Visual Servoing

In this laboratory exercise, you will control the motion of your robot based on visual feedback from your Zed camera. This technique is called "Visual Servoing" which typically classifies into two categories: Image Based (IBVS) and Position Based (PBVS). This lab has the following objectives:

1. Identify a bright orange cone with the zed camera using your own object detector
2. Implement IBVS to park your robot in front of a cone

Looking Ahead: In lab 4B you will use OpenCV with ROS for more sophisticated image detection algorithms and implement PBVS.

1 Detect a Cone

Objective: Write an image detector to detect an orange cone.

1.1 Launching Zed Camera

Launch the zed camera:

```
roslaunch zed_wrapper zed_depth.launch
```

Verify you can see the image from the zed camera. Using rqt_image_view and select the following topic from the drop down menu:

```
"/camera/zed/rgb/image_rect_color"
```

If you cannot view the image please refer to the steps for streaming data in Lab 2B.

1.2 Accessing Image Data

Write a ros subscriber for zed camera topic:

```
"/camera/zed/rgb/image_rect_color"
```

The zed camera publishes message of type Image from sensor_msgs. Learn about this message type by opening the file Image.msg located in

```
roscd sensor_msgs/msg
```

Image.msg has the information about how to access the pixels of the image.

1.3 Detect the cone

Use your favorite ros language to write a subscriber for the zed camera messages. Refer to image.msg to access the pixels of the image and create a detector to determine the location of an orange cone in an image.

2 Image Based Visual Servoing (IBVS)

In this part of the lab you will park your robot in front of an orange cone. The robot will have the cone in its field of vision and should drive directly to the cone and park in front of it. You will implement this using image based visual servoing (IBVS). In IBVS, the control law is based on the error between current and desired features on the image plane, and *does not involve any estimate of the pose of the target.*
Since the zed camera is mounted symmetrically on the robot we want the cone to show up in the center of the image it captures.

From Lab 1 we saw we can control the racecar using teleop.launch and a rostopic publisher. The commands are repeated here for your convenience:

```
roslaunch racecar teleop.launch

rostopic pub /vesc/ackermann_cmd_mux/input/navigation
ackermann_msgs/AckermannDriveStamped
'{header: auto, drive: {steering_angle: .2, speed: 1.2} }'
```

2.1 Combine Object Detector and Motor Commands

Use the object detector created in part 1.3 to determine which direction the robot should move to center itself in front of the cone. Combine this with the teleop commands to drive the robot to the cone.
The robot should not run over the cone! Combine some of the features from the previous lab to make sure it parks in front of the cone.