TrackAI - Human Detector and Tracker
ENPM700 MidTerm Project Proposal

Datta Lohith Gannavarapu (UID: 119455395)
Dheeraj Vishnubhotla (UID: 120427367)
Nazrin Gurbanova (UID: 120426469)

Introduction

In autonomous robotic systems, the three core components are perception, planning, and control. Together, these components allow the robot to observe its environment, make informed decisions, and carry out actions to achieve its objectives. Perception, in particular, plays a critical role in integrating real-time environmental data into the robot's planning algorithm. This project focuses on perception, specifically on detecting and tracking human obstacles using the monocular camera of a robot.

Problem Statement

For a mobile robot operating in confined or open spaces, detecting obstacles such as humans and objects is a vital task. The primary goal of this project is to develop a perception software module for a ACME Robotics. This module will enable the robot to detect humans and continuously monitor their movements to avoid collisions and navigate efficiently in dynamic environments.

Human detection involves identifying the presence of people in a specific area using algorithms that analyze visual data to locate and recognize features that represent human figures. Human tracking refers to the continuous monitoring of human movements by assigning a unique ID to each person and tracking their position across video frames. The primary challenge is to ensure the robot avoids collisions by accurately detecting and tracking humans.

Methodology

The project's algorithm begins with human detection using the pre-trained SOTA YOLOv8 model, which is trained on the COCO dataset. YOLO is a widely-used object detection system, and while the COCO dataset includes multiple object classes, this project focuses primarily on detecting humans. The detected bounding boxes of humans are then tracked using object tracking algorithms such as SORT (Simple Online and Real-time Tracking) and CSRT (Channel and Spatial Reliability Tracker).

SORT utilizes Kalman filters to combine object detection with data association and prediction. CSRT, on the other hand, uses a Discriminative Correlation Filter (DCF) to estimate an object's position by processing both spatial and visual features. Both tracking methods will be evaluated to determine which provides more accurate and reliable results. Once humans are detected and tracked, their positions will be transformed into the robot's coordinate frame or world coordinate frame. This data will then be sent to the navigation module for optimal path planning.

Software Development Process

The project will follow the Agile Iterative Process (AIP) in software engineering, breaking the project into tasks spread over 3-week iterations. Weekly sprint meetings will focus on refining the UML design and high-level project architecture. Collaborative tools, such as Git for version control, will be used, and the project will employ the practice of pair programming throughout the development process.

Software Technologies

- **Programming Languages:** C++
- **Development Tools:** CMake, Git, Cppcheck, Cpplint, Makefiles, VSCode
- **Testing Tools:** Valgrind, Google Test Suite
- **Documentation:** Doxygen
- **Continuous Integration and Code Coverage:** GitHub CI, CodeCov

External Dependencies

- **Detection Model:** YOLOv8
- **Libraries:** OpenCV, Eigen, Tracking algorithms

Potential Risks

- **Real-Time Processing Delays:** YOLOv8 may cause processing delays on limited robot hardware. This can be mitigated by optimizing the model's size, employing quantization, and selecting hardware or creating a master - slave connection with the robot and a capable processing system that meets real-time performance requirements.
- **Accurate Tracking:** Human tracking and ID assignment may fail in cases of occlusion or when multiple humans are detected. This issue will be addressed by refining the tracking algorithm.

Deliverables

- TrackAI - Human detector and tracker application
- Unit tests using the Google Test framework
- UML diagrams
- Detailed algorithm description, risks, technologies used, and dependencies
- Code coverage report using CodeCov
- Developer-level documentation

Team Members and Organization

- **Datta Lohith Gannavarapu** Driver
- **Dheeraj Vishnubhotla** Navigator
- **Nazrin Gurbanova** Design Keeper

The team has adopted the Test-Driven Development (TDD) methodology. In the initial phase, Datta Lohith Gannavarapu will be the driver, Dheeraj Vishnubhotla will be the navigator and Nazrin Gurbanova as the Design keeper. These roles will rotate among team members as the project progresses.

References

- Rajlich, V. *Software Engineering Current Practice*
- Zuraimi, M. A. B., Zaman, F. H. K. ”Vehicle Detection and Tracking using YOLO and DeepSORT,” 2021 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), Penang, Malaysia, 2021, pp. 23-29.
- LearnOpenCV, ”Object Detection and Reidentification with FairMOT,” `https://learnopencv.com/object-tracking-and-reidentification-with-fairmot/`