Robot driving itself into a van

Report Name	Project Outline
Author (User Id)	Matt Morgan (mam107)
Supervisor	Fred Labrosse (ffl)
Module	CS39440
Degree Scheme	GH76 (Artificial Intelligence and Robotics)
Date	February 4, 2019
Revision	1.0
Status	Release

Project description

A van is used to transport Aberystwyth universities large robots to and from test sites and these robots currently have to be manually driven into and out of the van for transportation. The Robot driving itself into a van project is an application that will be used to automate and streamline the process of setting up large robots for field trials by loading and unloading robots into and out of the transport van.

This project will use a forward facing LIDAR to guide the robot into and out of the van. This LIDAR should see a specific succession of patterns when driving into or out of the van and these patterns can be used to control the robot to drive the correct path into and out of the van.

The application will be designed and tested in a simulator before it is used on real robots. This is to make sure the real robots don’t get damaged while the application is still in development.

The end goal is for any large robot with a LIDAR to be able to successfully load itself into and out of a van at the push of a button. Currently, the scope is for the robot to already be at the back of the van and roughly aligned so it is ready to be driven into the van. This scope can be expanded upon so the robot doesn’t have to be at the back of the van or even near it.

Proposed tasks

There are a few **tools that need to be looked into**. One of these is the use of ROS[1] as this API is what Aberystwyth university uses to control its robots. Another tool required is a simulator. ROS has 2 main simulators, Stage[2] which is used for 2D simulations and Gazebo[3] which is used for 3D simulations. Judging by the fact that these large robots will have to be used in the real world which is 3D, Gazebo will have to be used and looked into.

The test environment(s) within the simulator need to be created before any development or testing can be started. The SDF model for the robot is provided but the van model needs to be created and both need to be put into an empty world.

Certain **Computer vision techniques** such as detecting sharp features[4] will need to be investigated to determine the container’s location relative to the robots location so the robot can adjust its trajectory when loading or unloading from the van.

Thorough testing needs to be conducted whilst the application is being developed due to the application running on real world heavy machinery. This testing will be conducted within the simulator to find and fix bugs because testing on a real robot could cause unintended physical damage.

The **Software environments** will need to be set up so that the ROS workspace works correctly with a custom directory. Online version control such as github[5] or svn[6] also needs to be used so there is a backup of all work done at all times.

Preparation for the two demonstrations in the form of notes, testing any live demonstrations and practising the demonstration before doing it. This preparation should result in both demonstrations going smoothly.

Keeping weekly notes for what I am working on during the week and for what is discussed in meetings. This includes anything methods attempted which are unsuccessful as well as successful methods.

Project deliverables

Investigation into pointcloud CV techniques: A report investigating computer vision techniques that translate well to raw pointcloud data will be produced so the robot has an accurate way of determining how it should move to successfully drive into the van.

Mid project demonstration: A live demonstration in simulation will be produced to show the functionality of the application. Notes for this demonstration will also be produced.

Final program: The final program which will be able to drive any of Aberystwyth university’s large robots that have a LIDAR attached up a ramp and into the van. It will also be able to unload any of these robots from the van.

Final demonstration: A live demonstration with the real robot loading and unloading from the van will be preformed. This will showcase how well the project went and how stable the application is.

Final report: A report detailing how the program works, the process followed to complete the program and anything attempted to get the separate parts of the program to work will be produced. This report needs to be very detailed as it will summarise 3 months of working on this project.

Initial annotated bibliography

[1] Various, Open Source Robotics Foundation. (Last updated: 29/11/2018) The ROS online documentation website. [online] Available at: https://wiki.ros.org/ [Accessed 04/02/2019]. License: Creative commons attribution 3.0.
This website gives all the official documentation for the open source robot operating system that the university uses to control the majority of their robots.

[2] William Woodall, with contributions from many others. (Last updated: 18/09/2018) The stage simulator package on the ROS website [online] Available at: https://wiki.ros.org/stage [Accessed 04/02/2019]. License: GNU General Public License version 2.
This webpage details the package details for the stage 2D simulator used in ROS. It also contains links for detailed documentation and API usage.

[3] Nate Koenig, Andrew Howard, with contributions from many others. (Last updated: 05/11/2018) The gazebo simulator website for ROS [online] Available at: http://gazebosim.org/ [Accessed 04/02/2019]. License: Apache 2.0.
This website is the homepage for the 3D simulator for ROS. I will be using this simulator for development and testing of the application.

[4] Christopher Weber, Stefanie Hahmann and Hans Hagen. (Published: 2010) Sharp Feature Detection in Point Clouds [online] Available at: http://ljk.imag.fr/membres/Stefanie.Hahmann/PUBLICATIONS/WHH10small.pdf [Accessed 06/02/2019].
This is a paper on detecting sharp features (such as edges of a container) with point cloud data. The techniques discussed in this paper can be used to locate the van's position in relation to the robots position.

[5] GitHub, Inc. (Last updated: 2019) Github website for online version control and code collaboration [online] Available at: https://github.com/ [Accessed 06/02/2019].
This website is for the git version control system. Git repositories are hosted at github.com online and can be accessed with any computer that has an internet connection.

[6] The Apache Software Foundation (Last updated 18/01/2019) Subversion version control system. [online] Available at: https://subversion.apache.org/ [Accessed 06/02/2019] License: Apache 2.0.
A version control system similar to git. Online hosting usually costs money unlike with using github or gitlab.