global track. These contributions are converted into normalized compatibility scores and then combined into a single overall association score. So, the different cues contribute to the evaluation according to the evidence available for each candidate pair. This represents the main multi-cue component of the BoT-SORT-inspired association strategy adopted in the developed fusion system.

The overall association score can therefore be expressed as

$$S_{ij}^{\text{raw}} = w_g S_{ij}^{\text{geom}} + w_a S_{ij}^{\text{app}} + w_l S_{ij}^{\text{local}} + w_t S_i^{\text{time}}. \tag{4.38}$$

where S_{ij}^{geom}, S_{ij}^{app}, S_{ij}^{local}, and S_i^{time} represent geometric, appearance, local-identity, and temporal compatibility, respectively. The coefficients w_g, w_a, w_l, and w_t denote weighting factors defining the relative contribution of the four cues. They are normalized so that their sum is equal to one. In the implemented configuration, geometry and appearance information receive the largest weights, contributing the most and thus acting as the dominant criteria, while local identity and temporal freshness act as complementary information assigned with lower weights.

Geometric compatibility
The geometric contribution is derived from the distance between the predicted global-track position and the incoming observation. It is defined as

$$S_{ij}^{\text{geom}} = \text{clip}\left(1 - \frac{d_{ij}^{\text{geom}}}{d_{ij}^{\text{allow}}}, 0,1\right), \tag{4.39}$$

where d_{ij}^{geom} is the Euclidean distance between the predicted track position and the incoming observation, while d_{ij}^{allow} is the allowed association distance introduced in the previous section. Therefore, an observation located close to the predicted track position produces a geometric compatibility score close to one, while the score decreases as the observation approaches the boundary of the admissible region. The allowed distance consequently acts as a normalization scale, making the geometric contribution relative to the candidate region defined for the considered track-observation pair.

Appearance compatibility
The appearance contribution exploits the Re-ID representations extracted by the local StrongSORT pipelines and propagated to the global fusion stage. Since the feature vectors are normalized, their appearance similarity can be evaluated through cosine similarity. Starting from this measure, the appearance distance used by the fusion logic for Re-ID-based comparisons is defined as

$$d_{ij}^{\text{app}} = 1 - \frac{\mathbf{e}_i^G \cdot \mathbf{e}_j}{\|\mathbf{e}_i^G\|_2 \|\mathbf{e}_j\|_2}. \tag{4.40}$$

Therefore, highly similar appearance representations produce a cosine similarity close to one and consequently an appearance distance close to zero, whereas increasingly dissimilar representations result in a larger appearance distance. For its contribution to the weighted multi-cue formulation, the appearance distance is expressed according to the common compatibility-score convention adopted for the different association cues:

$$S_{ij}^{\text{app}} = \text{clip}\left(1 - d_{ij}^{\text{app}}, 0,1\right). \tag{4.41}$$

In this representation, a value close to one indicates strong appearance compatibility, while a value close to zero indicates weak appearance compatibility. Although the resulting score directly corresponds to the cosine similarity from

which the appearance distance is derived, this quantity is explicitly defined to have a uniform representation of the different cues within the multi-cue association strategy. Moreover, through the clipping operation, the appearance contribution is constrained to the interval [0,1], preventing negative contributions to the weighted association score that could otherwise result from negative cosine similarity values. When a valid Re-ID representation is not available for either element of the candidate pair, the observation is not automatically interpreted as an appearance mismatch. Instead, a predefined neutral contribution, `no_reid_score`, is assigned to the appearance cue, allowing the remaining association information to determine the compatibility of the pair.

Camera-local identity compatibility

The third contribution exploits the camera-local identities. Although they cannot be interpreted as the final ones at the global layer, these identifiers can provide useful short-term continuity information when a new received observation originates from a camera for which the considered global track already stores a local-ID association.

For this purpose, the relationship between the incoming observation and the local-ID memory of the global track is classified into four conditions:

$$r_{ij}^{\text{local}} \in \{\text{match}, \text{unknown}, \text{none}, \text{conflict}\}. \tag{4.42}$$

A match indicates that the same camera-local identifier has recently been associated with the global track; unknown indicates that no currently valid correspondence can be established since the track does not have a sufficiently recent identity association for that camera; none denotes the absence of usable local-ID information; and conflict indicates that the same camera currently provides an identifier different from the one stored for that track.

The local relation is then converted into a compatibility score:

$$S_{ij}^{\text{local}} = \begin{cases} 1.00, & r_{ij}^{\text{local}} = \text{match}, \\ 0.55, & r_{ij}^{\text{local}} = \text{unknown}, \\ 0.45, & r_{ij}^{\text{local}} = \text{none}, \\ 0.10, & r_{ij}^{\text{local}} = \text{conflict}. \end{cases} \tag{4.43}$$

Hence, this ordering treats a recent local-ID match as supporting evidence but without letting the local tracker identifier become a permanent or dominant global identity constraint. In particular, the validity of a local-ID match is time dependent:

when the stored correspondence becomes sufficiently old, the relation returns to unknown. At this point, geometry and appearance cues are needed to re-establish the association.

Temporal compatibility
Finally, the temporal contribution reflects how recently the global track has been successfully updated. Given its current age a_i, the temporal score used during normal association is

$$S_i^{\text{time}} = \text{clip}\left(1 - \frac{a_i}{T_{\text{match}}}, 0{,}1\right). \tag{4.44}$$

where T_{match} represents the maximum age for which a global track can be involved in normal matching, defined in the implementation through the `match_max_age` parameter.

A recently updated track therefore receives a temporal score close to one, whereas this contribution decreases as the track becomes older. The temporal term does not determine identity by itself; rather, it slightly supports correspondences involving tracks whose state is still temporally reliable.

The compatibility factors are combined in the weighted sum S_{ij}^{raw} introduced above. However, the resulting value is not directly used as the final score entering the cost matrix. This is because additional safeguards are introduced to reduce the risk of wrong assignments when the individual cues provide partially conflicting evidence. A sensitive condition occurs when the relation with the incoming camera-local identifier is classified as conflict. As said, this piece of information alone cannot provide absolute evidence of a different person, since fragmentations of a local StrongSORT trajectory may also occur. At the same time, just ignoring the conflict and going on with the association on the basis of the weighted score could produce a global ID transfer to a nearby pedestrian. For this reason, a conflicting local-ID relation is admitted as a candidate only when both geometry and appearance provide sufficiently strong supporting evidence, both satisfying stricter dedicated thresholds. If this is not the case, the candidate pair is rejected before participating in the assignment process.

For the candidate pairs that remain admissible, the raw multi-cue score is further refined through a set of soft penalties designed to reduce the compatibility of associations presenting potentially risky characteristics. The cases considered for the application of these penalties are:

1. A spatial jump larger than the expected short-term displacement, when it is not supported by a recent local-ID match;
2. Local-ID conflicts extending beyond a restrictive spatial range;
3. Cross-camera associations classified as unknown, when the observation is located relatively far from the predicted global track position;
4. A candidate lying outside the tighter dynamic geometric threshold introduced in Section 4.3.3, even if it still remains inside the broader effective candidate region.

This last case shows how the dynamic threshold keeps having a role as an indication of expected geometric consistency, also during the multi-cue evaluation.

Therefore, denoting the cumulative penalty associated with pair (i, j) as P_{ij}, the resulting score can be expressed as

$$S_{ij} = \text{clip}\left(S_{ij}^{\text{raw}} - P_{ij}, 0{,}1\right). \tag{4.45}$$

The application of these penalties is nevertheless conditioned by the strength of the available appearance evidence. Indeed, when the Re-ID compatibility exceeds a predefined strong-similarity level, appearance information can slightly compensate for some of the soft geometric and local-continuity penalties. This is useful, for example, when a pedestrian changes camera and the camera-local identity relation is still unknown, or when the actual displacement slightly exceeds the expected dynamic region. In this way, Re-ID can increase tolerance to imperfect motion prediction, but without eliminating the geometric constraints of the global tracking process.

At this point, only candidate pairs whose final compatibility exceeds a minimum association score are retained as admissible. Their compatibility score is converted into an association cost according to

$$C_{ij} = 1 - S_{ij}. \tag{4.46}$$

Hence, high compatibility produces a low association cost, whereas the strategy applies a masking mechanism to those pairs whose final score does not satisfy the minimum requirement. They are assigned a prohibitively large cost (`BIG_COST`) and are effectively excluded from the assignment problem. The resulting cost matrix

therefore makes only track-observation correspondences that satisfy the matching requirements available for valid assignment.

The assignment is performed using the Hungarian algorithm, which processes the resulting matrix and returns the globally optimal one-to-one assignment between the currently matchable global tracks and the available observations. As already introduced in Chapter 3, the Hungarian algorithm minimizes the total association cost under the constraint that each track and each observation can participate in at most one selected correspondence. In the proposed fusion strategy, however, the assignment returned by the global optimization is interpreted as an association proposal rather than an immediately accepted match. This distinction is introduced because minimization of the total cost does not necessarily guarantee that every selected correspondence is sufficiently discriminative at the individual identity level. In particular, two competing associations may present almost equivalent multi-cue scores. For this reason, the implemented node performs post-process safety checks. In particular, each assignment proposed by the Hungarian algorithm undergoes an additional ambiguity test before it is accepted and used to update the corresponding global track. For a selected pair (i, j), the compatibility score of the proposed association is compared with the best alternative valid score involving either the same global track or the same incoming detection. Let S_{ij} denote the score of the selected pair and S_{ij}^{comp} the score associated with its strongest competitor. The assignment is considered sufficiently discriminative only if

$$S_{ij} - S_{ij}^{\text{comp}} \geq \Delta S_{\text{amb}}, \tag{4.47}$$

where ΔS_{amb} represents the minimum required score separation. If the difference falls below this margin, the association is classified as ambiguous and rejected. The test therefore evaluates not only whether the selected correspondence has a high absolute compatibility, but also whether it is clearly preferable to the competing hypotheses. This post-assignment validation is particularly relevant in crowded or geometrically ambiguous conditions, in which the optimal assignment is not that informative because it is not able to provide a sufficiently clear distinction with respect to the best alternative. The ambiguity margin therefore introduces a confidence criterion based on the relative separation between competing identity matches.

Consequently, the normal global association process can be summarized as a hierarchical decision strategy:

1. Geometric gating first restricts the set of physically plausible hypotheses;

2. The multi-cue score ranks the admissible pairs using spatial, appearance, local-identity, and temporal information;
3. Dedicated penalties reduce the confidence of potentially risky correspondences;
4. The Hungarian algorithm determines the globally optimal one-to-one assignment;
5. Post-assignment ambiguity checks finally validate each proposed correspondence before it is applied to the global track.

Finally, the association that has successfully passed all the checks is considered accepted and can thus be used to update the global track.

4.3.5 Local Identity Continuity and Re-ID Memory Management

This section is intended to justify a more conservative criterion introduced in the implemented system when updating an active global track. The successful association passing all the validation steps is immediately employed to update the kinematic state of the global track, thus in terms of track position, velocity, orientation, and temporal state. A dedicated step is then executed for what concerns the update of the identity-related information, requiring additional identity-trust conditions. This distinction does not represent a second verification of whether the observation belongs to the selected pedestrian, which has already been addressed by the association and post-assignment validation stages. Rather, it protects the sensitive information that strongly affects identity decisions. In particular, the camera-local ID history, the corresponding local-ID ownership information, and the global Re-ID representation are updated only when the accepted correspondence also satisfies stricter identity-consistency conditions, thereby reducing the risk that uncertain observations contaminate the identity memory of the global track.

The trust evaluation requires stronger support from geometric and appearance information as the reliability of the local-ID relation decreases, especially in the presence of an unknown or conflicting local-identity relation. In particular:

- A *match* local-ID relation requires the geometric distance to be below 0.55 m; when Re-ID information is available, the cosine distance is additionally required to be below 0.25;
- An *unknown* local-ID relation requires the geometric distance to be below 0.45 m and the cosine distance to be below 0.15;

- A *conflict* local-ID relation requires the geometric distance to be below 0.25 m and the cosine distance to be below 0.10.

The *none* condition, corresponding to the absence of usable local-ID information, is handled conservatively according to the same identity-trust limits adopted for an *unknown* relation.

When the identity update is considered trustworthy, the camera-specific local-ID history is updated with the newly confirmed correspondence and its timestamp. The corresponding local-ID ownership information is also refreshed, recording that the considered camera-local track is currently associated with the selected global track. This information provides an additional safeguard against the creation of a second global track from a local track that is already reliably associated with an existing one. At the same time, when a valid Re-ID descriptor is available, the global appearance representation is updated through the exponential moving average introduced in Section 4.3.1.

If the additional identity-trust conditions are not satisfied, the accepted observation still updates the kinematic and temporal state of the global track, while the existing identity-related information is left unchanged. In this way, the fusion node separates short-term state estimation from longer-term identity preservation: the former maintains the temporal evolution of the global track, while the latter is updated more conservatively to provide stable identity information for future track-management operations.

4.3.6 Track Recovery and Global ID Preservation

At the end of the normal association process, an incoming observation may remain unmatched. In this case, directly creating a new track to be coupled with the unmatched detection could unnecessarily fragment the trajectory of a pedestrian already represented within the fusion system through another still active track. For this reason, the fusion strategy proposed in this work first attempts a reactivation action trying to recover a previously existing track. The main recovery mechanism is devoted to *stale global tracks*, namely tracks whose age has exceeded the maximum allowed age for normal association, so they are too old to participate in the normal matching process but are still retained in the memory of the fusion system. Recalling the definition of the age of a global track:

$$a_i = t_{\text{now}} - t_i^{\text{last}}, \tag{4.48}$$

a track becomes eligible for stale recovery when

$$t_{\text{match}} < a_i \leq t_{\text{reactivation}}, \tag{4.49}$$

where $t_{\text{match}} = 2.6\,\text{s}$ represents the age threshold in normal matching, while $t_{\text{reactivation}} = 5.0\,\text{s}$ defines the maximum temporal horizon within which an existing global track can still be recovered. Therefore, this stage explicitly considers identities that were excluded from the normal Hungarian assignment because of their temporal age but have not yet been permanently removed from the fusion system.

Stale tracks are associated with a longer observation gap, and the uncertainty of their motion prediction is thus higher. For this reason, the recovery stage evaluates geometric compatibility with respect to the last stored position of the global track and adopts a wider spatial candidate region than the one established in normal matching. Starting from the dynamic distance threshold d_{ij}^{dyn} introduced in Section 4.3.3, a preliminary recovery gate can be expressed as

$$\tilde{d}_{ij}^{\text{rec}} = \gamma_r d_{ij}^{\text{dyn}} + \delta_r, \tag{4.50}$$

where γ_r is a multiplicative factor that enlarges the dynamic association threshold, while δ_r represents an additional fixed spatial margin. In the implemented configuration, they correspond to the `recovery_gate_scale` and `recovery_gate_extra` parameters, respectively.

A recent camera-local ID match provides a small additional spatial tolerance, while a sufficiently strong Re-ID correspondence can further enlarge the recovery region when the local-ID relation is either *match* or *unknown*. The resulting gate is finally constrained within a predefined minimum and maximum recovery distance. Therefore, in compact form, the effective recovery gate can be represented as

$$d_{ij}^{\text{rec}} = \text{clip}\left(\tilde{d}_{ij}^{\text{rec}} + \mathbb{I}\left[r_{ij}^{\text{local}} = \text{match}\right]\delta_l + \mathbb{I}\left[\mathcal{C}_{ij}^{\text{ReID}}\right]\delta_{\text{ReID}}, d_{\min}^{\text{rec}}, d_{\max}^{\text{rec}}\right), \tag{4.51}$$

with $\delta_l = 0.05\,\text{m}$, $\delta_{\text{ReID}} = 0.30\,\text{m}$, $d_{\min}^{\text{rec}} = 0.50\,\text{m}$, and $d_{\max}^{\text{rec}} = 2.20\,\text{m}$, where $\mathcal{C}_{ij}^{\text{ReID}}$ denotes the strong Re-ID condition described above. The appearance-based extension does not directly force a recovery, since the candidate must still satisfy the following multi-cue evaluation.

A stricter safeguard is applied in the presence of a camera-local identity conflict. In this case, recovery is considered only when valid Re-ID information is available and particularly strong geometric and appearance consistency is verified, requiring $d_{ij}^{\text{geom}} \leq 0.30\,\text{m}$ and $d_{ij}^{\text{app}} \leq 0.10$. Therefore, the wider recovery region cannot by itself override a conflicting camera-local identity.

Candidate track-observation pairs within the allowed spatial region for recovery are evaluated using the same general multi-cue formulation introduced for normal association, thus considering geometric, appearance, local-identity, and temporal information. However, the temporal contribution is normalized with respect to the longer reactivation horizon $t_{\text{reactivation}}$, consistently with the longer lifetime of the considered tracks:

$$S_i^{\text{time,rec}} = \text{clip}\left(1 - \frac{a_i}{t_{\text{reactivation}}}, 0{,}1\right). \tag{4.52}$$

The resulting recovery score must satisfy

$$S_{ij}^{\text{rec}} \geq S_{\text{min}}^{\text{rec}}, \tag{4.53}$$

where $S_{\text{min}}^{\text{rec}} = 0.55$; otherwise, the candidate is discarded.

When more than one global track represents a valid recovery candidate for the same unmatched observation, the candidate with the lowest association cost, and therefore the highest recovery score, is selected. Nevertheless, similarly to the post-assignment protection introduced for normal association, the recovery is rejected when the two best hypotheses are too similar. In particular, if

$$S_{\text{best}}^{\text{rec}} - S_{\text{second}}^{\text{rec}} < \Delta S_{\text{amb}}^{\text{rec}}, \tag{4.54}$$

where $\Delta S_{\text{amb}}^{\text{rec}}$ is the recovery ambiguity threshold, the recovery is considered ambiguous and no existing global track is reactivated. Following the same reasoning adopted for normal association, this additional condition prevents the preservation of an old identity when the available evidence does not clearly distinguish it from another potential candidate.

If the recovery succeeds, the global track is reactivated, updated with the state of the considered observation, and it returns active with its original global ID. Hence, the reactivation is particularly useful providing the chance that a temporary loss of a pedestrian does not necessarily imply the creation of a new identity.

As an additional safeguard against short-term fragmentation, the implementation also includes a mechanism to provide a conservative second-chance recovery for recently updated tracks that were not successfully associated during the normal assignment stage. An observation may remain unmatched while a fresh global track is still unassigned. In this case, fresh-track recovery is attempted to provide an additional opportunity to restore the correspondence but under more restrictive conditions. The primary recovery condition relies on strong geometric and appearance consistency, requiring valid Re-ID information, a geometric distance below 0.55 m, and an appearance distance less than 0.12. For non-conflicting camera-local identities, an additional geometry-dominant recovery condition is also considered. In this case, the correspondence can be recovered when the geometric distance is below 0.35 m, provided that the available appearance information is not strongly contradictory and that the geometric association is sufficiently unambiguous with respect to the other fresh global tracks. In the presence of a camera-local identity conflict, the geometry-dominant condition is not allowed. Recovery instead requires particularly strong geometric and appearance consistency, with stricter limits of 0.25 m and 0.08, respectively. This conservative strategy provides an additional protection against short-term identity fragmentation without allowing strong geometric proximity alone to override conflicting identity information.

Finally, when no valid recovery can be established, the creation of a new identity is not performed unconditionally: additional duplicate-prevention safeguards verify that the observation is not still compatible with, too close to, or sufficiently similar to an existing global track. In short terms, the control logic prefers not to risk generating an ID duplication and simply decides to consume the observation without performing a new creation. If the decision was overly conservative, a subsequent observation can provide new evidence for track recovery or for the creation of a new identity. Hence, these checks limit unnecessary global ID proliferation and, together with the recovery mechanisms, contribute to reducing trajectory fragmentation while preserving identity consistency.

4.3.7 Track Lifecycle, Reliability, and Output Publication

The final stage of the implemented fusion node concerns the lifecycle of the global tracks and the publish policy determining whether the internally maintained estimates can be considered sufficiently reliable and recent to be provided as output by the overall perception system. When a new global track is initialized, the observation responsible for its creation represents its first valid hit. However, the track is not immediately considered confirmed: in the implemented configuration, at least two successful observations are required before its state changes to confirmed.

This confirmation mechanism prevents isolated or short-lived observations from being immediately treated as stable pedestrian tracks. Whenever a subsequent observation is successfully associated with the track, its hit counter is incremented by one, and the number of consecutive missed updates is reset. Conversely, when an existing track is not updated during a fusion cycle in which observations from at least two cameras are available, its missed counter is incremented by one. It is important to consider the computational effort induced by three independent tracking pipelines and, consequently, the time they need to process data and produce a result for the upper global layer, not to mention the asynchronism among them. Hence, to avoid penalizing tracks because of short intervals in which insufficient data arrives from the cameras, the system prevents missed updates from being accumulated when fewer than two cameras provide observations during the considered fusion cycle. Tracks are therefore not immediately removed when observations are temporarily unavailable but are progressively moved through the different temporal stages already introduced in the association and recovery logic. In particular, a track can cease to satisfy the publication requirements while still remaining available for normal association or subsequent recovery. This separation between internal track persistence and output visibility is fundamental to preserve an identity during temporary observation gaps without providing outdated pedestrian states to downstream components.

The steps characterizing the temporal evolution of a global track can consequently be summarized through four relevant age thresholds:

- `publish_max_age` defines the maximum age the track can have while still being provided as output;
- `match_max_age` determines for how long the track remains eligible for the normal association process;
- `reactivation_max_age` defines the temporal horizon within which the identity can still be recovered through the stale-track recovery mechanism;
- `delete_max_age` establishes the maximum age after which the global track is permanently removed from the internal memory of the fusion system.

Therefore, global tracks pass through different steps within their lifecycle rather than being immediately deleted as soon as they disappear from the output.

In addition to these temporal conditions, each global track is associated with a reliability value that summarizes the stability and freshness of its current state. In particular, the reliability measure is computed as a heuristic score rather than a

probabilistic estimate of track correctness. It is therefore not intended to represent the probability that the estimated pedestrian state or identity is correct, but instead provides a normalized indicator used by the fusion node as an additional criterion to assess whether a global track is sufficiently consolidated and recent to be exposed as output.

The score is made up of three contributions:

- the number of successful updates accumulated by the track;
- the number of missed updates;
- the current age.

The contribution associated with successful observations is defined as

$$S_i^{\text{hit}} = \min\left(1, \frac{h_i}{H_{\text{norm}}}\right), \tag{4.55}$$

where h_i is the number of hits accumulated by the i-th global track and H_{norm} corresponds to the parameter `reliability_hits_norm`, set to 3. This term progressively increases as the track receives successful updates and saturates at one once the normalization value is reached.

Conversely, the missed-update and age contributions are expressed through two normalized penalties:

$$P_i^{\text{miss}} = \min\left(1, \frac{m_i}{M_{\text{pub}}}\right), \tag{4.56}$$

$$P_i^{\text{age}} = \min\left(1, \frac{a_i}{T_{\text{pub}}}\right), \tag{4.57}$$

where m_i is the number of missed updates, a_i is the current track age, M_{pub} corresponds to the `publish_max_missed` parameter, and T_{pub} to `publish_max_age`. Both penalties increase as the track becomes less recently supported by incoming observations.

The three contributions are then combined into the heuristic reliability score in the following way:

$$R_i = \text{clip}\left(w_h S_i^{\text{hit}} + w_m\left(1 - P_i^{\text{miss}}\right) + w_a\left(1 - P_i^{\text{age}}\right), 0{,}1\right), \tag{4.58}$$

with `reliability_hit_weight` (w_h) = 0.6, `reliability_missed_weight` (w_m) = 0.2, and `reliability_age_weight` (w_a) = 0.2. The resulting value is finally constrained to the interval [0,1]. Therefore, the largest contribution is associated with the number of successful updates, while missed observations and increasing track age reduce the reliability value with the same percentage (20%).

Overall, the publish policy evaluates four points to decide whether a track can be published or not:

- confirmed state;
- missed counter not exceeding the `publish_max_missed` value;
- track age not exceeding the `publish_max_age` threshold;
- reliability not below the `publish_min_reliability` threshold.

These conditions imply that only sufficiently consolidated and recent tracks are exposed by the fusion system and thus can be received by downstream components. This is important to ensure providing safe and trusty information downstream. But, at the same time, maintaining the internal state according to a longer horizon in the track lifecycle is another crucial point enabling the same global ID to become publishable if the individual is detected again.

In addition, before publication, a final conservative duplicate check is applied among the publishable global tracks, preventing two internal tracks that remain strongly compatible in terms of spatial and, when available, appearance information from being simultaneously exposed as two different pedestrians. The surviving tracks constitute the final unified representation generated by the multi-camera perception framework and are published through the `/tracked_people` topic, together with the corresponding visualization and debugging information used for system inspection and experimental validation.

4.4 Socially Aware Navigation Configuration

This section addresses the role of the human tracking information provided by the developed perception system within the socially aware navigation framework.

In particular, the availability of a consistent representation of the surrounding people enables the integration of the social contribution into the local trajectory evaluation. The section then focuses on the controller tuning process carried out in this work to balance the additional social objective with the other navigation requirements and thus obtain an appropriate overall robot behavior.

4.4.1 Integration of Human State Information

The global tracking system developed in this thesis work and previously described provides the navigation system with a unified representation of the pedestrians observed around the robot. In particular, the global tracks generated by the fusion node make pedestrian states available through a common output. This information represents a fundamental input for socially aware navigation, since the local controller must be able to account for the presence of surrounding humans when evaluating the possible local motion strategies and, consequently, the robot motion.

In the navigation architecture, the output of the multi-camera perception system is exploited by the social component integrated into the MPPI controller. In this way, the local planner can evaluate the sampled trajectories also according to the robot's interaction with the surrounding humans, by means of the integrated social critic which introduces a dedicated social contribution into the overall trajectory cost. Consequently, the tracking information establishes the connection between the perception and navigation layers, motivating how the availability of reliable and temporally consistent tracks is essential: the estimated pedestrian states provide the environmental information required by the social critic to influence the selection of the robot trajectory.

The introduction of this additional contribution, however, modifies the balance among the different navigation objectives, initially considered with their default weights by the local controller. Importantly, a trajectory that is preferable from a social perspective must still remain consistent with the planned path, preserve an appropriate direction of motion, and satisfy the geometric and safety constraints imposed by the environment. This consideration justified the need for tuning the local controller, aimed at identifying a configuration in which the social contribution could effectively modify the robot behavior in the presence of pedestrians without compromising the fundamental requirements.

4.4.2 Local Controller Critic Tuning

The proper integration of the social contribution into the MPPI trajectory evaluation required a dedicated tuning process. This process aimed at founding out a robot behavior showing responsiveness when noticing the human presence in its proximity, but at the same time, it was fundamental to obtain an appropriate balance among the different objectives considered by the controller. Since the final trajectory is selected according to the combined contribution of multiple critics, a wrong configuration can potentially compromise the coexistence of the multiple desirable navigation properties, leading to totally wrong robot motions. For this reason, the tuning process was not aimed at maximizing the effect of the social critic independently, but rather at determining a configuration in which socially appropriate reactions to pedestrians could cooperate with stable, safe, and goal-oriented navigation.

The tuning was carried out by progressively adjusting the contribution of the different critics and observing their influence on the trajectories generated by the controller. Particular attention was devoted to those critics regulating the geometric structure of the navigation task. In this context, *PathAlignCritic* played an important role in maintaining the robot motion aligned with the path generated by the global planner. An insufficient contribution of this critic could allow the controller to favor trajectories that reduced the social cost but deviated excessively from the planned route. Conversely, an excessively restrictive path-following behavior could limit the capability of the robot to perform the deviations required to avoid colliding with moving individuals. Hence, the objective was to find the trade-off between path adherence and the necessary freedom for socially motivated trajectory adaptation. A similar role was played by *PreferForwardCritic*, whose contribution was adjusted to favor trajectories consistent with the forward direction of motion. This aspect became particularly relevant when several competing trajectory costs were simultaneously active, since socially advantageous alternatives should not result in unnecessary or undesirable motion directions.

At the same time, the social objective had to remain compatible with the constraints related to physical safety and obstacle avoidance. The configuration of the costmap- and obstacle-related critics was consequently considered together with the social contribution, ensuring that trajectories advantageous from the perspective of human interaction remained feasible with respect to the surrounding environment. In fact, social compliance had to be treated as a plus, rather than as a replacement for the geometric and safety criteria already considered by the controller.

The final configuration was obtained by iteratively performing the tuning process and the subsequent experimental test, balancing these contributions with the objective of avoiding two opposite behaviors. From one side, a too weak social contribution could produce limited modification of the robot trajectory, reducing its social effectiveness. On the other hand, imposing the social cost as the dominating one could produce excessive interference with the regular progression toward the navigation goal. The selected tuning therefore balances all those aspects, allowing the MPPI controller to preserve the overall quality and robustness of the navigation behavior, even with the additional challenge of crowded spaces.

Chapter 5

Experimental Setup

This chapter presents the experimental setup adopted for the implementation and evaluation of the proposed solutions. The considered setup includes both the robotic platform equipped with the physical multi-camera perception system, employed in the experiments conducted in the PIC4SeR laboratory at Politecnico di Torino, and the simulation environment used throughout the development and testing activities.

5.1 Perception Models and Inference

The perception framework employed in this work relies on two pretrained deep learning models performing complementary tasks within each camera-specific tracking pipeline. Human detection and body keypoint estimation are performed using the YOLOv8s-Pose model [44], while the appearance descriptors required for person Re-Identification are extracted through the lightweight OSNet x0.25 network [45]. As described in Chapter 4, these appearance features are exploited by the local StrongSORT trackers and then propagated to the global fusion stage to support cross-camera identity association.

The networks were employed using their pretrained weights, without performing any additional training or fine-tuning as part of this thesis. YOLOv8s-Pose was pretrained on the COCO Keypoints 2017 dataset, which extends the COCO annotations with human body keypoints and is therefore specifically suited for human pose estimation [46]. Similarly, the OSNet x0.25 model was pretrained on the Market-1501 dataset [47], a person Re-Identification benchmark containing over 32,000 annotated pedestrian bounding boxes corresponding to 1,501 identities captured by six different cameras. Both models were executed using the OpenVINO

toolkit, which provides the inference backend adopted for the execution of the deep learning models. The pretrained OpenVINO model representations were directly integrated into the tracking nodes and independently executed for each camera stream. The same model configuration was therefore employed by the three camera-specific pipelines, providing a common detection, pose-estimation, and appearance-feature extraction architecture for the center, left, and right cameras.

5.2 Hardware and Sensors

This section describes the hardware and sensing components employed in the experimental activities. In particular, it presents the Clearpath Jackal mobile platform, details the RGB-D cameras and the implemented multi-camera payload, and describes the Vicon motion capture and the Gazebo simulation environment.

5.2.1 Clearpath Jackal Mobile Platform

The robotic platform involved in this thesis work is the Clearpath Jackal Unmanned Ground Vehicle (UGV). It is a compact four-wheel mobile robot designed for research applications in both indoor and outdoor environments. Its mechanical structure and upper mounting surface allow the integration of additional sensors and custom payloads, making the platform suitable for the exploration of different robotic solutions according to the application and research context. In particular, this flexibility enabled the installation of the multi-camera payload investigated in this work. The Jackal is a differential-drive mobile platform whose motion is controlled through linear and angular velocity commands, providing the mobility characteristics required for autonomous navigation applications. The platform also integrates wheel odometry and an onboard Inertial Measurement Unit (IMU), providing information about the motion of the mobile base and supporting its integration within the robotic navigation framework. In addition, the Jackal integrates an onboard computer that provides the computational interface between the mobile platform and the higher-level robotic software. The computer is connected to the platform microcontroller, which is responsible for low-level functions including motor control, power monitoring, and the acquisition of data from the integrated sensors. This architecture allows higher-level perception and navigation components to interact with the mobile base through the ROS framework [43].

The main mechanical and motion-related specifications are summarized in Table 5.1 [43].

Parameter	Value
Drive configuration	Four-wheel differential drive
External dimensions (L × W × H)	508 × 430 × 250 mm
Weight	17 kg
Maximum payload	20 kg
Maximum translational speed	2.0 m/s
Ground clearance	65 mm
Run time (basic usage)	4 h
User power	5 V at 5 A, 12 V at 10 A, 24 V at 20 A

Table 5.1: Main specifications of the Clearpath Jackal mobile platform.



Figure 5.1: Clearpath Jackal mobile platform employed in the experimental activities at the PIC4SeR laboratory.

For the physical implementation of the proposed perception system, the upper surface of the Jackal was used to accommodate the three RGB-D cameras through a dedicated mounting structure as shown in Figure 5.1. The resulting multi-camera payload and its physical arrangement are described in Section 5.2.3.

5.2.2 Intel RealSense D435 RGB-D Camera

Intel RealSense D435 RGB-D cameras are the sensing devices exploited for the perception task, which simultaneously provide conventional color images and metric depth information, thus the spatial representation of the observed environment.

This sensing technology is a key characteristic in robotics and computer vision applications. Unlike a standard RGB camera, where each image pixel only contains color information, an RGB-D camera associates the visual representation with information describing the distance of the corresponding scene points from the camera. This additional information can be obtained through different depth-sensing technologies, including stereo vision and time-of-flight sensing, and it is generally represented through a depth map, a 2D array. Each pixel of this map encodes the estimated depth of the corresponding observed scene point. For visualization purposes, depth maps can be represented as grayscale or color-coded images, where different intensity or color values correspond to different depth values.



(a) RGB image **(b)** Depth image

Figure 5.2: Example of the RGB and depth streams acquired by the Intel RealSense D435 during the physical experiments: (a) RGB image and (b) corresponding depth image.

Since the RGB and depth streams originate from distinct imaging sensors characterized by different optical centers and intrinsic parameters, a direct pixel-to-pixel correspondence between their native images cannot generally be assumed. Geometric alignment between RGB and depth information is therefore required when a depth measurement has to be associated with a pixel detected in the color image. Such capability of having RGB and depth information geometrically aligned is particularly important in robotic perception applications, especially in this work since it allows information extracted in the image plane to be converted into metric 3D coordinates. In particular, the RGB stream is exploited for person detection,

body pose estimation, local tracking, and appearance feature extraction, while the depth information is used to estimate the 3D position of each detected pedestrian.

Stereo depth estimation

The Intel RealSense D435 estimates depth through a *stereo vision system*, whose operating principle is conceptually similar to human binocular vision. The depth module consists of two infrared cameras, referred to as the *left and right imagers*, which simultaneously observe the same scene from slightly different viewpoints. The optical centers of the two imagers are separated by a fixed distance, known as the *stereo baseline*, which is approximately 50 mm for the D435 [48]. Due to the spatial separation between the two cameras, the same 3D scene point is projected at slightly different horizontal positions in the left and right images. A stereo rectification process is performed to geometrically rectify the two images so that the corresponding scene points can be searched along the same horizontal image line. The displacement between the horizontal coordinates of the two projections is referred to as *disparity*, defined as

$$d = x_L - x_R, \tag{5.1}$$

where x_L and x_R represent the horizontal image coordinates of the same scene point in the left and right infrared images, respectively.

A point located close to the stereo camera produces a relatively large displacement between the two views and therefore a large disparity. Conversely, as the distance of the observed point increases, the two projections become progressively closer and the disparity decreases. This inverse relationship between disparity and distance constitutes the basis of stereo depth estimation. Since the distance between the two infrared imagers and the camera focal length are known from the stereo geometry and calibration, the depth of the observed point can be recovered through triangulation. Under the simplified rectified stereo-camera model, the relationship is expressed as

$$Z = \frac{fB}{d}, \tag{5.2}$$

where Z is the depth of the observed point, f is the focal length of the stereo camera, B is the baseline between the left and right imagers, and d is the measured disparity.

This principle is used by the RealSense depth processing system to generate the depth map from the acquired stereo observations.

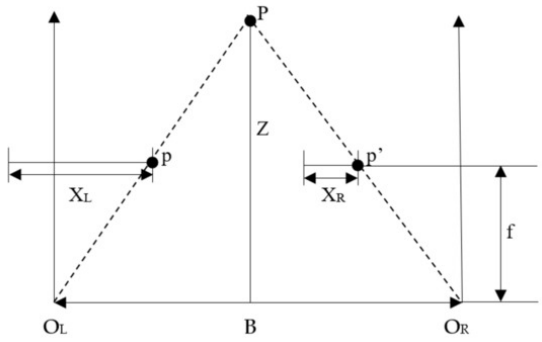


Figure 5.3: Geometric model of binocular stereo vision used for depth estimation through triangulation. Reproduced from [49].

Passive and active stereo vision

The estimation of disparity requires the identification of corresponding points or image regions in the left and right infrared images. This operation, generally referred to as the *stereo correspondence problem*, relies on the presence of sufficiently distinctive visual features that can be recognized in both views. When the observed scene contains sufficient natural texture, it is possible to determine correspondences directly from the images acquired by the two infrared sensors. This operating condition is commonly referred to as passive stereo vision. However, stereo matching becomes more challenging when the observed scene contains poorly textured or approximately uniform surfaces, since similar image regions may not contain sufficient information to determine reliable correspondences. To improve depth estimation under these conditions, the Intel RealSense D435 integrates an infrared projector, which projects a semi-random infrared pattern onto the observed scene. The projected pattern helps the stereo matching process since it introduces artificial texture that can be detected by the two infrared imagers, providing additional distinguishable features for stereo matching. This operating principle is commonly referred to as active stereo vision [50]. The RealSense D435 can therefore exploit both passive and active stereo operating conditions to support stereo correspondence.

It should be noted that the integrated infrared projector does not directly measure the distance to the observed points. Instead, it facilitates the identification of corresponding regions by increasing the amount of texture available to the stereo system, while the depth estimation principle remains based on stereo correspondence, disparity, and triangulation.

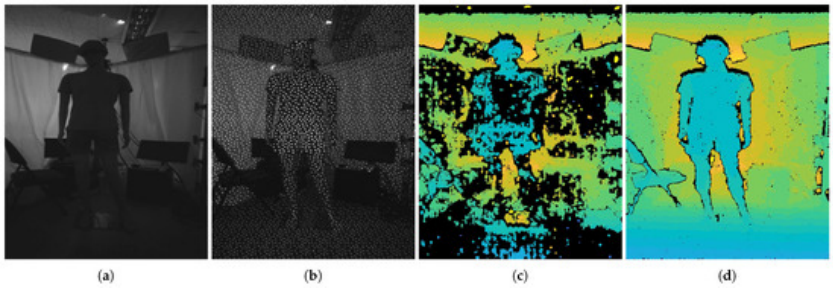


Figure 5.4: Examples of images with and without the projected pattern: (a) infrared image without the projected pattern, (b) infrared image with artificial pattern, (c) depth estimation without the projected pattern, and (d) depth estimation using active stereo. Reproduced from [51].

From depth measurements to 3D coordinates

Once the depth associated with an image point has been estimated, the corresponding 3D position can be reconstructed using the intrinsic parameters of the camera. This operation establishes the relationship between the 2D image coordinates used by the visual perception models and the metric 3D representation required by the tracking framework. According to the pinhole camera model, a 3D point $P = (X, Y, Z)$, expressed in the camera coordinate frame, is projected onto the image plane at pixel coordinates $p = (u, v)$ as

$$\begin{aligned} u &= f_x \frac{X}{Z} + c_x, \\ v &= f_y \frac{Y}{Z} + c_y. \end{aligned} \tag{5.3}$$

where f_x and f_y are the focal lengths expressed in pixel units, while c_x and c_y represent the coordinates of the principal point. These quantities are part of the intrinsic calibration parameters of the camera.

When the depth Z associated with the considered pixel is known, the previous equations can be inverted to recover the corresponding metric coordinates

$$\begin{aligned} X &= \frac{(u - c_x)Z}{f_x}, \\ Y &= \frac{(v - c_y)Z}{f_y}. \end{aligned} \tag{5.4}$$

Therefore, the image coordinates (u, v), together with the corresponding depth value Z, can be mapped to the 3D point $P = (X, Y, Z)$ expressed in the camera reference frame. This inverse mapping is generally referred to as *deprojection*. In the perception algorithms used in this thesis, this operation enables the 2D observations obtained from the RGB image to be converted into metric 3D pedestrian positions. The resulting 3D observations are transformed from the individual camera reference frames to the common reference frame used by the multi-camera tracking system.

Intel RealSense D435 hardware configuration

The Intel RealSense D435 integrates the stereo depth sensing architecture into a compact RGB-D device. Its depth module consists of two infrared imagers and an infrared projector supporting active stereo operation. A dedicated RGB sensor provides the color stream, while the Intel RealSense Vision Processor D4 performs the processing required for stereo depth estimation. The arrangement of the main sensing components is shown in Figure 5.5.

The main technical specifications of the Intel RealSense D435 relevant to the perception system developed in this work are summarized in Table 5.2.

Specification	**Intel RealSense D435**
Depth sensing technology	Active stereo
Depth module	Intel RealSense D430
Stereo imagers	2 × global-shutter infrared imagers
Stereo baseline	50 mm
Depth FOV	87° × 58° × 95° (H × V × D)
RGB FOV	69° × 42° × 77° (H × V × D)
Maximum depth resolution	1280 × 720
Maximum RGB resolution	1920 × 1080
Maximum depth frame rate	90 fps
Vision processor	Intel RealSense Vision Processor D4
Integrated IMU	No

Table 5.2: Technical specifications of the Intel RealSense D435 [48].

Both the RGB and depth streams of the RealSense cameras employed in this work were acquired with a resolution of 640 × 480 pixels at 30 fps . The same acquisition parameters, used in the real experiments, were reproduced in the simulated camera sensors through their URDF configuration. Moreover, the identical acquisition configuration was adopted for the center, left, and right cameras in order to maintain a homogeneous input data across the three camera-specific perception pipelines.

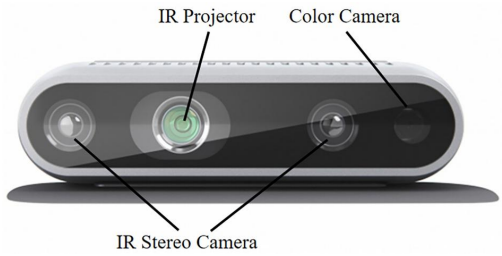


Figure 5.5: Optical components of the Intel RealSense D435, including the infrared stereo cameras, infrared projector, and RGB color camera. [52].

The cameras were integrated into the ROS 2 architecture through the RealSense ROS wrapper, which provides access to the different sensor streams through ROS topics. In particular, each camera provides the RGB image and the depth image, together with the corresponding camera calibration information provided through `sensor_msgs/msg/CameraInfo` messages, which contain the intrinsic parameters required for the deprojection operation previously described. The wrapper also provides the transformations associated with the different camera reference frames through the ROS 2 TF system.

Since the RGB and depth images are acquired by physically distinct sensors, their original image planes are not directly coincident. For this reason, depth-to-color alignment was enabled in the configuration through the RealSense ROS wrapper. This operation maps the depth measurements into the RGB camera coordinate system, allowing a depth value to be associated with the corresponding pixel location in the color image. This functionality is provided through the `align_depth.enable` processing block. The resulting aligned depth stream is published together with its corresponding camera information. This alignment is particularly important for the perception pipeline to obtain the corresponding 3D estimation of the human position.

Camera reference frames and coordinate transformations

The 3D position obtained through the deprojection process is initially expressed in the optical reference frame associated with the corresponding camera. Camera optical frames follow a specific coordinate convention, with the x-axis pointing to the right, the y-axis pointing downward, and the z-axis pointing forward along the optical axis. This convention differs from the standard coordinate system adopted in ROS for conventional robot reference frames. Indeed, according to the ROS convention, the x-axis points forward, the y-axis points to the left, and the z-axis points upward.

The corresponding reference frames are shown in Figure 5.6.

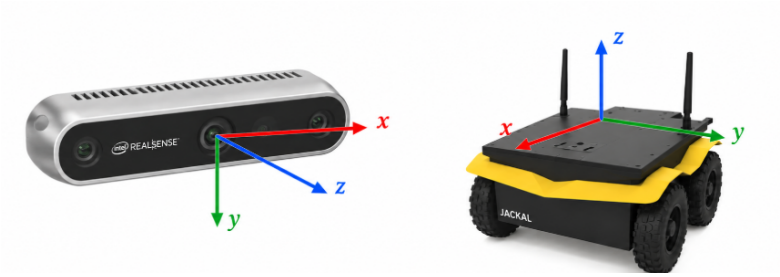


Figure 5.6: Conventions adopted for ROS 2 and optical coordinate frames.

In particular, the Jackal `base_link` frame follows this standard ROS convention. Therefore, the TF hierarchy provides the necessary transformations to correctly relate the optical coordinate frames of the cameras to the reference frame of the mobile platform.

In the implemented multi-camera system, each RealSense D435 is associated with its own reference-frame hierarchy, while the three cameras are rigidly connected to the Jackal according to their respective mounting positions and orientations, as defined during the multi-camera payload design. For this reason, the ROS 2 TF system is employed to enable the transformation of the 3D observations, obtained from the three independent camera viewpoints, through the robot reference-frame structure. Before entering the global fusion stage, the pedestrian positions estimated by each camera-specific perception pipeline are transformed into a common reference frame: the `map` frame. In this way, observations originating from the center, left, and right cameras are represented within the same metric coordinate system and can therefore be geometrically compared and associated by the upper fusion layer, independently of the camera from which they originate. The transformations between the reference frames associated with each camera are published through the ROS 2 `/tf` and `/tf_static` topics. In particular, each RealSense D435 is associated with a base camera frame and with dedicated frames for its sensing components: the RGB sensor, the depth sensor, and the two infrared imagers forming the stereo pair. For each sensing modality, an additional optical frame represents the corresponding measurements according to the optical coordinate convention.

The main reference frames associated with each camera are:

- `camera_link`, representing the base reference frame of the camera;
- `camera_color_frame` and `camera_color_optical_frame`, associated with the RGB sensor; the former follows the ROS convention, while the latter

follows the optical convention;

- `camera_depth_frame` and `camera_depth_optical_frame`, associated with the stereo depth sensor; the former follows the ROS convention, while the latter follows the optical convention.

Additional frame pairs are defined for the two infrared imagers (`infra1` and `infra2`) used by the stereo depth system. These frames are part of the complete RealSense TF hierarchy, although they are not directly employed within the tracking pipeline.

In the implemented system, this reference-frame structure is replicated for the center, left, and right cameras, with dedicated frame names identifying each device. The corresponding frame names adopted for each camera are reported in Table 5.3. In the reported naming pattern, `CAM` is replaced by `center`, `left`, or `right` according to the corresponding camera.

Reference frame	**Frame naming pattern**
Camera base frame	`camera_front_CAM_link`
RGB frame	`camera_front_CAM_color_frame`
RGB optical frame	`camera_front_CAM_color_optical_frame`
Depth frame	`camera_front_CAM_depth_frame`
Depth optical frame	`camera_front_CAM_depth_optical_frame`
Infrared 1 frame	`camera_front_CAM_infra1_frame`
Infrared 1 optical frame	`camera_front_CAM_infra1_optical_frame`
Infrared 2 frame	`camera_front_CAM_infra2_frame`
Infrared 2 optical frame	`camera_front_CAM_infra2_optical_frame`

Table 5.3: Reference-frame naming structure adopted for the center, left, and right Intel RealSense D435 cameras in the implemented ROS 2 configuration.

The three camera hierarchies are connected to the Jackal `base_link` through fixed transformations describing the mounting positions and orientations. Subsequently, the 3D measurements used in the tracking system are transformed to the common `map` frame through the corresponding TF chains. Specifically, the internal transformation hierarchy associated with the left RealSense D435 is shown in Figure 5.7 as a representative example. The `camera_front_left_link` frame acts as the parent frame for the color, depth, and two infrared sensor frames. Each sensor frame is, in turn, connected to its corresponding optical frame. The center and right cameras follow the same reference-frame structure, with the dedicated frame names reported in Table 5.3.

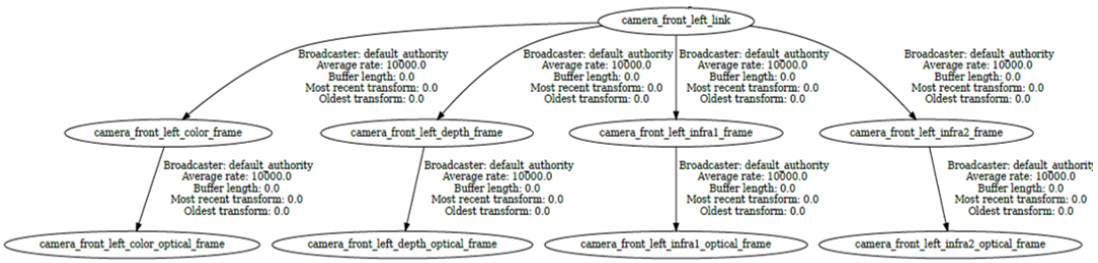


Figure 5.7: Reference-frame hierarchy of the left Intel RealSense D435, generated using the ROS 2 `tf2_tools` package.

5.2.3 Three-Camera Payload Implementation

The multi-camera configuration was physically constructed to be installed on the Clearpath Jackal platform. The objective was achieved by mounting three Intel RealSense D435 cameras on a dedicated structure that was first designed and then manufactured through 3D printing to provide rigid positioning of the cameras. The support was designed as a single component incorporating dedicated mounting seats for the three cameras, arranged and spaced according to the positions and orientations previously determined. The structure also includes dedicated mounting interfaces allowing the complete camera assembly to be securely installed on the supporting tower, which was in turn mounted on the Jackal's front mounting plate.

The three cameras were then positioned: the center camera was installed and oriented along the forward direction of the robot, while the left and right cameras were symmetrically oriented toward the lateral directions, reproducing the configuration established through the geometrical analysis and thus providing the horizontal field-of-view coverage defined during the multi-camera design.

The resulting physical implementation of the three-camera payload on the Jackal platform is shown in Figure 5.8.



Figure 5.8: Physical implementation of the three-camera perception payload on the Clearpath Jackal, showing the 3D-printed mounting structure.

5.2.4 Vicon Motion Capture System

During the physical experiments, the ground-truth information used as reference was acquired using the Vicon Motion Capture system installed in the PIC4SeR laboratory. The system consists of ten high-speed infrared cameras distributed around the experimental area, providing accurate tracking of reflective markers placed within the monitored environment. In particular, the Vicon cameras detect passive reflective markers by illuminating the scene with infrared light and observing the corresponding reflections. After system calibration, observations of the same markers acquired from multiple viewpoints can be combined to reconstruct their 3D positions through triangulation. Multiple markers arranged in a known configuration can then be grouped into a uniquely identifiable rigid object, whose position and orientation can be continuously estimated within the tracking area. Each object can thus be exploited to provide a reliable reference position for the individual wearing it.

Specifically, for pedestrian tracking, dedicated U-shaped wearable targets equipped with reflective markers were employed. The markers were arranged in

rigid and asymmetric configurations, allowing the Vicon system to distinguish the different targets and maintain their identification while the participants moved within the experimental area. In the experiments considered in this work, the wearable targets were identified as `U_1`, `U_2`, and `U_3`, providing an independent reference identity for each tracked pedestrian.



Figure 5.9: Vicon Motion Capture setup.



Figure 5.10: Wearable U-shaped reflective-marker targets (`U_1`, `U_2`, and `U_3`) employed as pedestrian ground-truth references during the physical experiments.

5.3 ROS 2 Experimental Architecture

The complete perception framework is defined within ROS 2 Humble, integrating the existing components with the newly developed ones according to the modular architecture introduced in Chapter 4. ROS 2 provides the communication infrastructure connecting the different computational processes, referred to as nodes, through dedicated communication channels called topics. In particular, in the implemented system, this communication infrastructure connects the three RGB-D cameras, the independent local tracking pipelines, and the global fusion layer. The resulting network of nodes and topics constitutes the ROS 2 computational graph, which represents the interactions and data flow between the different components of the system. The implemented architecture allows the individual processing stages to operate as independent nodes while exchanging sensor measurements and tracking information through dedicated ROS 2 topics.

5.3.1 Multi-Camera Data Flow and Topic Structure

The overall system proposed in this thesis is organized into three parallel processing branches, one for each Intel RealSense D435 camera. The center, left, and right cameras independently provide the RGB, aligned depth, and camera calibration information. Each branch processes the data provided by its own camera independently, while adopting the same detection and tracking configuration described in the previous sections. The global fusion node subscribes to the three local tracking topics (`/people_center`, `/people_left`, and `/people_right`, respectively) to process the acquired data as an upper layer. The observations received from the three camera-specific trackers are combined to generate a unified representation of the pedestrians surrounding the robot. The resulting global tracks are published on the `/tracked_people` topic and represent the final output of the multi-camera perception system provided to downstream components of the robotic architecture.

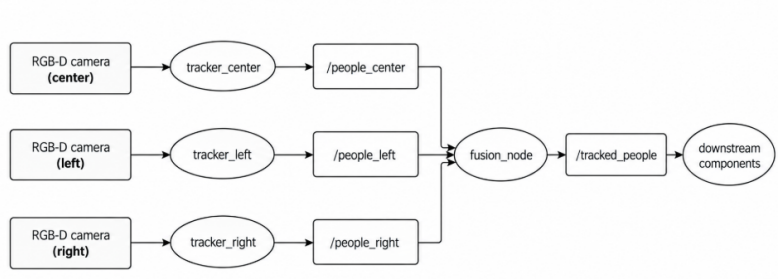


Figure 5.11: Block diagram of the ROS 2 multi-camera perception architecture.

Moreover, during the experimental validation, additional ROS 2 topics are included to collect ground-truth pedestrian information providing the reference positions. For the experiments conducted in the PIC4SeR laboratory, the ground truth was provided by the Vicon Motion Capture system, whose data stream is published through the `/vicon/people` topic. Conversely, during the experiments performed in simulation, ground-truth information was obtained from the `/human_states` topic, which provides the reference positions of 14 agents generated within the simulated environment. In both cases, the ground-truth information is not employed by either the local tracking pipelines or the global fusion node. Instead, it is recorded together with the perception outputs in the corresponding ROS bags and subsequently exploited as an independent reference for the quantitative evaluation of the tracking system.

5.3.2 ROS Bag Data Acquisition

The dataset exploited during the experimental phase was constructed through the recording of ROS 2 bags. Indeed, these recordings were used to collect the data generated during both the simulated and physical experiments, allowing the complete data streams to be stored and subsequently reproduced. This acquisition mechanism was particularly useful for the perception experiments, since the same sensor observations could be replayed multiple times while preserving the temporal relationships between the recorded messages.

During the physical experiments in the PIC4SeR laboratory, the recorded information included the raw data from the sensing units of the three cameras, the robot odometry, the TF information required by the perception pipeline, and the ground-truth data provided by the Vicon Motion Capture system, all collected within the same experimental acquisition.

An analogous data-acquisition procedure was adopted for the experiments performed in simulation. In this case, the main difference in the recorded data streams was related to the ground truth, since the reference positions were collected through the topic providing the states of the generated agents during simulation (`/human_states`). These experimental datasets were subsequently employed for the quantitative and qualitative evaluation of the tracking system, as described in the following chapter.

5.4 Simulation Environment for Social Navigation Validation

Following the development and validation of the multi-camera perception framework, an additional experimental stage was devoted to evaluating the integration of the acquired perception information within the navigation system. Indeed, the outcome of the perception system is made available to downstream components through the `/tracked_people` topic, to which the controller server subscribes to collect the people messages. In this way, the integrated social critic, acting as the social component of the local planner, can correctly account for the presence of surrounding people during robot navigation.

To evaluate both the correct integration of the perceived pedestrian information and its effect on the resulting socially aware navigation behavior under controlled and repeatable conditions, the corresponding experiments were conducted in simulation using Gazebo and RViz.

Gazebo is a robotics simulation environment that allows robots, sensors, agents, and surrounding environments to be reproduced within a virtual scenario while simulating their physical interactions. In this work, it was employed to reproduce an indoor environment, the Clearpath Jackal robot, its onboard sensing configuration, and the pedestrian agents required to generate dynamic human–robot interaction scenarios. The simulated robotic platform was equipped with the same three-camera perception configuration adopted for the physical system, allowing the complete perception and navigation architecture to cooperate within the virtual environment.

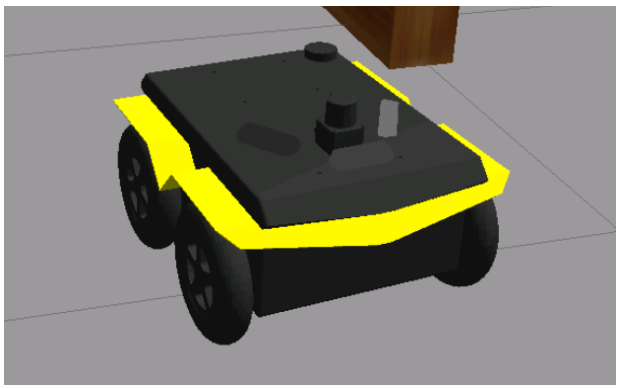


Figure 5.12: Clearpath Jackal equipped with the simulated three-camera perception configuration in Gazebo.

RViz, on the other hand, is a visualization tool provided within the ROS ecosystem for displaying and inspecting data exchanged by the different components of a robotic system. During the simulated experiments, it was employed to monitor the evolution of the navigation task and to visualize relevant information such as the robot state, planned navigation paths and candidate trajectories, costmap information, pedestrian-related data, camera data streams, and perception results.

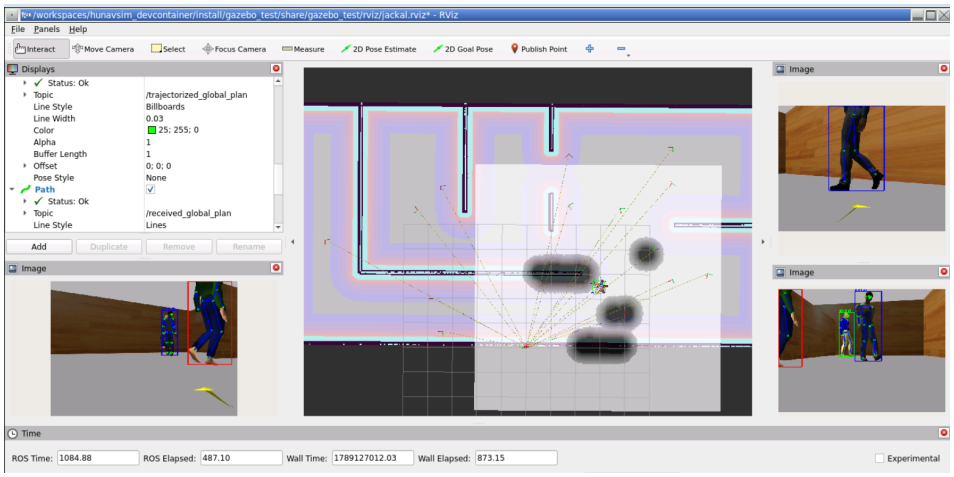


Figure 5.13: RViz visualization of the simulated social navigation scenario.

The complementary use of Gazebo and RViz therefore provided a controlled experimental environment in which the overall robot navigation behavior, with the human-perception results integrated into the navigation architecture, could be tested and visually monitored.

Chapter 6

Experimental Validation and Results

This chapter presents the experimental validation of the multi-camera perception system developed in this work and its employment within the social navigation framework. The proposed approaches are evaluated under progressively harder simulated and real-world conditions, using quantitative metrics to assess tracking performance and identity consistency.

6.1 Evaluation Methodology and Metrics

The performance of the proposed multi-camera perception system was assessed by comparing the final outcome generated by the Fusion Node with the corresponding ground-truth trajectories. A common evaluation methodology was designed and adopted for both simulated and real-world experiments, enabling a consistent evaluation under operating conditions of increasing difficulty.

Since the main objective of the global fusion stage is to provide the downstream components with robust human perception through the correct management of local people detections and the preservation of pedestrian identities over time, the adopted evaluation methodology addresses both tracking accuracy and temporal identity consistency. Ground-truth data are therefore used to determine the intervals in which each person can be meaningfully evaluated by the perception system and to define the corresponding identity-continuity episodes. Within these intervals, the global tracking output is assessed through fusion-specific identity metrics and standard multi-object tracking metrics.

6.1.1 Observability and Identity-Continuity Episodes

The identity consistency evaluation requires distinguishing between periods in which a person is actually observable by the multi-camera perception system and periods in which maintaining the same identity is no longer expected from the Fusion Node, according to the developed logic. For this reason, the evaluation is not performed over the entire duration of each recording as a single continuous interval but is organized into visibility episodes defined independently of the global IDs produced by the system.

Geometric observability is first reconstructed from the pose of the robot and the ground-truth position of each person in the map frame. The robot pose is interpolated at the ground-truth timestamps and used together with the known position and orientation, horizontal field of view, and operating range of the three RGB-D cameras. These pieces of information therefore enable the identification of the inspectable region at each time step. In particular, a person is considered geometrically visible whenever the corresponding ground-truth position lies within the sensing region of at least one camera.

Continuous geometrically visible samples are grouped into visibility segments. Consecutive segments separated by short temporary losses of observability are kept within the same identity-continuity episode, since the Fusion Node is explicitly designed to maintain or recover the identity of a person after short interruptions. Conversely, an observability loss exceeding the recovery horizon of the fusion system defines the beginning of a new episode. The observability loss is not related only to the person exiting the actual sensing area of the multi-camera system, but also includes the lack of inputs from the local pipelines. Indeed, the same criterion is applied when no incoming observations are available for a while. In fact, if the interval between two consecutive valid local observations of the same person exceeds the reactivation horizon, the next observation belongs to a new identity-continuity episode, since the previous global track is no longer recoverable by design.

This distinction is fundamental because it takes into account that geometric visibility does not necessarily imply the availability of a valid local detection. A person may lie within the theoretical field of view while temporarily not being detected or tracked by any local camera pipeline. Consequently, the fusion strategy may not receive useful information to be associated with an existing global track. The definition of evaluation intervals makes it possible to effectively measure the consistency of the global layer's output. Within each visibility episode, the Fusion Node is therefore expected to maintain a consistent global ID, including across normal transitions between cameras and short observation gaps. A change of global ID within the same episode represents an identity-continuity failure, whereas

assigning a new global ID after the beginning of a new episode is not considered an identity switch.

6.1.2 Fusion-Specific Metrics

To specifically evaluate the contribution of the global fusion stage, a set of identity-oriented metrics was computed over the defined visibility episodes. The availability of valid local observations makes the episode fusion-evaluable, since in those instants the Fusion Node effectively receives an input that can be represented by a global track. This correctly separates the performance evaluation of the fusion algorithm from missed observations occurring in the upstream local perception pipelines.

For each episode, four temporal quantities are considered: the geometrically visible duration T_{vis}, the fusion-evaluable duration T_{eval}, the duration for which a global ID is successfully associated T_{ass}, and the duration T_{dom} associated with the dominant global ID, i.e. the global ID maintained for the largest fraction of the evaluable episode. These quantities allow different aspects of the perception and fusion pipeline to be distinguished.

The local observation fraction measures how much of the geometrically visible interval is actually covered by valid local observations:

$$F_{\text{local}} = \frac{T_{\text{eval}}}{T_{\text{vis}}}. \tag{6.1}$$

This metric is therefore mainly useful for identifying missed observations occurring in the local perception pipelines, rather than failures of the global tracking stage.

The global association coverage instead measures the fraction of the fusion-evaluable duration for which a global ID is successfully assigned:

$$C_{\text{global}} = \frac{T_{\text{ass}}}{T_{\text{eval}}}. \tag{6.2}$$

The Global-ID purity evaluates identity consistency only during the intervals in which a global ID is available. It is defined as

$$P_{\text{ID}} = \frac{T_{\text{dom}}}{T_{\text{ass}}}. \tag{6.3}$$

A high purity therefore indicates that the same global ID dominates the assigned portion of the episode, whereas a lower value reveals the presence of competing global IDs. Finally, the Global-ID stability combines association availability and identity consistency by measuring the fraction of the complete fusion-evaluable duration covered by the dominant identity:

$$S_{\mathrm{ID}} = \frac{T_{\mathrm{dom}}}{T_{\mathrm{eval}}}. \tag{6.4}$$

Consequently,

$$S_{\mathrm{ID}} = C_{\mathrm{global}}\, P_{\mathrm{ID}}, \tag{6.5}$$

so that high stability requires both continuous global-ID assignment and preservation of the same identity.

In addition to these duration-based measures, discrete identity events are recorded. A global-ID switch occurs when the global ID associated with the same person changes within the same identity-continuity episode. Temporary losses of the global association are counted as association interruptions. Together, these event-based indicators complement the temporal metrics by indicating whether a reduction in stability originates from identity replacement or from temporary absence of a valid global association.

6.1.3 Standard Multi-Object Tracking Metrics

The validation was not confined to the fusion-specific metrics, as the global tracking output was also evaluated using standard Multi-Object Tracking (MOT) metrics. These metrics provide a complementary assessment of the overall tracking performance, answering different questions about detection accuracy, temporal association quality, and identity consistency with respect to the ground-truth trajectories.

The Higher Order Tracking Accuracy (HOTA) metric jointly evaluates the detection and association components of a multi-object tracking system. For a given matching threshold α, HOTA is defined as

$$\mathrm{HOTA}_{\alpha} = \sqrt{\mathrm{DetA}_{\alpha}\mathrm{AssA}_{\alpha}}. \tag{6.6}$$

where DetA_{α} represents the detection accuracy and AssA_{α} the association

accuracy.

Detection accuracy is defined as

$$\mathrm{DetA}_\alpha = \frac{|TP_\alpha|}{|TP_\alpha| + |FN_\alpha| + |FP_\alpha|}. \tag{6.7}$$

where TP_α, FN_α, and FP_α denote the sets of true-positive matches, false negatives, and false positives obtained at the considered matching threshold.

Association accuracy instead evaluates whether correctly detected targets remain consistently associated with the same identities over time. For each true-positive match c, its association quality can be expressed as

$$A(c) = \frac{|TPA(c)|}{|TPA(c)| + |FNA(c)| + |FPA(c)|}. \tag{6.8}$$

where $TPA(c)$ represents the correctly associated detections belonging to the corresponding ground-truth and predicted trajectories, while $FNA(c)$ and $FPA(c)$ represent missed and incorrect associations, respectively. The overall association accuracy is then

$$\mathrm{AssA}_\alpha = \frac{1}{|TP_\alpha|} \sum_{c \in TP_\alpha} A(c). \tag{6.9}$$

The final HOTA score is obtained by averaging the HOTA values over the considered matching thresholds. The separation between DetA and AssA is particularly useful in this work, since it makes it possible to distinguish errors related to the availability of the global tracks from errors related to their temporal identity consistency.

Multiple Object Tracking Accuracy (MOTA) provides an additional measure of overall tracking performance by combining missed targets, false detections, and identity switches. It is defined as

$$\mathrm{MOTA} = 1 - \frac{FN + FP + IDSW}{GT}. \tag{6.10}$$

where FN is the number of false negatives, FP the number of false positives, $IDSW$ the number of identity switches, and GT the total number of ground-truth

target instances considered during the evaluation. A higher MOTA therefore indicates a lower overall incidence of detection and identity-association errors.

Finally, trajectory-level identity consistency is evaluated through ID Precision (IDP), ID Recall (IDR), and the corresponding ID F1-score (IDF1). These metrics are based on the numbers of identity true positives ($IDTP$), identity false positives ($IDFP$), and identity false negatives ($IDFN$):

$$\text{IDP} = \frac{IDTP}{IDTP + IDFP}. \tag{6.11}$$

$$\text{IDR} = \frac{IDTP}{IDTP + IDFN}. \tag{6.12}$$

$$\text{IDF1} = \frac{2\,IDTP}{2\,IDTP + IDFP + IDFN}. \tag{6.13}$$

IDP measures the correctness of the identities assigned by the tracker, whereas IDR measures the extent to which the ground-truth identities are correctly recovered. IDF1 combines both aspects into a single score and therefore provides an overall indication of identity preservation along the evaluated trajectories.

These trajectory-level identity metrics thus provide a standardized characterization of identity tracking performance.

6.2 Dataset

The experimental validation of the multi-camera perception system was conducted using datasets acquired both in simulated and real-world environments. A total of six ROS 2 bags were considered, comprising three experiments performed in the Gazebo simulation environment and three experiments carried out with the physical robotic platform in the PIC4SeR laboratory. The two datasets were employed to evaluate the proposed perception framework under different operating conditions and tracking scenarios, involving different numbers of agents.

6.2.1 Simulation Dataset

The simulation dataset was acquired while moving the robot through the simulated indoor environment in Gazebo. The robot was spawned together with the three-camera model, and the pedestrians were simulated using HuNavSim, a ROS 2 framework for simulating human agents and their motion in robotic environments. The experiments included a total population of 14 HuNav agents distributed across different regions of the simulated scenario. Different skin configurations were assigned to the agents in order to introduce visual appearance variability among the simulated pedestrians, providing distinguishable visual characteristics for the Re-ID stage of the local tracking pipeline.

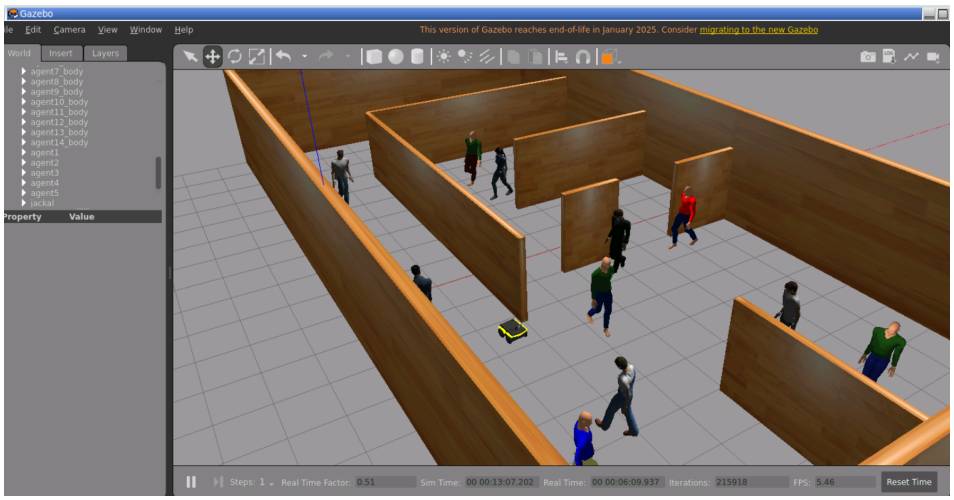


Figure 6.1: Gazebo simulation environment populated with HuNav agents.

The dataset consists of three ROS 2 bag sequences containing the synchronized information required to reproduce the multi-camera perception pipeline and perform the tracking evaluation, as described in Section 5.3.2.

The recorded bags capture different levels of tracking complexity in terms of the extent of the region explored by the robot and, consequently, the number of observable people. Specifically:

- In the first experiment, the Jackal moved through a region in which three moving agents were observed.
- The second experiment included four observed agents, with three moving pedestrians and one stationary pedestrian, while the robot moved through the environment.

- The third experiment covered a larger portion of the simulated environment. During the 152 s sequence, the robot explored multiple regions and encountered nine different agents.

6.2.2 Real-World Dataset

The real-world dataset was acquired in the PIC4SeR laboratory using the physical Jackal platform, upon which the three-camera payload was mounted. The dataset consists of three ROS 2 bags with an average duration of approximately one minute per recording, including synchronized data such as the raw RGB-D streams from the three Intel RealSense D435 cameras and the odometry information, as presented in Section 5.3.2. Ground-truth information was acquired using the Vicon Motion Capture system available in the laboratory, as already explained. Three participants took part in the experiments, identified as U1, U2, and U3. In particular, each participant wore a dedicated U-shaped frame equipped with reflective markers arranged in a distinct asymmetric configuration, allowing the motion-capture system to uniquely identify and track the corresponding person, as illustrated in Figure 6.2. Reflective markers were also positioned on the Jackal mounting surface to enable its pose to be tracked, as well.



Figure 6.2: Participants wearing the U-shaped reflective-marker targets.

The three recorded bags represent different levels of experimental difficulty in terms of robot motion and relative motion between the platform and the participants.

Specifically:

- In the first experiment, the Jackal remained stationary while the three participants moved around the robot. In this scenario, changes in the observations were due to pedestrian motion.
- In the second experiment, the Jackal was moved slightly within the laboratory, while the three participants continued moving around it. This introduced simultaneous robot and pedestrian motion and continuous changes in the relative viewpoints of the three cameras.
- In the third experiment, the Jackal was repeatedly and rapidly moved back and forth throughout the laboratory. The robot thus continuously changed its direction of motion, while the three participants remained in motion, resulting in frequent changes in the relative configuration between the robot and the participants.

6.3 Simulation Results

First experiment.
In the first experiment, three moving agents were observed during the sequence. Figure 6.3 shows the temporal global-ID assignment for each ground-truth agent.

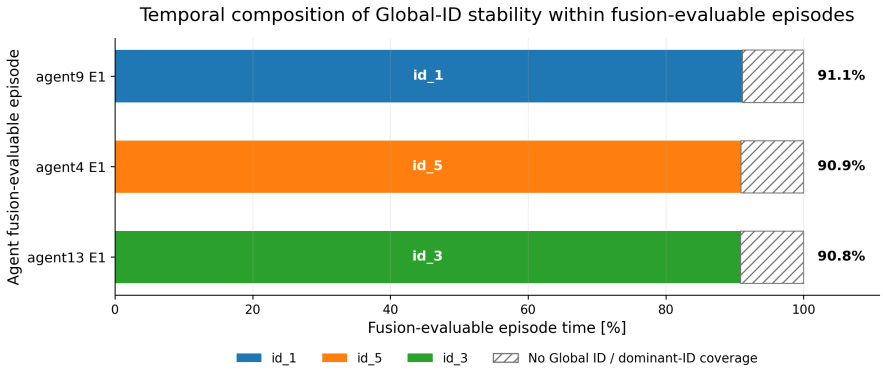


Figure 6.3: Temporal global-ID assignment for the first simulation experiment.

As shown in Figure 6.3, the global tracking outcome is able to provide a unified representation of the humans observed in the scene, since each ground-truth agent is associated with a distinct global ID, which remains consistent throughout the corresponding episode. Specifically, agent9 is associated with `id_1`, agent4 with

`id_5`, and agent13 with `id_3`. No global-ID switch is observed for any of the three agents.

The fusion-specific results are summarized in Table 6.1.

Metric	**Result**
Global association coverage	90.97%
Global-ID purity	100.00%
Global-ID stability	90.97%
Observed global-ID switches	0
Global association interruptions	16

Table 6.1: Fusion-specific metrics for the first simulation experiment.

The percentages reported in Table 6.1 confirm the identity-consistency behaviour observed in Figure 6.3. All three ground-truth agents are geometrically observable during the experiment and generate a single fusion-evaluable identity-continuity episode each (E1), resulting in three evaluable episodes overall. Therefore, during the experiment, no observation loss is sufficiently long to close a visibility episode and start a new one. All three evaluable episodes are successfully represented by a global ID. A Global-ID purity of 100% is obtained, confirming that no competing global identities are assigned to the same ground-truth agent within its identity-continuity episode. The global association coverage reaches 90.97%, indicating that a valid global ID is available for most of the fusion-evaluable duration. The difference from perfect stability is due to small percentages of global association losses, during which no identity replacements occur. These losses correspond to intervals in which local observations are available, but no global ID is assigned.

The evaluation of this first experiment is completed by the standard MOT metrics reported in Figure 6.4. The results confirm the overall consistency of the global tracking output. The two components characterizing the HOTA score present high values, with a slightly better result for the association component (AssA). This indicates that the system is particularly effective in maintaining temporal associations once the global tracks are established. The tracking accuracy is also supported by the relatively strong MOTA value, while the high identity-related scores further support the identity-consistency results obtained from the fusion-specific evaluation. Moreover, the standard MOT evaluation reports a total of 16 trajectory fragmentations across the tracks of all three observed agents. These fragmentations refer to temporary interruptions in the association between the ground-truth trajectories and the global tracking output and are consistent with the temporary losses of global association previously mentioned. Nevertheless,

identity consistency is not affected, since the system maintains the same global identities. This behavior is reflected by the absence of identity switches over the entire experiment.

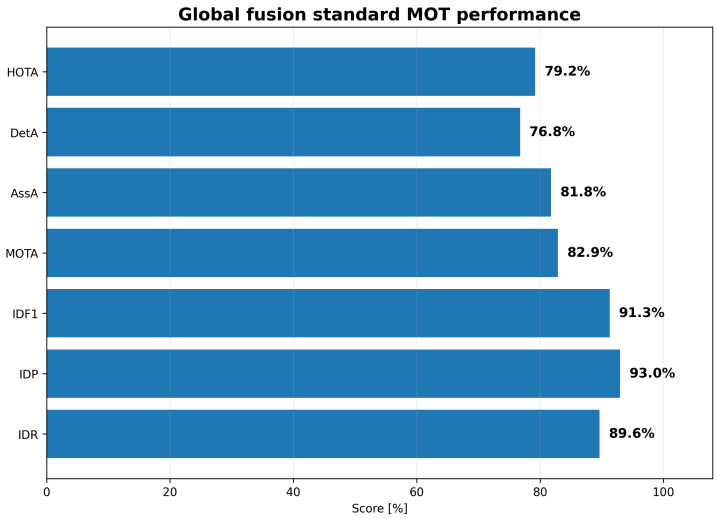


Figure 6.4: Standard Multi-Object Tracking performance metrics for the first simulation experiment.

Second experiment.
During the second experiment, while moving through the simulated environment, the robot detects four agents, three of which are moving while one remains stationary. Figure 6.5 reports the temporal global-ID assignment over the visibility episodes. The obtained results show again a good performance of the system, consistently with the behavior observed in the previous case. Indeed, each observed agent is associated with a unique global ID, which remains distinct from the others and consistent throughout the experiment. Specifically, agent7 is associated with `id_4`, agent9 with `id_5`, agent11 with `id_2`, and agent14 with `id_1`. No global-ID replacement is observed. The grey intervals correspond to periods in which a valid local fusion input is available, but no global ID is assigned, whereas the hatched intervals indicate temporary losses of observability.

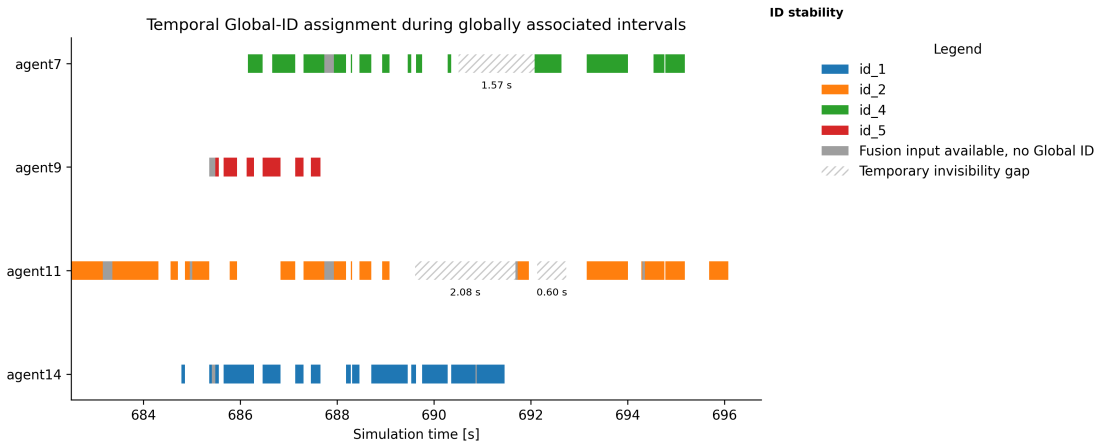


Figure 6.5: Temporal global-ID assignment across the visibility episodes of the four agents in the second simulation experiment.

In this experiment, multiple temporary observability gaps occur without causing an identity change. Agent7 becomes temporarily unobservable for approximately 1.57 s, while agent11 experiences two visibility losses of approximately 2.08 s and 0.60 s, respectively. In all cases, the same global ID assigned before the interruption is recovered when the agent becomes geometrically observable again. This confirms the capability of the global fusion stage to maintain identity continuity across short periods in which the person leaves the sensing region of the multi-camera system.

Figure 6.6 provides a complementary representation of the same tracking behavior with respect to the cumulative ground-truth path of each agent. The figure illustrates the assigned global IDs with different colors and distinguishes among periods in which local observations are available but no global association is produced, periods in which the agent is geometrically observable but no valid local observation is provided to the Fusion Node, and periods in which the agent lies outside the geometrically visible region of the multi-camera system. This representation shows that temporary observation interruptions may occur at different stages of the perception process. In particular, dashed segments identify losses of local observation while the corresponding agent is still geometrically observable, whereas dotted segments indicate actual exits from the multi-camera geometric visibility region. In both cases, when a valid observation becomes available again, the Fusion Node is able to recover the previously established global ID without generating a new identity. The continuity of the same color across the interrupted portions of the trajectories provides a direct visualization of this behavior. The cumulative-path plot is representative also because it clearly shows the stationary condition of agent11, since its ground-truth travelled distance remains approximately constant

throughout the sequence. Despite the absence of significant pedestrian motion, the agent undergoes some observation interruptions as the robot moves and the relative configuration between the pedestrian and the three camera fields of view changes. Nevertheless, `id_2` is consistently retained after the interruptions.

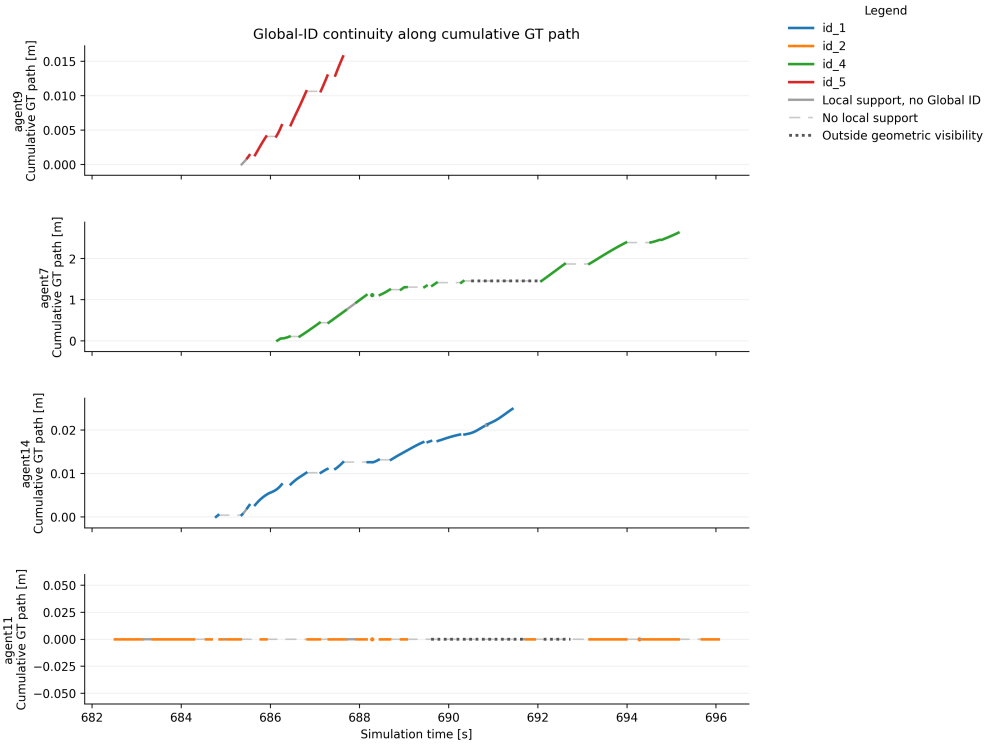


Figure 6.6: Cumulative ground-truth paths and global-ID assignments for the second simulation experiment.

Each of the four geometrically visible agents generates a single fusion-evaluable identity-continuity episode, resulting in four evaluable episodes overall. The fusion-specific metrics obtained for the second experiment are reported in Table 6.2.

Metric	**Result**
Global association coverage	94.13%
Global-ID purity	100.00%
Global-ID stability	94.13%
Observed global-ID switches	0
Global association interruptions	8

Table 6.2: Fusion-specific metrics for the second simulation experiment.

Therefore, the temporary losses of geometric visibility and local observation occurring during the sequence remain within the recovery horizon of the Fusion Node and do not cause the generation of new independent identity-continuity episodes. A global association coverage of 94.13% is obtained, while the Global-ID purity reaches 100%. No global-ID switches are observed, confirming that each agent maintains a unique global identity within its corresponding episode. As illustrated in the previous figures, the eight global association interruptions observed across the four agents during the experiment do not prevent the Fusion Node from recovering the previously established identity when a valid observation is available again.

The evaluation is completed by the standard MOT metrics, whose main results are summarized in Figure 6.7. The results show high overall tracking performance, with a balanced contribution of detection and association accuracy to the HOTA score. In particular, the slightly higher AssA compared with DetA indicates again a higher reliability of the temporal associations once the global tracks are established. The identity-related metrics are particularly high, consistent with the fusion-specific analysis and with the absence of global-ID replacements observed during the entire experiment. No identity switches are reported by the standard MOT evaluation. A total of eight trajectory fragmentations are observed across the tracks of the four agents, indicating temporary interruptions in the correspondence between the ground-truth trajectories and the global tracking output. The correspondence is subsequently recovered by the system while maintaining the same identity.

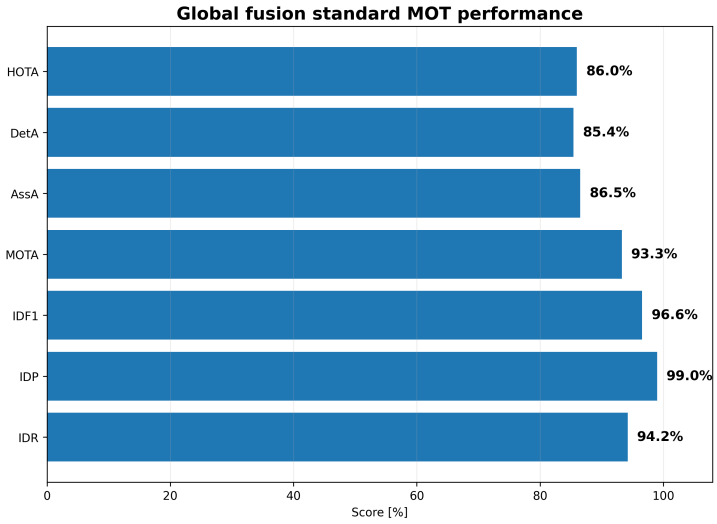


Figure 6.7: Standard Multi-Object Tracking performance metrics for the second simulation experiment.

Third experiment.
The third simulated experiment represents the most extensive and challenging scenario considered in the simulation dataset. It lasts longer and, throughout the experiment, the Jackal explores different areas of the simulated indoor environment, where it encounters many more agents. Compared with the previous experiments, the larger explored area produces more heterogeneous observation conditions, with several pedestrians entering, leaving, and re-entering the observable region of the multi-camera system.

Figure 6.8 shows the temporal global-ID assignment obtained for the nine observed agents. In this experiment, several agents are observed across multiple independent identity-continuity episodes. Indeed, the extended motion of the robot produces longer periods in which previously observed pedestrians leave the geometrically visible region or valid observations are no longer provided by the local tracking pipelines. In particular,the absence of valid observational support for a time interval longer than the recovery horizon closes the current visibility episode and makes a new one begin.

The temporal representation provided by Figure 6.8 clearly illustrates the resulting fragmentation of the observation intervals. Specifically, just one agent is evaluated within a single episode, while all the other pedestrians are observed again after longer interruptions, resulting in two or more independent episodes. Changes in global ID occur only after the beginning of a new identity-continuity episode, when the previous global track can no longer be expected to remain recoverable. Such changes therefore do not represent identity switches within a continuous tracking interval, but rather the initialization of a new global identity when the person is observed again. The prolonged time intervals separating two successive visibility episodes therefore make the assignment of a new global identity reasonable, since the encountered agent can plausibly be considered as a newly detected person. Within the individual identity-continuity episodes, instead, the global-ID assignment remains largely stable despite the more fragmented observations and the frequent changes in the relative configuration between the robot and the pedestrians.

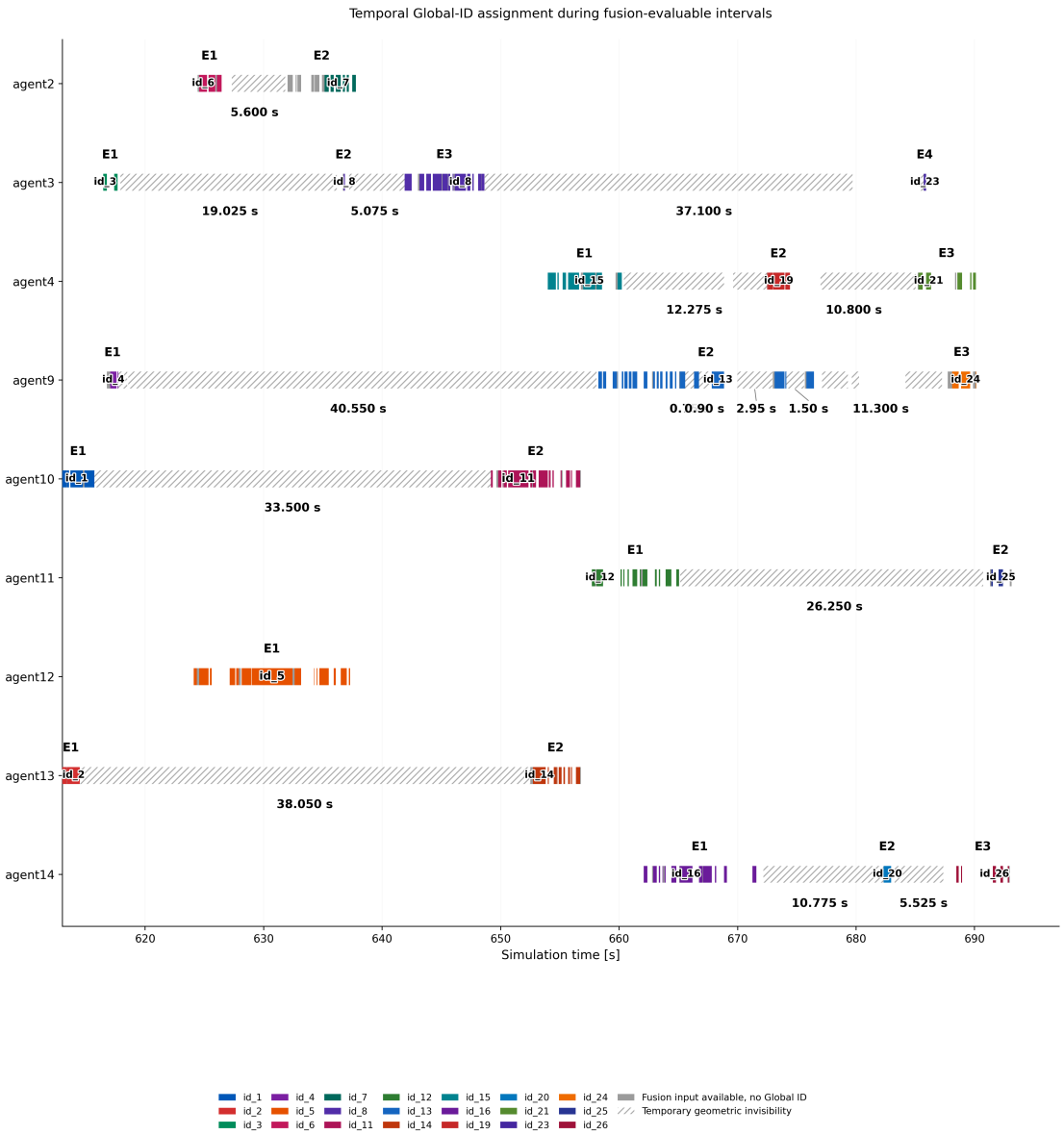


Figure 6.8: Temporal global-ID assignment across the visibility episodes in the third simulation experiment.

Overall, the results therefore indicate that the Fusion Node is able to preserve temporal identity consistency under substantially more complex observation conditions, while allowing a new global identity to be established when the previous track has exceeded its defined recovery horizon. This behavior of the system is further supported by the fusion-specific metrics summarized in Table 6.3.

Metric	Result
Fusion-evaluable agents	9
Fusion-evaluable episodes	22
Global association coverage	88.68%
Global-ID purity	100.00%
Global-ID stability	88.68%
Global-ID switches	0
Global association interruptions	36

Table 6.3: Fusion-specific metrics for the third simulation experiment.

The fusion-specific metrics reported in Table 6.3 quantitatively confirm the identity-consistency behaviour observed in Figure 6.8. The nine fusion-evaluable agents generate a total of 22 independent identity-continuity episodes, reflecting the observation conditions described above. Despite this increased fragmentation, a Global-ID purity of 100% is obtained and no global-ID switches are observed, confirming that identity consistency is preserved within the individual identity-continuity episodes. As a direct consequence of the long time intervals with observation interruptions, the global association coverage reaches a lower percentage.

To complete the tracking performance assessment, the standard MOT metrics are considered. Despite the increased complexity of the scenario, the results indicate a satisfactory overall tracking performance. In particular, the association component remains stronger than the detection component, still suggesting the system's effectiveness in preserving temporal correspondences. The identity-related metrics also remain high, consistent with the identity stability observed through the fusion-specific evaluation. The lower detection performance reflects the more challenging observation conditions encountered during the longer sequence, in which a larger number of agents are observed while the robot moves across different regions of the environment.

To provide an overall comparison of the tracking performance across the simulated scenarios, Table 6.4 summarizes the standard MOT metrics obtained in the three experiments.

Metric	Experiment 1	Experiment 2	Experiment 3
HOTA	79.21%	85.96%	76.67%
DetA	76.76%	85.41%	72.75%
AssA	81.75%	86.52%	80.84%
MOTA	82.88%	93.27%	80.34%
IDF1	91.28%	96.55%	89.90%
IDP	92.99%	98.99%	89.77%
IDR	89.64%	94.23%	90.03%

Table 6.4: Standard MOT metrics for the three simulation experiments.

The comparison highlights the effect of the different experimental conditions on the tracking performance. The second experiment achieves the highest values across the considered metrics, whereas the third experiment presents the lowest HOTA and detection-related performance in correspondence with the more challenging and fragmented observation conditions. Nevertheless, the association and identity-related metrics remain comparatively high also in the third experiment, confirming the ability of the global tracking layer to preserve temporal identity consistency even under the most complex simulated scenario.

6.4 Real-World Results

First experiment.
The first real-world experiment evaluates the multi-camera perception system under conditions in which the Jackal remains stationary. The variations in the observations acquired by the three cameras are thus primarily determined by pedestrian motion and their transitions across the camera fields of view, since the sensing platform retains a fixed position. The experiment therefore provides a first assessment of the proposed perception framework under real sensing conditions, including temporary occlusions, missing local observations, and RGB-D measurement uncertainty.

The temporal global-ID assignment for the three participants is illustrated in Figure 6.9. Two global-ID switches can be observed for two participants, U1 and U3, respectively, both occurring during the initial portion of their tracking histories. In both cases, the previously established global track is not successfully recovered following a short interruption. The association is rejected by the matching criteria of the Fusion Node and, after the recovery attempt fails, a new global track is initialized. Consequently, U1 transitions from id_3 to id_6, while U3 transitions from id_2 to id_4. These events correspond to a fragmentation of the global

identity within the same visibility episode, rather than to an exchange of identities between different participants. After the new global identities are established, however, they remain consistently associated with the corresponding participants up to the end of the experiment. U2, instead, does not experience any global-ID switch throughout the entire experiment.

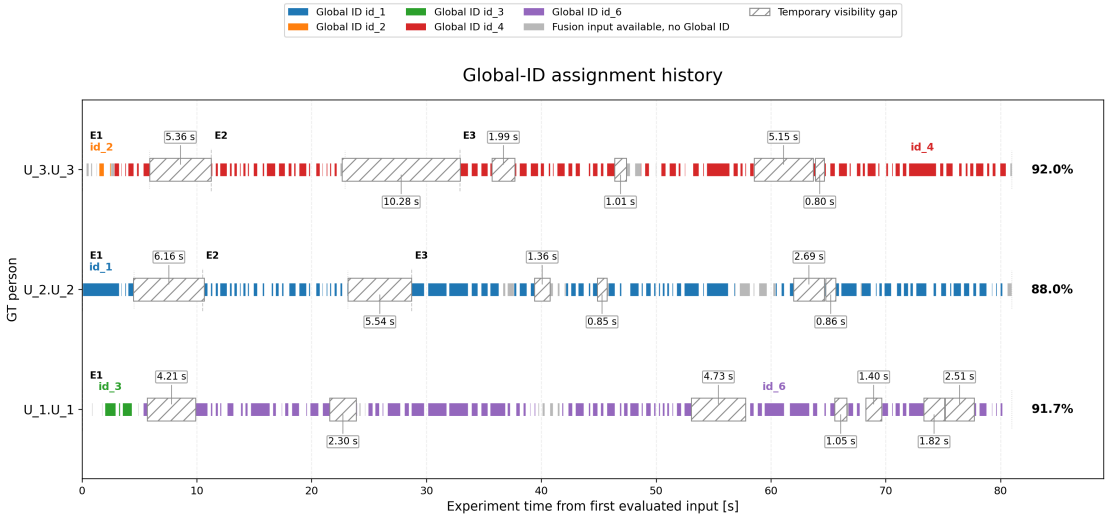


Figure 6.9: Temporal global-ID assignment across the visibility episodes in the first real-world experiment.

Analyzing the evolution of the tracks, the results also show that temporary observation losses do not necessarily result in the initialization of a different global identity, even when the loss overcomes the recovery horizon and a new episode starts. Several interruptions are successfully bridged while preserving the previously established global ID, thanks to strong geometric compatibility and appearance consistency among the incoming observations. The Fusion Node is therefore able to reassociate the new observation with the previously established global identity. The three participants exhibit different episode structures during the experiment. U1 is evaluated within a single identity-continuity episode. U2 is instead characterized by three independent episodes, while preserving id_1 across all of them. Similarly, U3 generates three episodes and, after the initial transition from id_2 to id_4, the same id_4 is reassociated across all subsequent episodes. These latter cases are particularly relevant because identity preservation across independent episodes is not required by the evaluation criterion, yet the Fusion Node is still able to recover a compatible previously established global identity. The temporal representation also shows that a global ID is available for most of the intervals in which valid local observations are provided to the Fusion Node. The residual unassigned intervals in which no valid global association is produced are relatively small and represented

in grey. The resulting global association coverage, together with the Global-ID purity, Global-ID stability, number of switches, and re-entry preservation for each participant, is summarized in Table 6.5.

GT person	Coverage	ID purity	ID stability	Switches	Re-entry preservation
U1	95.39%	96.11%	91.69%	1	100%
U2	88.02%	100.00%	88.02%	0	100%
U3	93.80%	98.13%	92.05%	1	94.44%

Table 6.5: Fusion-specific metrics for each participant in the first real-world experiment.

Considering the experiment as a whole, the global association coverage reaches 92.24%, while the overall Global-ID purity is 99.06% and the Global-ID stability reaches 91.37%. A total of two global-ID switches are observed, corresponding to the transitions previously identified for U1 and U3. Overall, these results indicate that, despite the two initial identity fragmentations, the global tracking output remains highly stable throughout most of the experiment.

A qualitative analysis of the trajectories estimated by the multi-camera tracking system is performed by comparing them with the corresponding Vicon ground-truth trajectories over two distinct time windows of the first experimental recording. In particular, in the first interval, covering approximately 12 s of the recording, the three participants move mainly in parallel along the laboratory corridor, as shown in Figure 6.10. The reconstructed trajectories preserve the overall direction and shape of the participants' motion and remain consistently associated with the corresponding individuals.

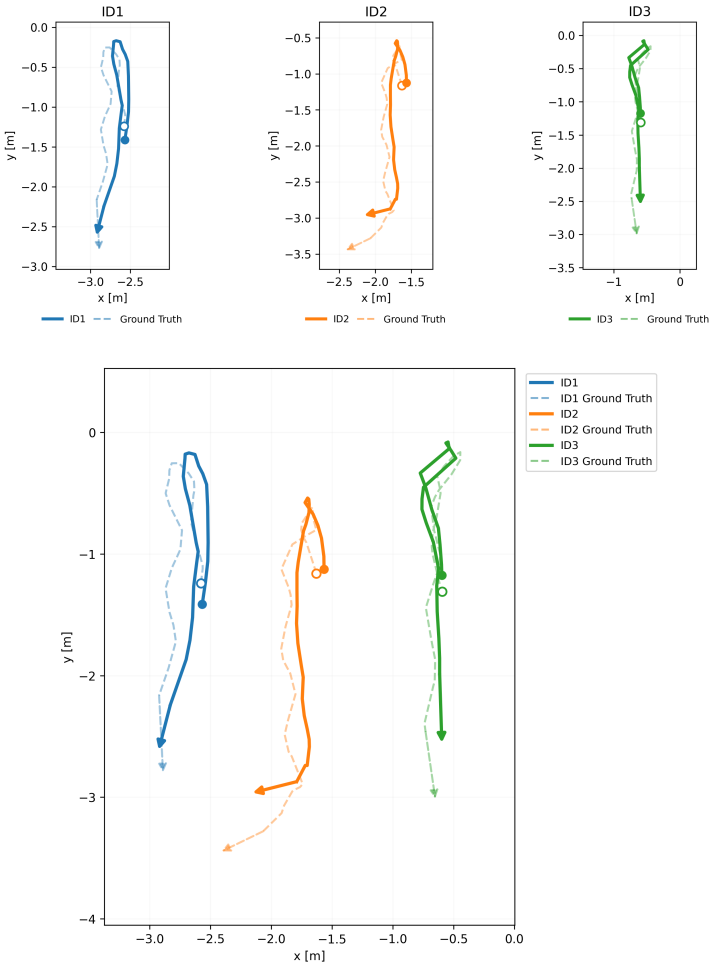


Figure 6.10: Global and ground-truth trajectories in the first temporal window: individual trajectory comparisons and combined representation.

A second interval, spanning approximately 20 s, captures instead a more challenging configuration, in which the participants move along intersecting paths. As illustrated in Figure 6.11, the global tracks reproduce the main evolution of the corresponding ground-truth trajectories. Despite the increased complexity of the relative motion, the estimated trajectories remain associated with the corresponding participants, while some spatial deviations with respect to the Vicon reference can be observed.

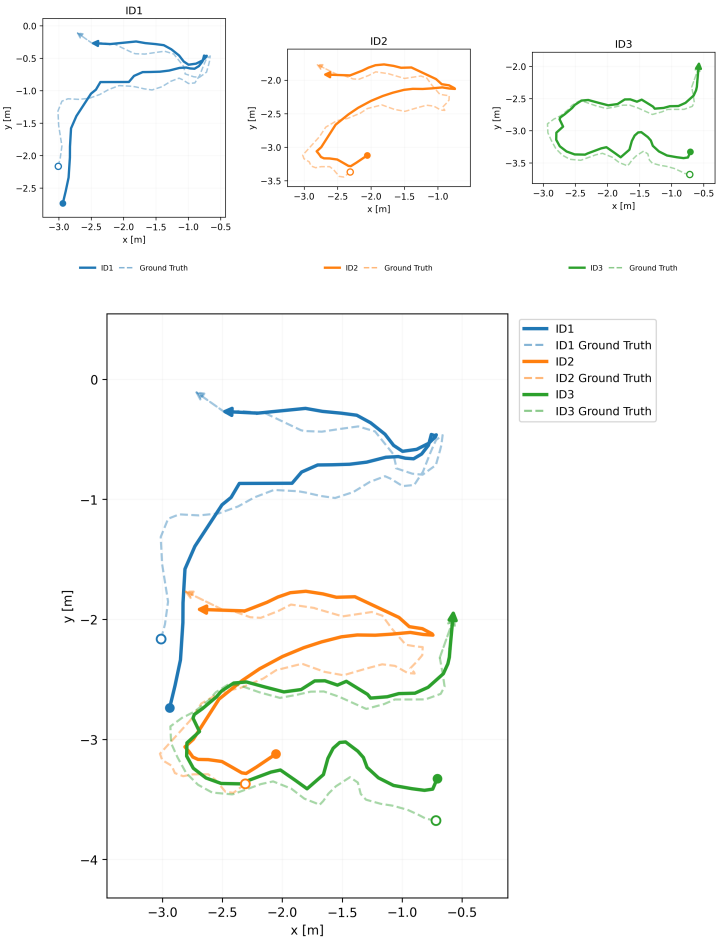


Figure 6.11: Global and ground-truth trajectories in the second temporal window: individual trajectory comparisons and combined representation.

Second experiment.
The second real-world experiment exhibits a similar identity-association behavior and is therefore summarized quantitatively in Tables 6.6 and 6.7. In this experiment, identity changes were mainly concentrated in the initial part of the recording, before the global tracks reached stable associations. In particular, participant U1 experienced two global-ID switches during the initial tracking phase, after which the assigned identity remained stable for the remainder of the recording. A similar initial identity reassignment was observed for U2, whereas no within-episode global-ID switch occurred for U3. For the latter participant, the change in global ID occurred at the beginning of a new identity-continuity episode and is therefore not classified as an identity switch according to the recovery-aware evaluation methodology

introduced in Section 6.1.1. Inspection of the fusion events showed that the initial ID switches observed for U1 and U2 did not correspond to identity exchanges between different participants, but rather to temporary fragmentation of the global ID during cross-camera associations. In these cases, the fusion strategy was unable to recover the previously assigned global ID, while the conditions required to prevent the creation of a new global track were not simultaneously satisfied. In particular, although the incoming observations could remain geometrically close to an existing track, the combined geometric and appearance compatibility required by the recovery and duplicate-protection mechanisms was not sufficiently strong. Consequently, a new global ID was initialized for the participant.

Overall, the experiment confirms that the observed identity inconsistencies are mainly associated with short-lived fragmentation, rather than with incorrect identity exchanges between different targets. Once a stable global ID was established, persistent associations were generally maintained, even over multiple independent identity-continuity episodes.

Participant	Episode	Dominant global ID	Global-ID stability [%]	ID switches
U1	E1	id_6	85.0	2
U2	E1	id_4	89.4	1
U3	E1	id_3	86.0	0
U3	E2	id_8	99.5	0

Table 6.6: Identity-continuity episodes and dominant global IDs in the second real-world experiment.

Metric	Result
Evaluable agents	3
Identity-continuity episodes	4
Global association coverage	95.44%
Global-ID purity	94.14%
Global-ID stability	89.85%
Global-ID switches	3
Association interruptions	2

Table 6.7: Fusion-specific evaluation results for the second real-world experiment.

The fusion-specific results show a global association coverage of 95.44%, indicating that a global ID is available for most of the fusion-evaluable duration.

To complement the identity-consistency analysis, the spatial behavior of the tracking system was also examined by comparing the estimated global track trajectories with the corresponding Vicon ground-truth trajectories. A representative time interval within the recorded bag in which all the participants were simultaneously tracked was selected, allowing their motion to be evaluated under the same observation conditions. Figure 6.12 shows the different motion patterns within the monitored area, providing a qualitative assessment of the ability of the multi-camera system to reproduce the spatial evolution of each participant. The estimated global tracks generally follow the overall shape and direction of the corresponding ground-truth trajectories.

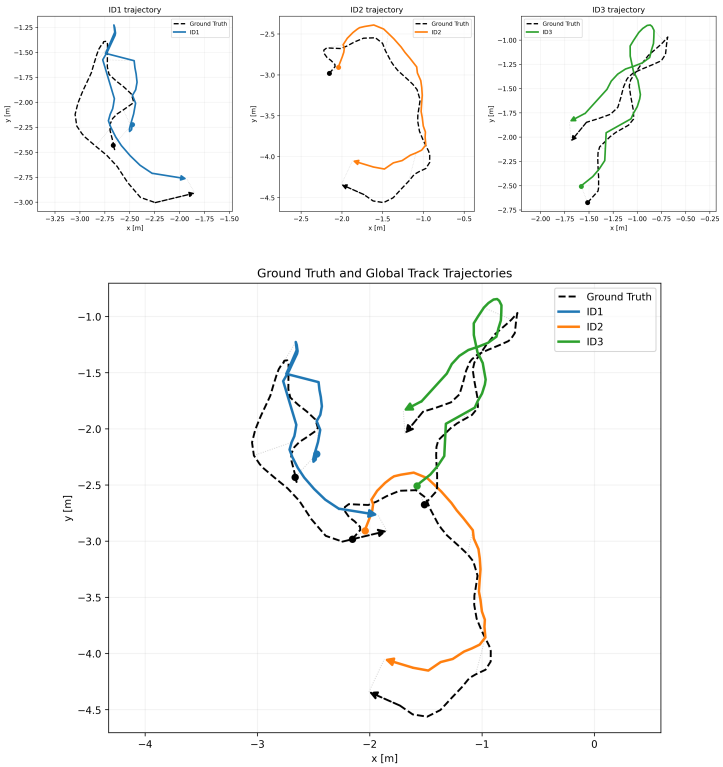


Figure 6.12: Global and ground-truth trajectories in the selected temporal window of the second real-world experiment: individual trajectory comparisons and combined representation.

Third experiment.
The third real-world experiment represents a stress test for the developed tracking system, introducing substantially more challenging conditions than in the previous recordings. In this experiment, the robot was repeatedly and rapidly moved within

the limited space available in the laboratory, producing frequent and abrupt changes in its orientation with respect to the participants. These continuous changes in viewpoint strongly affected target observability. During the rapid rotations of the robot, participants frequently left the geometrically observable region covered by the three cameras and could remain unobserved until the robot's orientation again provided a suitable field of view. In addition, further temporary losses could occur before a valid local tracking observation became available to the Fusion Node. The experiment therefore generated repeated and considerably longer interruptions in the observation of the participants than those encountered in the first two real-world recordings. Several of these interruptions exceeded the temporal recovery horizon within which an existing global track can be maintained or reactivated. Consequently, the newly acquired observations were assigned to new identity-continuity episodes. In these conditions, a participant observed again after a substantial change in robot orientation is processed as a newly encountered target and therefore is assigned a new global ID.

This behavior results in a considerably larger number of global IDs over the complete recording than in the previous experiments. The impact of these challenging observation conditions on the global tracking process is reflected in the fusion-specific metrics reported in Table 6.8.

Metric	**Real-World Experiment 3**
Evaluable agents	3
Identity-continuity episodes	13
Global association coverage	79.58%
Global-ID purity	92.42%
Global-ID stability	73.55%
Within-episode global-ID switches	2
Association interruptions	4

Table 6.8: Fusion-specific evaluation results for the third real-world experiment.

The three evaluable participants generate a total of 13 independent identity-continuity episodes, reflecting the repeated and prolonged observation losses occurring during the experiment. Despite these challenging conditions, the global-ID purity remains relatively high at 92.42%, while only two within-episode global-ID switches are observed.

Overall, the real-world experiments exhibit lower tracking performance than the simulated scenarios, as expected from the additional uncertainty introduced by the physical perception pipeline. In particular, RGB-D measurements are

affected by depth uncertainty and estimate points on the visible body surface rather than the reference position measured by the Vicon system. Together with the transformations from the individual camera frames to the common map frame, this introduces spatial discrepancies that directly affect the point-based MOT evaluation, in addition to the challenges associated with cross-camera identity management. For this reason, HOTA, DetA, and AssA are reported both as averages over the complete range $\alpha = 0.05 - 0.95$ and at $\alpha = 0.50$. The averaged values provide an overall evaluation across progressively stricter spatial matching conditions, whereas, with the adopted distance scale $D_{\text{HOTA}} = 1.5\,\text{m}$, $\alpha = 0.50$ corresponds to a maximum association distance of 0.75 m, more suitable as a reference condition for the considered RGB-D tracking system. Both values are reported in Table 6.9 to preserve the complete HOTA evaluation while allowing the effect of stricter spatial thresholds on the real-world results to be distinguished more clearly.

Metric	**Experiment 1**	**Experiment 2**	**Experiment 3**
HOTA avg.	59.87%	65.86%	55.65%
DetA avg.	60.33%	65.11%	62.46%
AssA avg.	59.44%	66.64%	50.01%
HOTA @ 0.50	82.93%	85.80%	73.65%
DetA @ 0.50	84.06%	85.20%	77.48%
AssA @ 0.50	81.81%	86.50%	70.00%
MOTA	82.20%	82.72%	75.34%
IDF1	88.97%	87.85%	73.63%
ID Precision	88.41%	84.60%	77.36%
ID Recall	89.54%	91.36%	70.24%
ID switches	2	3	2

Table 6.9: Standard MOT metrics for the three real-world experiments.

The first two experiments show comparable performance, with MOTA values around 82%, IDF1 close to 88–89%, and substantially higher HOTA-based scores at $\alpha = 0.50$ than over the complete threshold range. Together with the limited number of within-episode ID switches, these results are consistent with the previously observed behavior, in which most identity inconsistencies correspond to short-lived fragmentation rather than ID exchanges failures. Clearly lower metrics values appear in the third experiment, where the operating conditions become harder. In particular, the values of DetA (77.48%) and AssA (70.00%) at $\alpha = 0.50$ indicate that, although a substantial fraction of the observations can still be spatially matched, maintaining consistent trajectory-level associations becomes

more challenging under repeated losses and reacquisitions. Nevertheless, the larger number of global IDs generated in this experiment is mainly due to observations belonging to new independent visibility episodes, thus interpreted as new targets in the scene.

6.5 Social Navigation Results

The final experimental phase focused on the integration of the resulting data streamed through the `/tracked_people` topic into the local controller for robot navigation. In particular, the global pedestrian information provided by the multi-camera tracking and fusion pipeline was made available to the MPPI controller, enabling the *Social Critic* introduced among its optimization plugins to correctly account for the presence and motion of surrounding people during trajectory evaluation.

The validation of this component was conducted exclusively in the simulated environment and was not extended to real-world trials. The simulated environment was therefore used both for the extensive tuning of the controller critic parameters and for the final evaluation of the resulting navigation behavior. During the tuning process, particular attention was devoted to achieving an appropriate trade-off between social compliance and the primary navigation objectives of the controller. Excessively strong social penalties can cause the robot to deviate unnecessarily from the planned path or excessively restrict its motion, whereas insufficient weighting reduces the influence of nearby pedestrians on the selected trajectory, as discussed in the last section of Chapter 4. The main adjustments focused on the weights of the *PathAlignCritic*, *Obstacle* and *Costmap* critics, and *PreferForwardCritic*, which contribute to the fundamental navigation objectives, and on their balance with the weight assigned to the *Social Critic*. The final configuration provided a suitable balance between these competing objectives, allowing the robot to safely move through the simulated indoor scenario among people, while successfully reaching the desired final location.

The obtained results showed satisfactory navigation performance. When pedestrian interactions affected the locally available trajectories, the MPPI controller adapted the robot's motion according to the contribution of the *Social Critic*. Once the pedestrian interaction was no longer relevant, the robot progressively returned towards the planned path and continued its motion toward the assigned goal. The resulting behavior therefore preserved the main navigation objective while incorporating the pedestrian information generated by the perception framework into the local trajectory-selection process.

Figure 6.13 provides representative snapshots of the resulting navigation behaviour during the simulated experiment. The planned path, represented by the green line, is shown together with the optimal trajectory, represented by the dotted line, selected by the MPPI controller at each considered step. At the same time, the surrounding pedestrians are represented through the Global-ID markers generated by the multi-camera tracking and fusion pipeline. The sequence highlights how the local trajectory is continuously evaluated while accounting for the perceived presence of surrounding people. The robot remains well aligned with the planned path while adapting its local motion according to the pedestrian information and progressing toward the assigned goal.

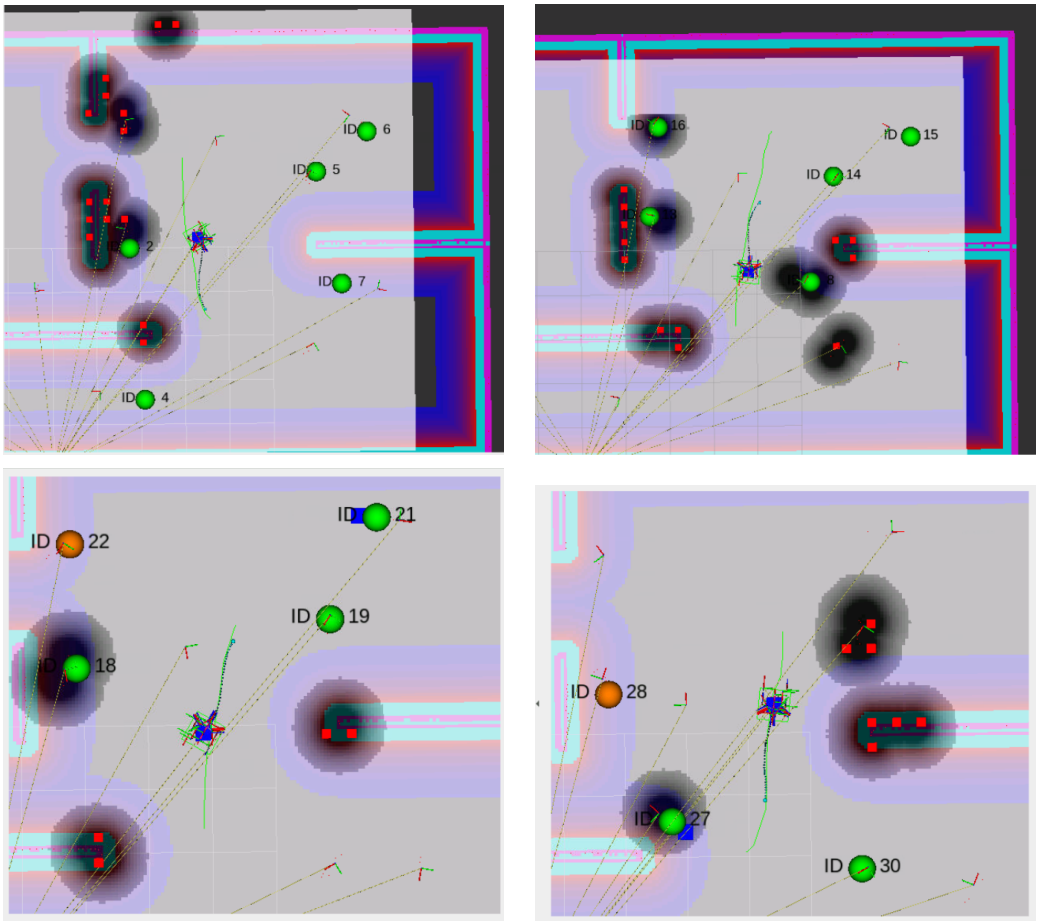


Figure 6.13: Representative RViz snapshots of the simulated social navigation experiment.

Chapter 7

Conclusions

This thesis addressed the perception task with the objective of improving the navigation capabilities of a mobile service robot operating in indoor environments shared with people. In this context, a multi-camera architecture was developed for the Clearpath Jackal platform to extend the observable region around the robot. Independent detection and tracking pipelines were integrated with a global fusion strategy specifically developed to obtain a unified representation of the surroundings. The experimental validation showed overall satisfactory performance of the developed perception framework. Particularly high results were obtained in the simulated experiments, where both the fusion-specific evaluation and the standard MOT metrics demonstrated high association coverage and identity consistency. As expected, a reduction in performance was observed when moving from simulation to the real-world experiments. Real sensing introduces several sources of uncertainty that are absent or considerably reduced in simulation. In particular, RGB-D measurements are affected by depth uncertainty and measurement noise, while the estimated pedestrian position corresponds to points reconstructed from the visible body surface rather than exactly to the reference position provided by the Vicon system. Additional discrepancies can also be introduced by the transformations required to express the observations of the three independent cameras in the common reference frame. In general, pedestrian motion under real-world conditions makes the association problem intrinsically more challenging. Despite these additional difficulties, the real-world experiments maintained good overall tracking performance. More challenging conditions were intentionally introduced in the third experiment, where the robot was repeatedly moved and rapidly reoriented. The restricted dimensions of the experimental environment further increased the difficulty, forcing a larger number of Global IDs and independent visibility episodes. The achieved tracking performance also resulted from an extensive iterative tuning process. The global fusion strategy relies on several parameters controlling the different stages of

the tracking process, including geometric and appearance compatibility thresholds, ambiguity conditions, duplicate suppression criteria, and the temporal parameters governing the different phases of the Global Track lifetime. Their values were progressively adjusted according to the behavior observed during the development and experimental validation until obtaining the final configuration presented in this work. The resulting behavior therefore demonstrates the integration of the developed perception framework within a complete socially aware navigation architecture, although the final closed-loop navigation validation was limited to the simulated environment. Although the adopted procedure allowed satisfactory performance to be achieved, several directions can be considered for future development. A first natural extension would be the validation of the complete perception–navigation architecture on the physical robot, evaluating the interaction between real multi-camera perception and socially aware motion planning in closed loop. Experiments could also be extended to larger and more representative environments, such as long corridors, public buildings, museums, or airport-like spaces, where the robot could perform more natural and extended trajectories and interact with a larger number of pedestrians without the motion constraints imposed by the small laboratory environment. Further improvements could address the uncertainty associated with real measurements directly within the fusion process. Rather than representing each pedestrian observation only through an estimated position, future implementations could associate the measurement with an explicit estimate of its spatial uncertainty and exploit this information during prediction, gating, and data association. This could make the fusion strategy more robust to variations in RGB-D measurement quality and to observations acquired under different sensing geometries. Finally, the dependence on manually tuned parameters could be reduced through adaptive or learning-based optimization strategies. Selected fusion parameters could be adjusted according to the characteristics of the observed scene, or their values could be optimized from experience collected while the robot operates and interacts with surrounding agents. Such an approach could improve the adaptability of the tracking system to different environments and operating conditions. Overall, the experimental results demonstrate the feasibility of the proposed approach while also identifying the main challenges that arise when moving from controlled simulation to physical perception. These results provide a basis for further developments toward more robust socially aware navigation of mobile service robots.

Bibliography

[1] Finpiemonte. *ALBA Robot: la startup innovativa che trasforma i veicoli personali elettrici in veicoli autonomi*. URL: https://www.finpiemonte.it/news/alba-robot-la-startup-innovativa-che-trasforma-i-veicoli-personali-elettrici-veicoli-autonomi (visited on 09/12/2026) (cit. on p. 2).

[2] Amitava Chatterjee, Anjan Rakshit, and N. Nirmal Singh. «Mobile Robot Navigation». In: *Vision Based Autonomous Robot Navigation: Algorithms and Implementations*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 1–20. ISBN: 978-3-642-33965-3. DOI: 10.1007/978-3-642-33965-3_1. URL: https://doi.org/10.1007/978-3-642-33965-3_1 (cit. on p. 2).

[3] Tod S. Levitt and Daryl T. Lawton. «Qualitative Navigation for Mobile Robots». In: *Artificial Intelligence* 44.3 (1990), pp. 305–360. ISSN: 0004-3702. DOI: 10.1016/0004-3702(90)90027-W. URL: https://www.sciencedirect.com/science/article/pii/000437029090027W (cit. on p. 2).

[4] Phani Teja Singamaneni, Pilar Bachiller-Burgos, Rachid Alami, et al. «A Survey on Socially Aware Robot Navigation: Taxonomy and Future Challenges». In: *The International Journal of Robotics Research* 43.10 (2024). DOI: 10.1177/02783649241230562. URL: https://journals.sagepub.com/doi/full/10.1177/02783649241230562 (cit. on pp. 3, 4).

[5] Gonzalo Ferrer, Anaıs Garrell, and Alberto Sanfeliu. *Social-Aware Robot Navigation in Urban Environments*. 2013 (cit. on p. 3).

[6] Richard Szeliski. *Computer Vision: Algorithms and Applications*. 2nd ed. Cham: Springer Nature Switzerland, 2022. ISBN: 978-3-030-34371-2. DOI: 10.1007/978-3-030-34372-9. URL: https://doi.org/10.1007/978-3-030-34372-9 (cit. on p. 4).

[7] Nicola Bellotto and Huosheng Hu. «Multisensor-Based Human Detection and Tracking for Mobile Service Robots». In: *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics* 39.1 (2009), pp. 167–181. ISSN: 1083-4419. DOI: 10.1109/TSMCB.2008.2004050. URL: https://doi.org/10.1109/TSMCB.2008.2004050 (cit. on p. 4).

[8] Wenhan Luo, Junliang Xing, Anton Milan, Xiaoqin Zhang, Wei Liu, and Tae-Kyun Kim. «Multiple Object Tracking: A Literature Review». In: *Artificial Intelligence* 293 (2021), p. 103448. ISSN: 0004-3702. DOI: 10.1016/j.artint.2021.103448. URL: https://doi.org/10.1016/j.artint.2021.103448 (cit. on pp. 4, 15, 40).

[9] Gioele Ciaparrone, Francisco Luque Sánchez, Siham Tabik, Luigi Troiano, Roberto Tagliaferri, and Francisco Herrera. «Deep Learning in Video Multi-Object Tracking: A Survey». In: *Neurocomputing* 381 (2020), pp. 61–88. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2019.11.023. URL: https://doi.org/10.1016/j.neucom.2019.11.023 (cit. on p. 4).

[10] Temitope Ibrahim Amosa, Patrick Sebastian, Lila Iznita Izhar, Oladimeji Ibrahim, Lukman Shehu Ayinla, Abdulrahman Abdullah Bahashwan, Abubakar Bala, and Yau Alhaji Samaila. «Multi-camera Multi-object Tracking: A Review of Current Trends and Future Advances». In: *Neurocomputing* 552 (2023), p. 126558. DOI: 10.1016/j.neucom.2023.126558. URL: https://doi.org/10.1016/j.neucom.2023.126558 (cit. on pp. 4, 5, 15, 16).

[11] Bruno Siciliano and Oussama Khatib, eds. *Springer Handbook of Robotics*. 2nd ed. Cham: Springer International Publishing, 2016. ISBN: 978-3-319-32550-7. DOI: 10.1007/978-3-319-32552-1. URL: https://doi.org/10.1007/978-3-319-32552-1 (cit. on p. 5).

[12] Bernhard Rinner and Markus Quaritsch. «Multicamera Tracking». In: *Computer Vision: Detection, Recognition and Reconstruction*. Ed. by Ender Konukoglu and Bastian Leibe. Cham: Springer International Publishing, 2020, pp. 373–401. DOI: 10.1007/978-3-030-03009-4_13. URL: https://doi.org/10.1007/978-3-030-03009-4_13 (cit. on p. 5).

[13] Michael Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. «ROS: An Open-Source Robot Operating System». In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) Workshop on Open Source Software*. Kobe, Japan, 2009 (cit. on p. 6).

[14] Steven Macenski, Tully Foote, Brian Gerkey, Chris Lalancette, and William Woodall. «Robot Operating System 2: Design, Architecture, and Uses in the Wild». In: *Science Robotics* 7.66 (2022), eabm6074. DOI: 10.1126/scirobotics.abm6074. URL: https://doi.org/10.1126/scirobotics.abm6074 (cit. on p. 6).

[15] Sumin Kang, Sungwoo Yang, Daewon Kwak, Yura Jargalbaatar, and Donghan Kim. «Social Type-Aware Navigation Framework for Mobile Robots in Human-Shared Environments». In: *Sensors* 24.15 (2024), p. 4862. DOI: 10.3390/s24154862 (cit. on p. 8).

[16] Steven Macenski, Tyler Moore, David V. Lu, Alexander Merzlyakov, and Michael Ferguson. «From the Desks of ROS Maintainers: A Survey of Modern and Capable Mobile Robotics Algorithms in the Robot Operating System 2». In: *Robotics and Autonomous Systems* 168 (2023), p. 104493. ISSN: 0921-8890. DOI: 10.1016/j.robot.2023.104493. URL: https://doi.org/10.1016/j.robot.2023.104493 (cit. on p. 10).

[17] Steve Macenski, David V. Lu, Francisco Martín, Ruffin White, and Jonatan Ginés Clavero. «The Marathon 2: A Navigation System». In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2020 (cit. on pp. 10, 18, 19).

[18] Dieter Fox, Wolfram Burgard, and Sebastian Thrun. «The Dynamic Window Approach to Collision Avoidance». In: *IEEE Robotics & Automation Magazine* 4.1 (1997), pp. 23–33. DOI: 10.1109/100.580977 (cit. on p. 10).

[19] Christoph Rösmann, Frank Hoffmann, and Torsten Bertram. «Integrated Online Trajectory Planning and Optimization in Distinctive Topologies». In: *Robotics and Autonomous Systems* 88 (2017), pp. 142–153. ISSN: 0921-8890. DOI: 10.1016/j.robot.2016.11.018. URL: https://doi.org/10.1016/j.robot.2016.11.018 (cit. on p. 10).

[20] James B. Rawlings, David Q. Mayne, and Moritz M. Diehl. *Model Predictive Control: Theory, Computation, and Design*. 2nd ed. Madison, WI: Nob Hill Publishing, 2020 (cit. on p. 10).

[21] Grady Williams, Paul Drews, Brian Goldfain, James M. Rehg, and Evangelos A. Theodorou. «Aggressive Driving with Model Predictive Path Integral Control». In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. Stockholm, Sweden, 2016, pp. 1433–1440. DOI: 10.1109/ICRA.2016.7487277. URL: https://doi.org/10.1109/ICRA.2016.7487277 (cit. on pp. 11, 24, 26, 27).

[22] Dirk Helbing and Péter Molnár. «Social Force Model for Pedestrian Dynamics». In: *Physical Review E* 51.5 (1995), pp. 4282–4286. DOI: 10.1103/PhysRevE.51.4282. URL: https://doi.org/10.1103/PhysRevE.51.4282 (cit. on pp. 12, 32).

[23] M. Martini, N. Pérez-Higueras, A. Ostuni, M. Chiaberge, F. Caballero, and L. Merino. «Adaptive Social Force Window Planner with Reinforcement Learning». In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2024, pp. 4816–4822 (cit. on p. 12).

[24] N. Pérez-Higueras. *LightSFM: A Lightweight Implementation of the Social Force Model for Social Local Navigation*. GitHub repository. URL: https://github.com/robotics-upo/lightsfm (cit. on pp. 12, 32).

[25] Elia Trevisan, Khaled A. Mustafa, Godert Notten, Xinwei Wang, and Javier Alonso-Mora. «Dynamic Risk-Aware MPPI for Mobile Robots in Crowds via Efficient Monte Carlo Approximations». In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Hangzhou, China, 2025, pp. 313–320. DOI: 10.1109/IROS60139.2025.11246822. URL: https://doi.org/10.1109/IROS60139.2025.11246822 (cit. on p. 13).

[26] Niko Picello, Fernando Herrero, Sergi Hernández, Alejandro López, and Angel Santamaria-Navarro. «Leveraging Pedestrian Detection and Tracking in Robotics Navigation: A Survey With Practical Illustrations». In: *IEEE Access* 13 (2025), pp. 158926–158937. DOI: 10.1109/ACCESS.2025.3607191 (cit. on pp. 13, 14).

[27] Muhammad Sajid, Naoufal Raissouni, and Abdelilah Hakim. «Human Detection and Tracking for Mobile Robots: A Survey». In: *IEEE Access* 11 (2023), pp. 54472–54502. DOI: 10.1109/ACCESS.2023.3279506. URL: https://doi.org/10.1109/ACCESS.2023.3279506 (cit. on pp. 14, 15).

[28] Zhenyu Xie, Zelin Ni, Wenjie Yang, Yuang Zhang, Yihang Chen, Yang Zhang, and Xiao Ma. «A Robust Online Multi-Camera People Tracking System With Geometric Consistency and State-aware Re-ID Correction». In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops)*. 2024, pp. 7007–7016. DOI: 10.1109/CVPRW63382.2024.00694 (cit. on p. 16).

[29] Xiaoyu Li, Peidong Li, Lijun Zhao, Dedong Liu, Jinghan Gao, Xian Wu, Yitao Wu, and Dixiao Cui. «RockTrack: A 3D Robust Multi-Camera-Ken Multi-Object Tracking Framework». In: *arXiv preprint arXiv:2409.11749* (2024). DOI: 10.48550/arXiv.2409.11749 (cit. on p. 16).

[30] Bryan J. Thibodeau, Stephen W. Hart, Deepak R. Karuppiah, John D. Sweeney, and Oliver Brock. «Cascaded Filter Approach to Multi-Objective Control». In: *Proceedings of the 2004 IEEE International Conference on Robotics and Automation (ICRA)*. 2004, pp. 3877–3882. DOI: 10.1109/ROBOT.2004.1308872 (cit. on p. 22).

[31] Open Navigation LLC. *MPPI Controller*. Navigation2 Documentation. Navigation2. URL: https://docs.nav2.org/configuration/packages/configuring-mppic.html (visited on 08/18/2026) (cit. on pp. 29, 30).

[32] Fei-Fei Li, Ranjay Krishna, and Danfei Xu. *CS231n: Convolutional Neural Networks for Visual Recognition, Lecture 12: Detection and Segmentation*. Stanford University. Course lecture slides. 2020 (cit. on p. 36).

[33] Fei-Fei Li, Justin Johnson, and Serena Yeung. *CS231n: Convolutional Neural Networks for Visual Recognition, Lecture 11: Detection and Segmentation*. Course lecture slides. 2017 (cit. on p. 36).

[34] Anastasiia Zharovskikh. *Empowering Sports with 3D Pose Estimation*. InData Labs. May 2020. URL: https://indatalabs.com/blog/3d-pose-estimation (visited on 09/03/2026) (cit. on p. 37).

[35] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. *Ultralytics YOLOv8*. Version 8.0.0. 2023. URL: https://github.com/ultralytics/ultralytics (visited on 09/03/2026) (cit. on pp. 38, 39).

[36] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. «You Only Look Once: Unified, Real-Time Object Detection». In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 779–788 (cit. on p. 38).

[37] Allan Kouidri. *YOLOv8 Pose Estimation: Next-Gen KeyPoint Detection*. Ikomia. Jan. 2024. URL: https://www.ikomia.ai/blog/yolov8-pose-advanced-estimation (visited on 09/03/2026) (cit. on p. 39).

[38] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. «Simple Online and Realtime Tracking». In: *2016 IEEE International Conference on Image Processing (ICIP)*. 2016, pp. 3464–3468. DOI: 10.1109/ICIP.2016.7533003 (cit. on p. 41).

[39] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. «Simple Online and Realtime Tracking with a Deep Association Metric». In: *2017 IEEE International Conference on Image Processing (ICIP)*. 2017, pp. 3645–3649. DOI: 10.1109/ICIP.2017.8296962 (cit. on pp. 42–44).

[40] Yunhao Du, Zhicheng Zhao, Yang Song, Yanyun Zhao, Fei Su, Tao Gong, and Hongying Meng. «StrongSORT: Make DeepSORT Great Again». In: *IEEE Transactions on Multimedia* 25 (2023), pp. 8725–8737. DOI: 10.1109/TMM.2023.3240881 (cit. on pp. 44, 45).

[41] Nir Aharon, Roy Orfaig, and Ben-Zion Bobrovsky. «BoT-SORT: Robust Associations Multi-Pedestrian Tracking». In: *arXiv preprint arXiv:2206.14651* (2022). DOI: 10.48550/arXiv.2206.14651 (cit. on pp. 46, 48).

[42] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. «ByteTrack: Multi-Object Tracking by Associating Every Detection Box». In: *Computer Vision – ECCV 2022*. Vol. 13682. Lecture Notes in Computer Science. Springer, 2022, pp. 1–21. DOI: 10.1007/978-3-031-20047-2_1 (cit. on p. 48).

[43] Clearpath Robotics. *Jackal Unmanned Ground Vehicle: Technical Specifications*. Clearpath Robotics. URL: https://www.clearpathrobotics.com/wp-content/uploads/2014/09/JACKAL-_DATA-SHEET2.pdf (visited on 08/31/2026) (cit. on pp. 52, 92).

[44] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. *Ultralytics YOLOv8*. Version 8.0.0. 2023. URL: https://github.com/ultralytics/ultralytics (cit. on p. 91).

[45] Kaiyang Zhou, Yongxin Yang, Andrea Cavallaro, and Tao Xiang. «Omni-Scale Feature Learning for Person Re-Identification». In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 3702–3712. DOI: 10.1109/ICCV.2019.00380 (cit. on p. 91).

[46] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. «Microsoft COCO: Common Objects in Context». In: *Computer Vision – ECCV 2014*. Springer, 2014, pp. 740–755. DOI: 10.1007/978-3-319-10602-1_48 (cit. on p. 91).

[47] 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)*. 2015, pp. 1116–1124. DOI: 10.1109/ICCV.2015.133 (cit. on p. 91).

[48] Intel Corporation. *Intel RealSense D400 Series Product Family Datasheet*. 337029. Intel Corporation. URL: https://www.intelrealsense.com/wp-content/uploads/2022/04/Intel-RealSense-D400-Series-Datasheet-April-2022.pdf (visited on 08/31/2026) (cit. on pp. 95, 98).

[49] Xinhua Wang, Dayu Li, and Guang Zhang. «Panoramic Stereo Imaging of a Bionic Compound-Eye Based on Binocular Vision». In: *Sensors* 21.6 (2021), p. 1944. DOI: 10.3390/s21061944 (cit. on p. 96).

[50] Anders Grunnet-Jepsen, John N. Sweetser, Paul Winer, Akihiro Takagi, and John Woodfill. *Projectors for Intel RealSense Depth Cameras D4xx*. White Paper. Intel Corporation, 2018. URL: https://www.intelrealsense.com/wp-content/uploads/2019/03/WhitePaper_on_Projectors_for_RealSense_D4xx_1.0.pdf (visited on 08/31/2026) (cit. on p. 96).

[51] Mingyu Jang, Hyunse Yoon, Seongmin Lee, Jiwoo Kang, and Sanghoon Lee. «A Comparison and Evaluation of Stereo Matching on Active Stereo Images». In: *Sensors* 22.9 (2022), p. 3332. DOI: 10.3390/s22093332 (cit. on p. 97).

[52] Maoyong Li, Jiqiang Huang, Long Xue, and Ruiying Zhang. «A Guidance System for Robotic Welding Based on an Improved YOLOv5 Algorithm with a RealSense Depth Camera». In: *Scientific Reports* 13 (2023), p. 21299. DOI: 10.1038/s41598-023-48318-8 (cit. on p. 99).