Human Detector & Tracker – Proposal

Introduction

We propose to develop a robust human obstacle detector and tracker using a monocular camera, directly usable in a robot's reference frame. The deliverables will include but not be limited to, a new module for detection & tracking in C++ using high-quality software engineering practices, class and activity diagrams, an up-to-date GitHub repository with complete documentation and unit tests integrated with Travis CI and Coveralls.

Project Organization

Product development shall be done using the Agile Iterative Process where tasks will be tracked using a backlog table. All the tasks will be outlined and backlog tables for each iteration will also be maintained. The entire project will split into two sprints. Team members shall follow the pair programming method for development, where the roles of navigator & driver will be interchanged every sprint.

The release model will follow a rapid prototype development process and the testing model to be used will be unit testing where several unit tests will be designed for complete code coverage.

Managerial Process

The aim of the management is to create a quality deliverable with no bugs. Risk mitigation for the human detector will be performed by testing it on curated data from the COCO dataset. A fallback plan of using a Viola Jones face detector and dilating the output box in order to hopefully capture the human, is in place.

Technical Process

The project will be developed using C++. Additionally, for image operations OpenCV will be used. A YOLOv3 neural network trained on the COCO dataset will be used as a human detector and a tracker.

The camera is considered to be at a fixed position on the robot. A transformation matrix ${}^{R}T_{C}$ relates the body frame of the robot to the camera frame [1]. The X coordinates of the object to the camera will be calculated by using a one-time calibration which will calculate the approximate focal length of the camera. During the calibration of the camera, the pixel to meter conversion factor is also noted. The bounding box outputs from the YOLOv3 model will be converted to SI units by using the constants of calibration. The resulting detections will be multiplied with the ${}^{C}T_{R}$ to output co-ordinates in the robot's body frame.

Complete documentation will be maintained for each component, with proper code commenting. Doxygen will be used to generate relevant documentation and cpplint and cppcheck will be used for style sheet adherence and static code analysis.

For the estimation of depth, the human is assumed to be upright and not occluded as calibration is to be done in full upright conditions. Lighting is also required to be fairly consistent.

Appendix

References

[1] Ma, Lu & Ghafarianzadeh, Mahsa & Coleman, Dave & Correll, Nikolaus & Sibley, Gabe. (2014). Simultaneous Localization, Mapping, and Manipulation for Unsupervised Object Discovery. Proceedings - IEEE International Conference on Robotics and Automation. 2015. 10.1109/ICRA.2015.7139365.