Diane Ngo (UID: 117554960) Dani Lerner (UID: 114054883)

ACME Perception Proposal

Introduction

The project chosen for Acme Robotics' module is the Perception task, where a human obstacle detector and tracker will be implemented. Perception in robotics is of huge importance as it can heavily influence the safety of the robot and its surroundings. With cutting edge technology, a neural network will be combined with image segmentation in order to achieve the task of detecting humans with high accuracy. Generally, the perception works with a camera, providing the robot a vision system. Because it is a type of sensor, the feed is sent to the controller so the robot can quickly stop if something has been detected in its way. For easy use, the camera will be a front view camera. In our implementation, the proposed idea is a robot would be pathing to designated areas in a warehouse, so a vision system is relevant for a dynamically changing environment. This will be greatly beneficial to Acme Robotics, as they can implement the vision system for any of the robots in any situation where humans are working alongside or in the same area as the robots.

Development and Design Process

The development process that will be used is the Agile Iterative Process (AIP). To start the design process, an idea must be in place in order to start setting up requirements. While thinking of the requirements, we have to carefully consider feasible plans, budget, safety, and any constraints or needs that Acme Robotics wants. For this project, budget is not considered in the design. The biggest concern to consider for a requirement is the accuracy of human obstacle detection. Ideally, at max accuracy the robot will always detect anything in its way to prevent any accidents and collisions from happening. After drafting the initial requirements, then the next step is to create the product backlog. A sprint backlog is also to be created to allocate proper time for each task. Daily standup meetings will be in place as well in order to keep communication at its highest, and to keep the team on track.

Deliverables and Milestones

The deliverables for this project include a URL to the github repository, a 3 minute video presentation of the design and methodology, and a QuadChart of the project. This can be considered as the requirements from Acme Robotics. Project milestones are needed in order to have a good idea of the progress updates of the work. Such milestones are: getting the training and test data, designing the baseline classes in UML format, deciding the build system, determining which neural network to use, training the network, and reaching a desired accuracy. The success criteria for a perception based project is the desired accuracy that the neural network should achieve on test data. This accuracy is determined to be at least 90%.

The risks for this project can be split into two categories: project and technical. Technical risks correspond to the development process while project risks correspond to the finished product. The project risk is an inaccurate human detection module. False positives will cause errant motion and inefficient — or even ineffective — implementation of the tasks while false

negatives could potentially bring harm to human workers. The technical risks are more subtle. The first, diverging software development, occurs when communication and planning between the programmers is insufficient. This will cause large wastes in the programmers time since they must merge the paths and may require a reconstruction of their architecture. The second is the absence of any success criteria. This is again caused by a lack of planning and the programmers may waste a lot of time to meet a goal they have not met or, they might have spent too much time initially to meet a higher goal than the system required.

Software and Libraries

When human and robot workers share a space, software precautions must be taken to ensure no humans are injured during operation. Therefore, our vision system must be able to detect humans at high rates and prevent injury from occurring. For this reason, we will use C++ as our primary language. Scripting languages — such as python — may yield simpler implementations but have significant performance overhead; meanwhile, low-level languages — such as assembly — may provide optimal performance but would be impossible to code in a short span of time. We believe C++ is ideal since it balances high performance capabilities with high-level libraries we can use to implement the task at hand.

There are a few libraries we know we will need for this task and a few others we can only assume will be necessary. We cannot guarantee that the following description will contain every library we use but we will attempt to produce a comprehensive list. First and foremost, this is a computer vision project and we will need OpenCV to handle much of the image processing. It is a powerful open source library containing computer vision implementations. Second, we will need a machine learning library to handle human detection. TensorFlow is a powerful ML library and contains pre-built models for detecting humans but we must ensure that these models are robust for our given application.

Since an image is nothing more than a matrix of pixels, we predict we will need the Eigen library to perform matrix math to develop this perception software. Lastly, we are certain we will need standard libraries but we are unsure which of them we will need. We will most likely use vectors, strings, and iosteam. We may also decide to use other data structures (e.g. queues and unordered maps) or other libraries such as stdexcept. It is very possible that we will need other libraries in this project but it is hard to know until we start the software development.

Pair Programming Execution

We believe that the most efficient programming procedure uses a software engineering process called “Pair Programming.” In this exercise, the driver physically types in the code while the navigator looks on, ensuring a lack of bugs as well as code smells. While this consumes the time of two programmers, it combines the knowledge of both and yields better code quality. It also saves time in the long run since the presence of bugs will be less likely. We believe that there are two main tasks: creating the glue code for the human detection algorithm and testing. We will select which driver corresponds to which task based on our relative strengths while the second person shall be the navigator.