Project Proposal for Acme Robotics

Human Detection and Pose Estimation (HDPE) using Monocular Camera Feed

Proposed By :

FNU Koustubh(120273766) , Keyur Borad(12426049) , Swaraj M Rao(120127007)

Overview and Purpose:

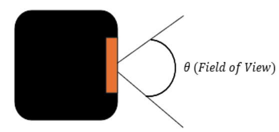
The project aims to develop a Human Detection and Pose Estimation (HDPE) system for Acme Robotics, using a monocular camera feed as the primary sensor input. The system is designed to detect humans in real time, estimate their pose, and generate bounding boxes around detected humans within the camera's field of view. Additionally, it will estimate the depth of each detected human using MiDas, a state-of-the-art deep learning model for monocular depth estimation. The module will output the coordinates of each detected human in the robot's reference frame, enabling the robot to avoid collisions and plan safe trajectories. This perception system will contribute to autonomous navigation by providing obstacle information to the path-planning algorithm.

This project is crucial for Acme Robotics as it provides an essential safety and navigation component, necessary for robots that operate in environments with humans. It ensures that the robots can move autonomously while avoiding collisions, making it suitable for applications in warehouses, healthcare, or service industries.

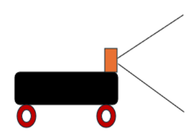
Design and Development Process:

The HDPE module will be developed using Test-Driven Development (TDD) principles, ensuring high code quality and adherence to Acme Robotics' standards. Our development process follows these steps:

- Design Phase :
  - A complete UML diagram (class diagrams and sequence diagrams) will outline the HDPE system's structure, defining class hierarchies, methods, and interactions. This ensures the perception module is modular and adheres to Object-Oriented Programming (OOP) principles such as inheritance and polymorphism.



- Development Phase:
  - The project will use C++17 as the main programming language, integrated with CMake for building the project. Unit tests will be created using Google Test, and static code analysis will be performed using clang tidy. The repository will be managed on GitHub, with Continuous Integration (CI) pipelines set up for testing and coverage reports using CodeCov.



- Technologies/Tools to be used :
  - Programming Language : C++17
  - Libraries : OpenCV (For Image Processing), Eigen (For Matrix Operations) and MiDas (For Depth Estimation)
  - Build System : CMake
  - Testing : Google Test for unit testing
  - CI/CD : GitHub Actions, CodeCov for test coverage
- External Dependencies :
  - OpenCV (Image processing and Human Detection) – BSD License(Version 4.4.0 and lower) else Apache 2 License (for 4.5.0 and higher versions)
  - Eigen (Matrix and Vector Operations) - Mozilla Public License 2.0

- MiDas (Monocular Depth Estimation) - MIT license

Component Functionality :

The HDPE system will consist of the following functional features. For **Human Detection,** using a pre-trained deep learning model from OpenCV's DNN module the system will detect humans within the camera frame. For **Depth Estimation** MiDas will be used to estimate the depth of detected humans, providing additional data about their distance from the robot. For **Bounding Box Display,** Visual feedback will be provided by drawing bounding boxes around detected humans in the video feed. For **Pose Estimation,** Pose Coordinates will be estimated for the detected humans in the frame. The coordinates of each human detected will be output in the robot's reference frame, allowing for obstacle avoidance by the path planner.

Potential Risks and Mitigation:

Ensuring the camera feed and human detection system are well-synchronized is critical for maintaining real-time accuracy in detecting human presence and updating their positions. To achieve this, the system must be built efficiently using solid programming techniques that minimize delays and optimize performance. It's also important to build a perception system that remains reliable even when faced with minor disruptions. For example, if the camera's height or position on the robot shifts slightly, the system should still deliver accurate information about the location of detected individuals. To achieve this level of precision, using proven computer vision algorithms and conducting thorough testing with high code coverage is essential. This approach helps guarantee that the system remains stable and dependable, even under variable conditions.

Final Deliverables :

A software which when built and executed takes in feed from a monocular camera and detects the humans in the frame. The software will be implemented in C++17, including unit tests, fully integrated with CI/CD pipelines. Class diagrams and activity diagrams representing the structure of the HDPE system. Unit tests with over 90% code coverage. A README file, providing detailed developer-level documentation, including usage instructions, setup guide, and API reference.

Team Organization / Task Distribution:

We will interchange between the roles of Driver, Navigator and Design Keeper as per the demand of the work. The software development cycle would utilize pair programming using the Agile Iterative Process. Pair programming will be documented through detailed commit messages on GitHub, along with regular code reviews and pull request workflows.

References :

1. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 779–788). https://doi.org/10.1109/CVPR.2016.91
2. Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., & Koltun, V. (2022). Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-Shot Cross-Dataset Transfer. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(3).