



ENPM808X: Software Development for Robotics

Midterm Project Proposal

Author 1: Tarun Trilokesh
UID: 118450766

Author 2: Sai Surya Sriramoju
UID: 119224113

Overview

Acme Robotics is on the brink of introducing an innovative robotics solution that aims to revolutionize robot-human interactions in shared environments. To ensure the successful launch of this product, there's a pressing need to focus on the perception module. This proposal presents the blueprint for a Human Obstacle Detection and Tracking system using the YOLO (You Only Look Once) framework.

Component Overview

The focal point of this proposal is the Perception component, specifically the Human Obstacle Detection and Tracking system. Leveraging a monocular camera, the module will detect and track humans in the robot's operational area in real-time using the YOLO framework.

Design and Implementation Strategy

We will adopt best practices in software development, including Test-Driven Development (TDD) and the Agile Iterative Process (AIP). The design intricacies will be illustrated using UML diagrams, which will provide insights into class hierarchies, dependencies, and component interactions.

Technologies and Tools

- Programming Language: C++14
- Build System: CMake
- Unit Testing: Google Test framework
- Static Analysis: cppcheck
- Coding Standard: Google C++ Styleguide with cpplint validation
- Documentation: Doxygen

External Libraries

- YOLO (You Only Look Once) for real-time human detection and tracking
- OpenCV for supplementary image processing tasks

All the libraries selected are open-source, ensuring seamless integration and avoiding potential legal issues.

Algorithms and Approaches

- YOLO framework for rapid and accurate human detection

Defending the Use of YOLO

YOLO, as a state-of-the-art deep learning framework, offers several advantages that make it an ideal choice for our project:

- **Speed:** YOLO performs detection and classification in a single forward pass, making it incredibly fast and suitable for real-time applications.
- **Accuracy:** Despite its speed, YOLO maintains a high level of accuracy, outperforming many other detection methods.
- **Real-time Processing:** YOLO's architecture is optimized for real-time object detection, making it perfect for applications like ours where timely human detection and tracking are crucial.
- **Simplicity:** Without the need for additional tracking systems, our pipeline remains streamlined and efficient.

Challenges and Solutions

- **False detections:** To counteract this, we will utilize the YOLO framework's robustness and supplement it with iterative training.
- **Dealing with occlusions:** In densely populated settings, occlusions can be a challenge. Regular model updates and training can help address this concern.

Key Deliverables

- A robust Human Obstacle Detection and Tracking system using YOLO, accompanied by thorough documentation and tests
- UML diagrams detailing the system design
- A GitHub repository containing the code, tests, and a continuous integration setup
- Detailed developer documentation in the README and via Doxygen annotations
- A working demonstration of the module's efficacy in real-world scenarios

Team Collaboration

Pair programming will be a cornerstone of our development process to ensure high code quality and promote team synergy. Periodic sprint reviews and retrospectives will ensure alignment with Acme's vision and project deadlines.

With a strategic approach and the unparalleled capabilities of the YOLO framework, this proposal aims to expedite the project's progress.