Monocular Human Detection using YOLO

Proposed By:
Navdeep Singh (120098024)
Sachin Jadhav (1194845224)

Software Plan

The system will comprise two interlinked components: the human detector and the human tracker. The human detector will leverage a pre-trained YOLO model to scan each video frame and identify the presence of humans, generating bounding boxes that provide spatial information about their locations.

The human tracker will then be responsible for maintaining a continuous understanding of each human's movement and position relative to the robot's camera frame. This setup assumes the robot's camera is positioned at a fixed height, facilitating accurate tracking and interaction with individuals in its environment.

The human detection component will utilize a pre-trained open-source YOLO convolutional neural network (CNN) model, converted to ONNX format, for identifying humans in the input frames. For tracking and localization, the system will leverage geometric computer vision techniques, which are efficient and reliable for this purpose. YOLO is selected due to its status as a benchmark in human detection, where the trade-off between slightly higher computational cost and superior performance is justifiable. In contrast, for human tracking, classical computer vision approaches provide a stable solution without introducing unnecessary complexity or computational overhead.

Tools to be Used

The key tools and technologies employed for developing the system are:

- **C++**: Main programming language for high-performance processing.
- **OpenCV**: Utilized for geometric computer vision tasks (licensed under Apache 2.0).
- **ONNX**: For integrating the pre-trained YOLO model (licensed under Apache 2.0).
- **CMake**: Build system for managing project compilation and dependencies.

Definitions and Acronyms

YOLO (You Only Look Once): A state-of-the-art, real-time object detection system based on deep learning.

ONNX (Open Neural Network Exchange): An open format to represent machine learning models, facilitating the integration of pre-trained models across different frameworks.

NMS (Non-Maximum Suppression): A technique used to filter out redundant bounding boxes during the detection process, ensuring only the most confident predictions are kept.

CNN (Convolutional Neural Network): A class of deep neural networks widely used for analyzing visual data.

AIP (Agile Iterative Process): A development methodology characterized by iterative and incremental delivery of software features, enabling continuous improvement and adaptation.

Cost

The primary costs associated with the project include computational resources for training and testing the detection and tracking models, as well as the labor associated with development, testing, and documentation. Since open-source libraries and pre-trained models are utilized, software costs will be minimized. However, additional expenses may arise if specialized hardware (e.g., GPUs) is required to optimize real-time performance on the robot.

Development Process and Organisation

The software development process will follow the Agile Iterative Process (AIP) with a pair programming approach. Sachin Jadhav will focus on developing the human detection component, while Navdeep Singh will concentrate on the human tracking system. To ensure quality, Sachin will be responsible for writing the unit tests, and Navdeep will execute these tests to validate the functionality. This sequence is chosen because the tracking system provides the final output of the perception stack. Both Sachin and Navdeep will contribute equally to the documentation for their respective components.

References

YOLOv5 Official Repository
Ultralytics. (n.d.). YOLOv5. Retrieved from https://github.com/ultralytics/yolov5
OpenCV Official Website
OpenCV. (n.d.). Open Source Computer Vision Library. Retrieved from https://opencv.org