ENPM808X Mid-Term Proposal

Aman Sharma', Po-Yu Huang', and Shantanu Parab'

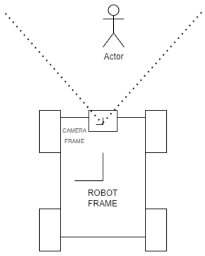
'Professional Master's in Robotics,University of Maryland

October 2022

Overview

ACME Robotics is planning to build an autonomous robot that can maneuver through a room to drop parcels from one location to another specified by a user.However, the room is filled with other human doing their job. The robot has travel through the area without collision with the humans to its designated position.

Figure 1: Frame Representation



Solution

We plan on solving this issue using the monocular camera fitted on the robot. We will be developing a software for detecting humans and tracking them to integrate with the robot. The input stream of images will be using a monocular camera. The obstacle detected will be with respect to the frame of the robot using which the robot can further make decisions to maneuver in the area without colliding with the humans.

Assumptions

: We will be assuming a few things for our implementation as follows.

- The robot and the obstacles are in the same plane.
- As we are using a monocular camera we will be having no information about the depth.To overcome this we will be assuming a ideal height for the obstacles and deriving the depth based on geometric calculations
- We will also be assuming a standard Field of View of camera for ease of calculations
- Images from a standard camera will be used for training and testing.

Implementation

The module will be integrated into the robot in such a way that it can provide the relative position of the human obstacle with respect to the robot's frame of reference. The camera is mounted to the front of the robot.

Development Process

We will be following the **Agile Iterative Process** along with **Test Driven Development**. The team will work in weekly iterations and keep a track of the backlog in a backlog chart. The commits at the end of the day will be built overnight proceeded by a daily meeting to discuss the impediments, reflect on the previous build and decide future deadlines. We will be using Git Version Control to keep a track of the progress. The unit test will be created to check the functionality of the code. We will be using best practices from TDD to minimize the errors and best coding and documentation to provide documentation.

Team

Aman (*Navigator*) will guide the driver to implement the code and check for errors in the code.
Po-Yu Huang (*Driver*) will be developing the code and the test cases.
Shantanu (*Design Keeper*) will be keeping the track of the progress and backlogs along with the documentation of the project.

Tools

We will be using the **OpenCV** library which is open source to transform and process the image input from the camera. The programming language of choice would be **C++** . We will be making use of the **Cmake** build system to build our software.In order to check for memory leaks we will be using **Valgrind**.The **Kalman Filter** approach will be used for tracking.

Algorithm-YOLO

We will be using the YOLO (*You Only Look Once)* algorithm as a part of the software. YOLO algorithm works by dividing the image into N number of grids and for each grid value, we have anchor boxes for detecting and localizing different objects in a single grid. Unlike most of the detectors, YOLO does not produce region proposals but comes up with a number of bounding boxes that can be later suppressed through a non-max suppression and using IOU thresholding. The probabilities for a particular class is included in the predictions along with the bounding boxes. This makes the detection fast and accurate as there is only a single stage involved in the process. There are different versions of this algorithm but for the project, we will be using the most basic one in order to detect the objects. Some more recent versions use a number of techniques for the same improving the accuracy as well as the inferencing speed but the underlying principle for it remains the same.

Risks and Mitigation

Risks

- Poor Quality video.
- Modeling Biases
- Ambient Light
- Incompatible processing and frame speed

Mitigation

- Get proper video quality
- Check for compatible processing
- Have a light setup to illuminate the area under observation
- Get a bigger data set.

Deliverable

We will provide a software that can provide the coordinates of a human obstacle with respect to a robot frame. These coordinates will be based on few assumption defined in the proposal and should be considered as estimates. Precise outcomes can be obtained by changing these assumptions to actual real time parameters. A comprehensive documentation of the software will be provided for easy use and learning.