# Park in Front of the Green Marker

## Objectives

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).

Specifically, you will use the ZED camera to detect a green marker in front of the vehicle. Then you will design a control system that parks it in front of the green marker.

## Deliverables

- ❑ Use the ZED camera to capture video in the ROS environment, and detect all green blobs, find out their centers.
- ❑ Develop a control system that will enable the car to park in front of the blob.

## Challenge Problems

- ❑ Park 1ft to the right or left of the marker
- ❑ Make your car do line following

# Green blob detection with image position

Augment your blob detection from yesterday's lab to detect the green marker. Publish the position of the marker relative to the image width and height using the message detection format from Tuesday's afternoon lab.

Implement and test your blob detector by using rosbag. Make sure the performance is fast enough to work in realtime motor commands for the next section of the lab.

# Control Systems for Visual Servoing

The objective is to park your robot in front of a green marker. The robot will start with the marker in its field of vision and should drive directly to the marker and park in front of it (5-3 inches from the front). You will implement this using image based visual servoing (IBVS) and solely on zed camera (no laser scan data!). 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 marker to show up in the center of the image it captures.

## Error Signal

First augment your blob detector to publish an error signal that's normalized to [-.5, .5]. You can do this by finding the horizontal center of your blob and dividing it by the width of the image.

$$e(t) = 0.5 - Xc(t) \div ImageWidth$$

Where $Xc(t)$ is the horizontal center of the blob *in pixels* at time $t$ and $Imagewidth$ is the width of the *image in pixels*.

Test out your error signal with the rosbags you collected earlier. Verify the signal is near 0 when the blob is in the center of the screen and positive when the blob is to the right.

## Proportional control

We can use the feedback control methods we discussed last week to drive our error signal to zero, and thus bringing the blob directly in front of the car.

Recall, the expression for a proportional controller is:

$$u(t) = Ke(t)$$

Where $K$ is the proportional gain parameter and $u(t)$ is the output of the controller at time $t$.

## Parking the car

In addition to driving your car to the green marker, you need to make sure you don't drive over it. You're welcome to find a solution using your blob detector (like the size of the contours) or use the LIDAR.

## More Object Detection with OpenCV

Once you complete the parking task you’re encouraged to explore different object detection algorithms already implemented with OpenCV.