Computer Vision and Lane Segmentation in Autonomous Vehicles

Joseph Fallon jdfallon@ucsd.edu
Evan Kim mik213@ucsd.edu
Ka Ming Chan kmc025@ucsd.edu

ABSTRACT

Perception is absolutely vital to navigation. Without perception, any corporeal entity cannot localize itself in its environment and move around obstacles. An entity must be able to detect objects, interpret depth, as well as size, and move accordingly. While these processes may seem mundane, since they are inherently given in most biological entities, these processes are deeply complex. Replicating continuous object detection, calculating depths, and actuating movement is no easy task. All of these tasks stem from perception. In autonomous vehicle applications, the vehicle uses a stereo camera, equipped with two lenses, to mimic the binocular attributes of the human eye. With this camera, the vehicle can achieve all of these tasks, however the performance of the object detection and localization heavily relies on the capabilities of the stereo camera.

To create an autonomous vehicle (AV) capable of racing on the Thunderhill Raceway track in Berkeley, California, the team must supplement a stereo camera capable of supporting image perception and computer vision. The team was given two cameras, the Intel RealSense D455 and the ZED camera. Both utilize binocular vision and infrared sensors to be able to calculate the depth of the object in their field of view and feed the SBC on the AV. In this analysis, the team will compare the capabilities of the two cameras and responsibly select a camera capable of supporting the object detection. Additionally, to contribute to DSC 180B's collaborative AV, the team will develop a lane segmentation algorithm that will help extract lanes from the camera feed.

INTRODUCTION

From the effort of last quarter, the team has successfully assembled and developed an robot car that is capable of running on a designated racing track

autonomously. The team has accomplished this goal by implementing Robot Operating System (ROS) code with Python, and mainly using the methods of perception and computer vision. While having the autonomous car running on itself is achievable in a designated track inside the UCSD tent, the team's goal for this quarter is to reinforce in making this mission more robust, and ultimately make it possible to race on a real track representing the one at the Thunder Raceway. To contribute to this mission, one of the team's tasks is to figure out the better choice between the two stereo cameras, ZED and Intel RealSense D455, which are both available from UCSD's lab.

Why is it important to optimize the choice of camera equipped for the autonomous vehicle? In summary, the method of the team's AV for finding directions relies heavily on the ability of perceiving and detecting the lane. Using ROS data perception, computer vision image processing, and machine learning, the AV "sees" its way on the track and navigates. By data, it refers to the images captured by the equipped camera on the vehicle. Since the images are in a continuous flow and in an open environment as the vehicle is driving, the camera is expected to operate at high frequencies, provide good enough resolutions, and capture adequate distances. Mainly, these expectations are to be fulfilled on an open environment race lane, all contributing in the effort for the lane detection process, as well as in the machine learning algorithm. Lane detection is one of the crucial elements which drives the car, and the performance of lane detection is one the team's major research.

To evaluate the performances of the stereo cameras provided and compare them in terms of fulfilling the listed expectations, the team sets out to develop a set of methods and experiments for doing so. Operating at high frequencies refers to the camera's ability of capturing images at a high frame rate per second (FPS), and is essential because any delay of the data during autonomous driving leads to mistakes in navigating. The image data needs to be fast and smooth, even for sacrificing the resolutions. While using low resolutions, it is still expected that the cameras can be seeing relatively clearly, far, and enough. Also, as the Thunderhill track is an outdoor track, the camera needs to be adapting to the open environment setting, especially to the lighting condition. Since, known by the progress of the teams so far, outdoor lighting

affects the perception of the cameras heavily.

The methods of testing the cameras under these expectations for accomplishing lane detection are listed in the methodology section. Learning about the FPS, contesting on distances and lightings by experimenting on depth and environment robustness, as well as applying the filter algorithms for edge (lane) detection, will help the team on researching computer vision theories and coming up with the conclusion of the camera choice between ZED and Intel RealSense D455.

Each camera is set to be using the most optimal FPS mode. The ZED camera is using 100 FPS at VGA 376p, and the Intel RealSense D455 is using 60 FPS at 480p.

METHODOLOGY

The constraints of a racing application are very different from most AV applications. The environment is high speed, outdoors, and has varying lighting conditions. A responsible choice of camera will need to fit the constraints of the raceway.

- **High frames per second (FPS):** Given the race application, the camera must be able to feed the image processing with a lot of images. More images, means more information and in a high speed environment, more information can support the reactive times of the AV.
- **Depth:** The Thunderhill Raceway is immense, with a track length of up to 5 miles. The camera will need to be able to detect objects far from itself to anticipate turns.
- **Environment Robustness:** A track will not always produce the most ideal situations. A responsible camera choice must be able to perform well in many different environments. From low-light to high-light, indoors and outdoors, and everywhere in between, the camera needs to be able to feed the SBC with high quality images.
- **Filter testing:** While people can look at two images and discern which is high quality, the computer perceives images in a much different manner. To see the performance in a more computer based sense, the team will feed similar images into various

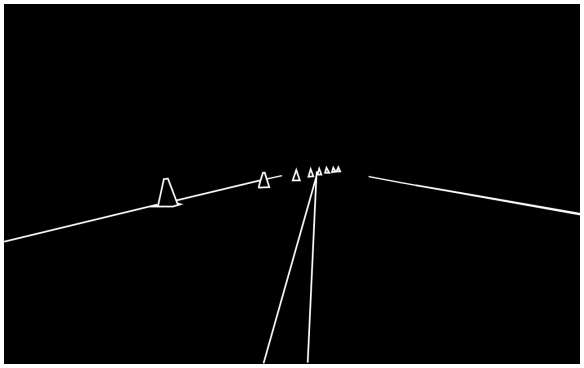
filters to see how the computer performs between the two cameras.

Evaluation: To evaluate the cameras,the team needs to understand how they perform in a computer vision setting. Simply looking at an image and qualitatively deciding which looks better will not suffice for the team's analysis. If the team wants to truly support computer vision,the team needs to understand how the computer interprets these images and how they are processed into informative metrics. In the team's application, the team will solely focus on the RBG images supplied to the SBC at the lowest resolution to ensure a high FPS. Recall that the team wants to compare the Intel RealSense D455 vs. The ZED stereo camera. To evaluate the images, the team has chosen to use the Structural Similarity Index Measure (SSIM). The SSIM measures the similarity between two images without knowing which is the better quality image. It calculates the similarities between the five filter-transformed images and the ground truth, providing five SSIMs for each filtered image that can be used to evaluate the performances for each of the filters. SSIM compares two images on the following attributes: the luminance, contrast, and structure of the images. Given that all the images generated by the image processing and the ground truths are black and white, the luminosity and contrast will contribute almost nothing to the SSIM metric, thus focusing more so on structure. The higher the metric is, the better the filter as it is more similar to the ground truth in terms of successfully detecting the lanes on the track. Generated images are compared against their hand-made ground truth images. The ground truth images represent the traced lines of relevant objects in the vehicle's path and were made in Adobe Photoshop byt manually tracing the edges of relevant features in the images. The team will then generate images from five different filters: Sobel X Gradient, Canny, Prewit, Roberts, and Laplacian of Gausian filters. The generated images are first converted into a single channel grayscale image and a Guassian blur is applied to follow the conventions of most autonomous vehicle applications. After the similar images are generated for each camera, each filtered image is compared to the ground truth and the SSIM is calculated. With this metric, the team can decide which filter on which camera is better.

Depth Test: Depth is an important aspect to creating a proper object detection algorithm.

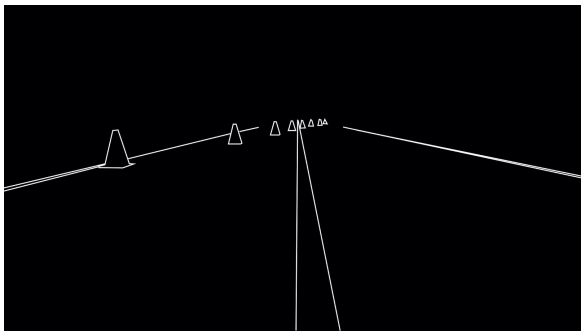
Knowing how far the vehicle can see till can help the team prepare for track applications. A responsible choice for camera selection would be able to see and detect objects roughly forty feet in front of the vehicle. To replicate this, the team set up cones in an environment representative of the Thunderhill Raceway. The cones were placed five feet apart and images were captured from both the Intel RealSense D455 and the Stereo Lab ZED camera. Ground truth images were then generated and the images were pumped through the same pipeline to gather SSIM metrics from the two cameras. By comparing the quantitative metrics of both cameras, the team is able to differentiate which camera would best support the object detection of the computer vision. Below are the images the team generated between both cameras and their respective ground truth images.

Intel RealSense D455 depth images at 480p:



	SSIM
Laplacian	0.388268
SobelX	0.583227
Canny	0.820720
Prewitt	0.488385
Roberts	0.303899

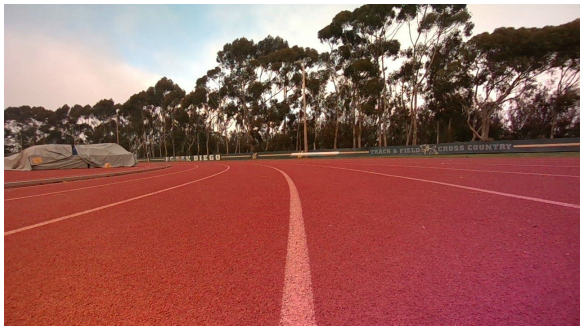
ZED Camera depth images at VGA:

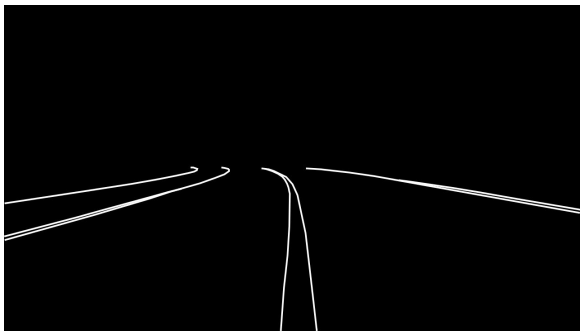


	SSIM
Laplacian	0.568469
SobelX	0.549586
Canny	0.827884
Prewitt	0.474871
Roberts	0.331807

Curvature Test: Curves are abundant in a track application. The team generated images that represent the vehicle approaching a turn with the goal of observing which camera was more robust in a turning setting. Once again, the images were pumped into the same SSIM pipeline and the output metrics will be compared to see which camera has better performance on the turns. Below are the images the team generated between both cameras accompanied by their respective ground truth images.

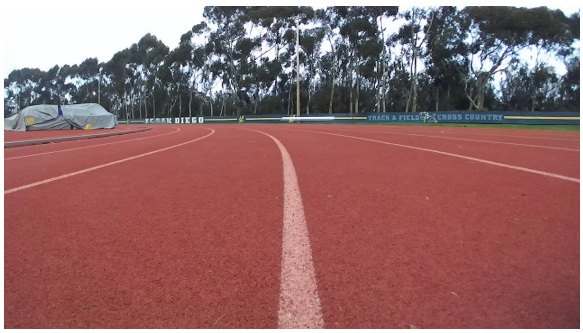
Intel RealSense D455 curvature images at 480p:

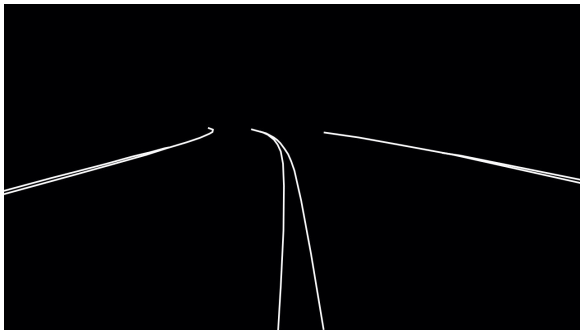




	SSIM
Laplacian	0.418455
SobelX	0.552177
Canny	0.757025
Prewitt	0.453530
Roberts	0.259245

ZED Camera curvature images at VGA:



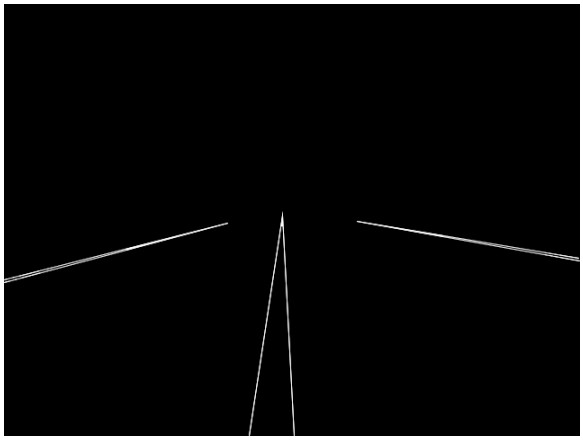


	SSIM
Laplacian	0.580549
SobelX	0.627617
Canny	0.765439
Prewitt	0.514492
Roberts	0.373033

Environment Robustness Test: Conditions vary in a racing application and a responsible stereo camera choice will be robust to changes in lighting conditions. The team generated images both in a high light setting and a low light setting between both cameras with the goal of evaluating the performance between the two. Ground truth images were generated and the images are filtered similar to the previous tests. Below are the images the team generated between both cameras, again, with their respective ground truth images.

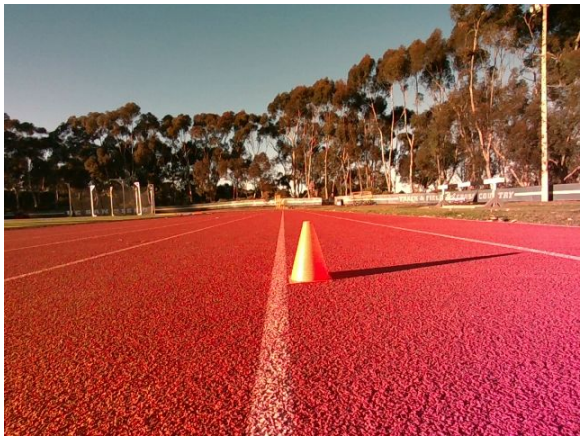
Intel RealSense D455 highlight images at 480p:

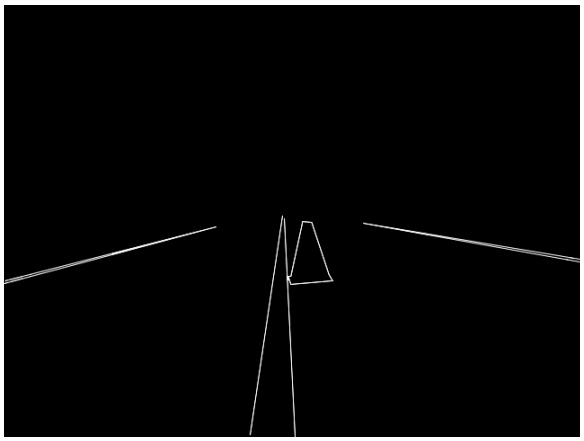
No Objects



	SSIM
Laplacian	0.297728
SobelX	0.402596
Canny	0.563127
Prewitt	0.338706
Roberts	0.186086

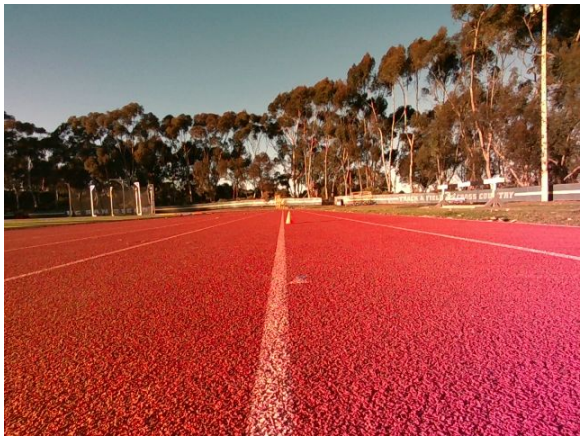
1 meter

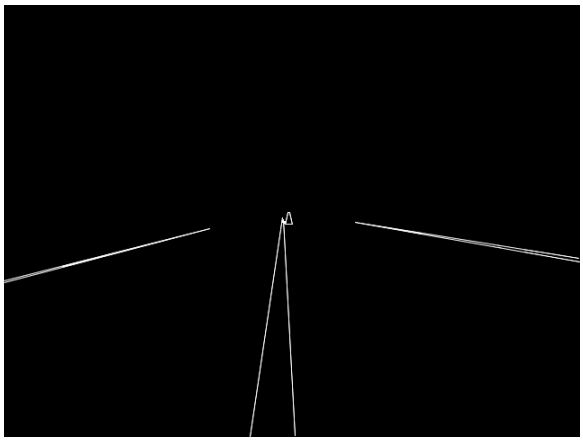




	SSIM
Laplacian	0.287047
SobelX	0.398361
Canny	0.563850
Prewitt	0.335272
Roberts	0.182313

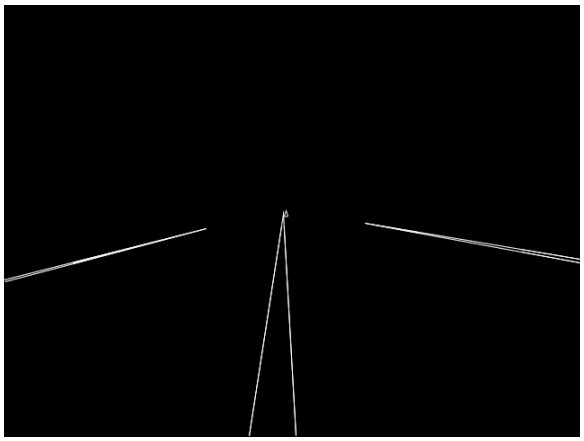
5 meter





	SSIM
Laplacian	0.288895
SobelX	0.409878
Canny	0.568324
Prewitt	0.337917
Roberts	0.185895

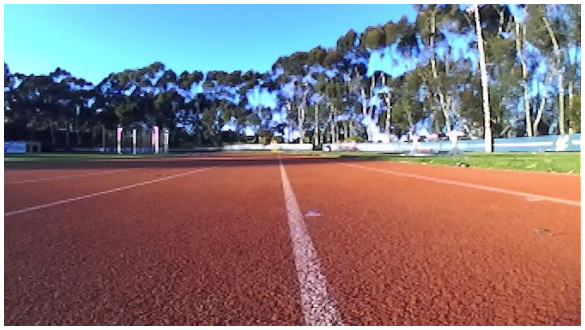
10 meter

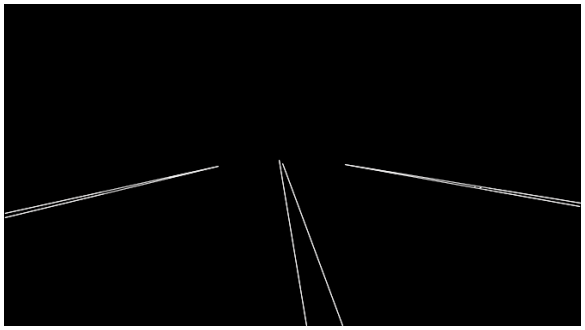


	SSIM
Laplacian	0.279548
SobelX	0.403381
Canny	0.564125
Prewitt	0.330421
Roberts	0.181130

ZED Camera at highlight images VGA:

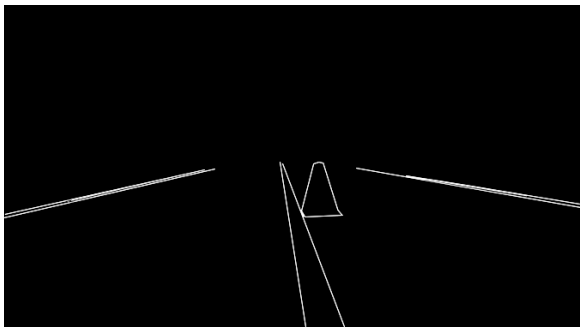
No Objects





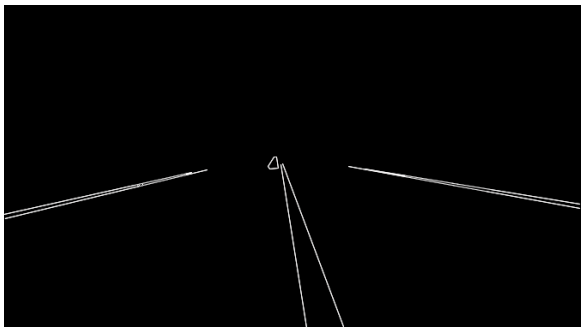
	SSIM
Laplacian	0.402304
SobelX	0.492320
Canny	0.647027
Prewitt	0.374422
Roberts	0.253933

1 meter



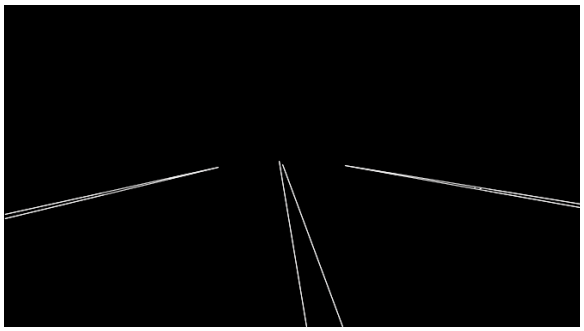
	SSIM
Laplacian	0.499782
SobelX	0.566120
Canny	0.646505
Prewitt	0.449354
Roberts	0.325030

5 meter



	SSIM
Laplacian	0.526799
SobelX	0.594729
Canny	0.641806
Prewitt	0.469797
Roberts	0.369391

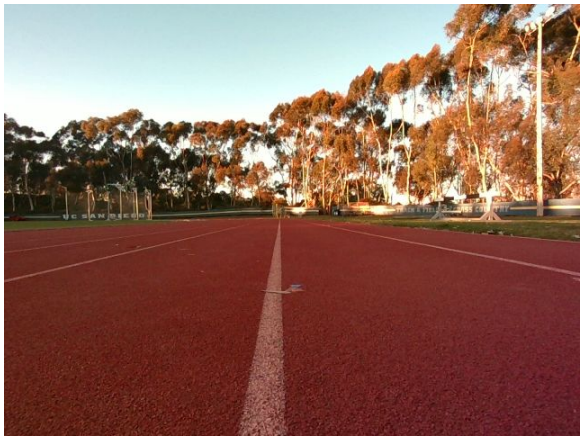
10 meter

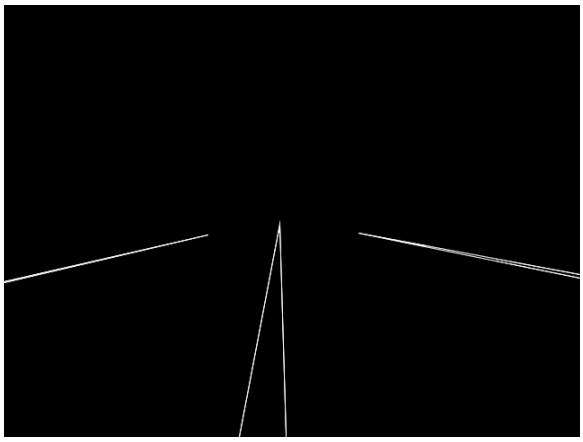


	SSIM
Laplacian	0.550224
SobelX	0.618143
Canny	0.650845
Prewitt	0.492314
Roberts	0.419085

Intel RealSense D455 low light images at 480p:

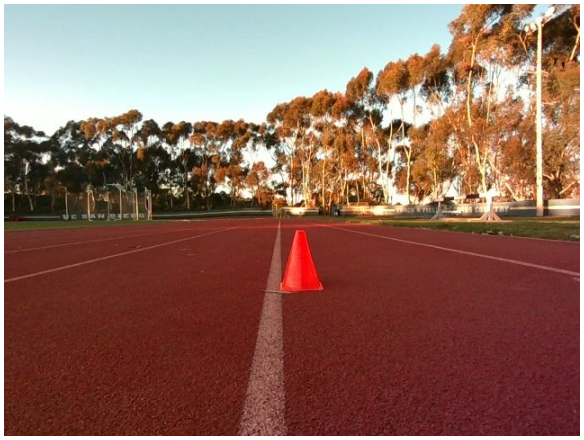
No Objects

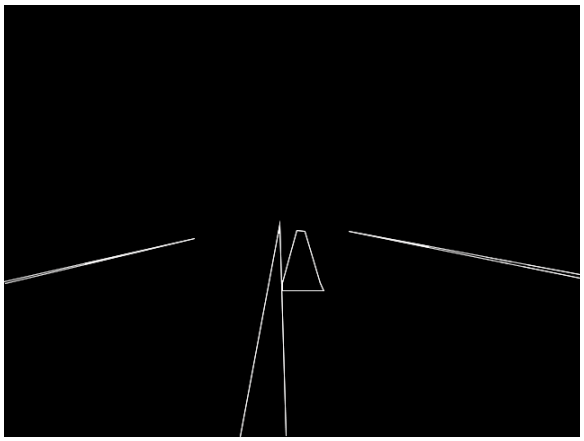




	SSIM
Laplacian	0.489122
SobelX	0.596428
Canny	0.702509
Prewitt	0.527260
Roberts	0.346840

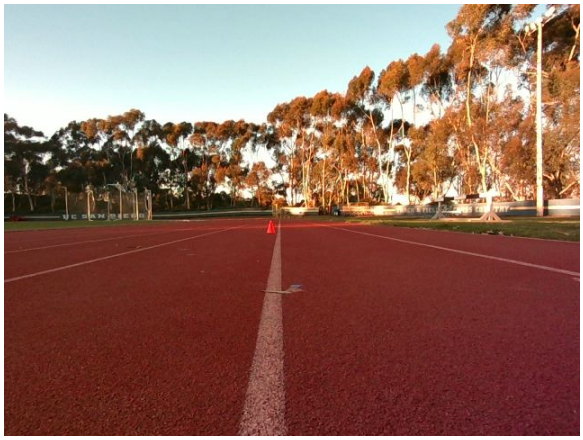
1 meter

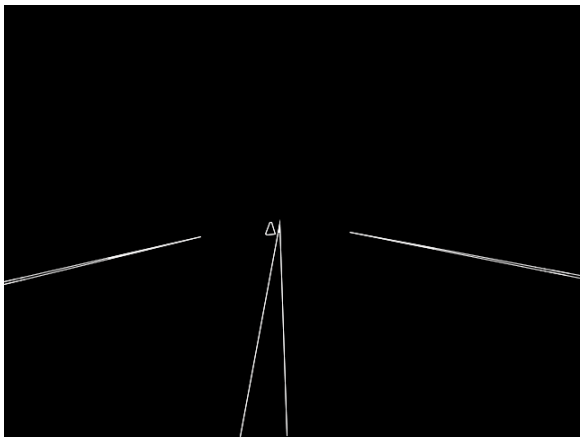




	SSIM
Laplacian	0.487530
SobelX	0.589255
Canny	0.696626
Prewitt	0.528938
Roberts	0.339650

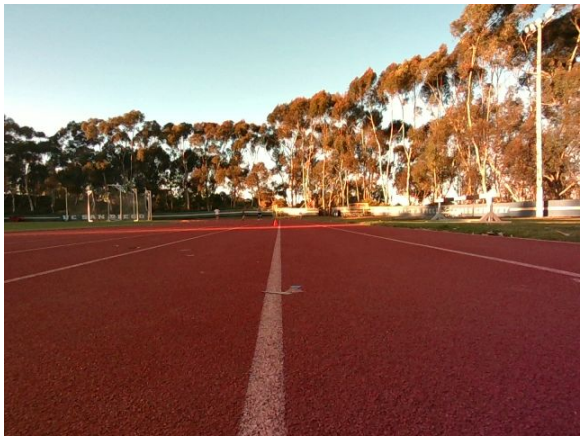
5 meter

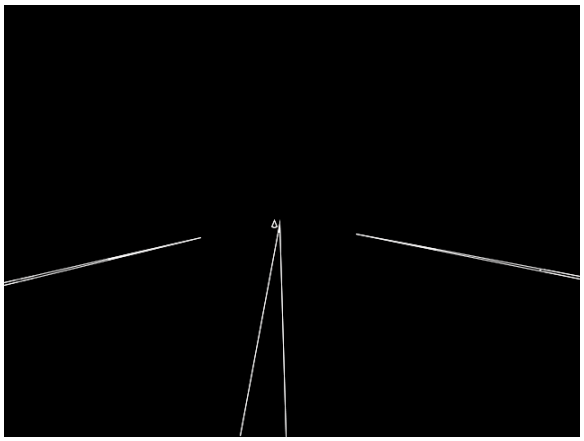




	SSIM
Laplacian	0.501094
SobelX	0.594541
Canny	0.699006
Prewitt	0.533413
Roberts	0.343776

10 meter

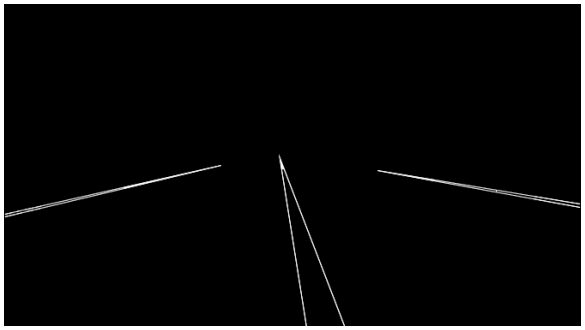




	SSIM
Laplacian	0.478442
SobelX	0.593722
Canny	0.692890
Prewitt	0.535097
Roberts	0.346597

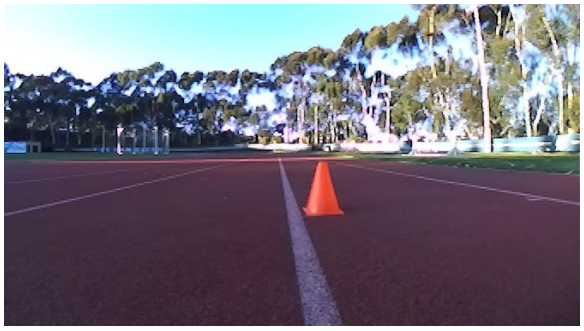
ZED Camera at low light images VGA:

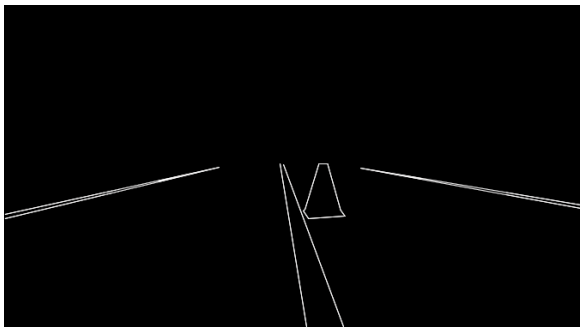
No Objects



	SSIM
Laplacian	0.593851
SobelX	0.631943
Canny	0.670276
Prewitt	0.590860
Roberts	0.502557

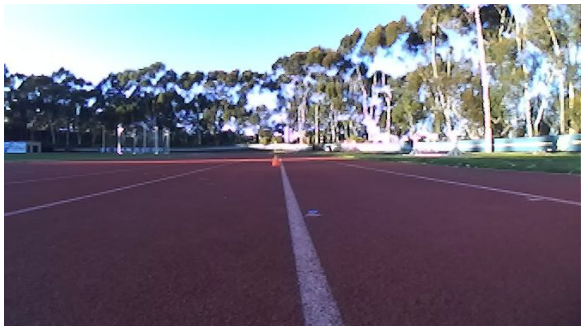
1 meter

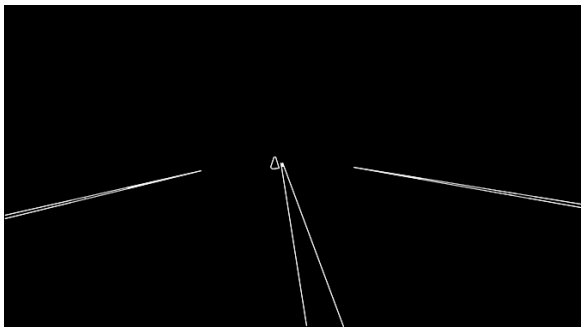




	SSIM
Laplacian	0.565016
SobelX	0.617000
Canny	0.651998
Prewitt	0.568563
Roberts	0.480084

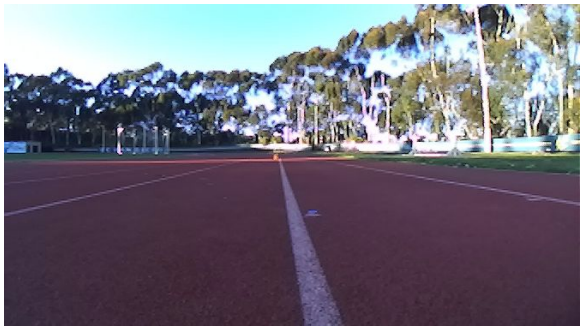
5 meter

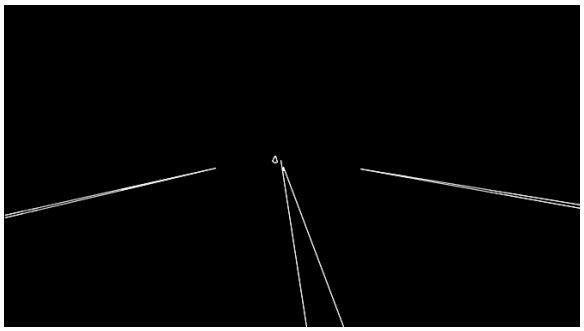




	SSIM
Laplacian	0.571155
SobelX	0.621855
Canny	0.657152
Prewitt	0.576451
Roberts	0.484832

10 meter

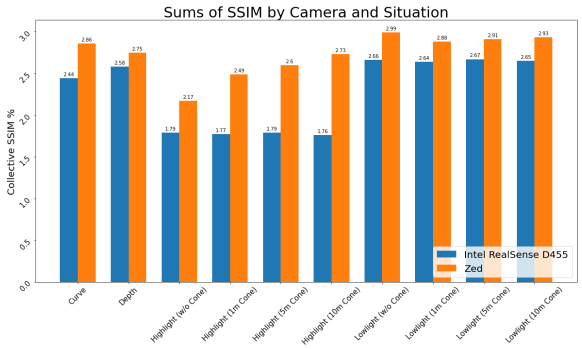




	SSIM
Laplacian	0.574924
SobelX	0.622643
Canny	0.660440
Prewitt	0.581854
Roberts	0.490235

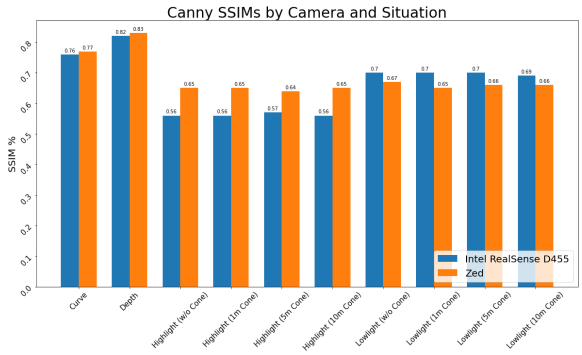
RESULTS

To get a comprehensive performance evaluation of the cameras across all filters, the team summed the SSIMs of every filter. Observing the graph below, it is clear to see that the ZED camera outperforms the Intel RealSense D455 with all differences greater than .17. Across the depth and all low light situations, the difference between the summed SSIMs are small. However, the SSIM differences between the cameras tend to grow in every iteration of the high light situations favoring the ZED camera.



(Figure 1: Sums of SSIM by Camera and Situation)

The canny filter consistently returns high SSIM values. As such, the team investigated the difference in performance regarding this filter. For the most part, it seems that the trend from the comprehensive evaluation echoes into the Canny evaluation. Again, the highlight situations show more of a difference favoring the ZED camera, however the values do not increase over each depth iteration of the evaluation. The differences between the Depth and Curve are marginal with a difference of 0.01. Interestingly, the Intel RealSense D455 is outperforming the ZED camera in low light situations with an average difference of .0375. While this average is small, it is interesting to see the Intel RealSense perform better than the ZED.



(Figure 2: Canny Filter SSIM by Camera and Situation)

DISCUSSION AND CONCLUSION

In figure 1, the ZED camera is the clear winner. With consistently high SSIM values beating the Intel RealSense D455 in every situation, the team would easily recommend the ZED camera. However, in any AV application, a single image will not be filtered five times. Only one filter will be chosen as the edge detection algorithm.

However, just for the evaluation purpose between the two cameras, summing up the SSIMs makes sense as an intuitive metric, and it proves to be telling a message.

The differences between the SSIMs of the two cameras in high light settings are notably large. From the other teams' feedback, the Intel RealSense D455 is known to have issues in light sensitivity problems. While configuration of the Intel RealSense camera for sensing light is still on research, as both of the cameras are used under default settings in the experiment, they are being contested fairly. It is summarizable that the ZED camera performs better than the Intel RealSense D455 camera in high lighting conditions. This conclusion puts the ZED on top because racing on the Thunderhill will likely happen during daytime.

That being said, the team found that for any outdoor AV image processing application, the ZED camera proves to be more robust. Additionally, the ZED camera was in fact able to outperform the Intel RealSense D455 at a lower resolution with a much higher frame rate according to the SSIMs result. As mentioned in the introduction, the ZED is running 100 FPS at VGA 376p, while the Intel RealSense D455 is running 60 FPS at 480p. Still, the ZED outperforms the Intel RealSense by employing a much higher FPS, and as well as resolution. The team has conducted experiments by recording videos to prove this.

It is mentionable that the ZED SDK offers some more features that the Intel RealSense SDK does. While they both have multiple sensors equipped inside their bodies besides just the cameras, like IMUs, ZED offers some insightful features. As they are both able to generate depth heat maps, and calculate distances between themselves and the pixels in the image, ZED offers a function of generating 3D Point Cloud maps. When accessing an environment, and learning about the actual construction of the environment is needed, a 3D Point Cloud map reconstructs a space map of the environment with pixels in 3D, viewable by the ZED SDK. This can be done by browsing around with the ZED stereo camera, which is supposedly capable of perceiving depth. Also, the ZED offers an in-depth guidance and official suitable environment in implementing YOLO (You Only Look Once), a neural network object recognition system. While object

recognition is not the most crucial aspect for running on Thunderhill, this is insightful in further autonomous vehicle research and development.

It is surprising that the ZED camera is outperforming the Intel RealSense D455 camera in so many aspects, based on the team's analysis. Considering that the ZED is actually more expensive than the Intel RealSense, this would make sense. For the purpose of selecting a camera for the collaborative vehicle, the team recommended that the ZED be integrated. Fortunately, the team was supplied with both cameras and cost was not an issue. In outside mission critical applications where safety is at utmost importance, the team still recommends the ZED camera. The easy-to-use SDK and high frame rate make the ZED camera more attractive than the Intel RealSense D455. However, for outside hobbyist applications, where another team may need to purchase the camera, the Intel RealSense D455 may be a more appealing product. The differences in the canny situation shown in Figure 2 are marginal, and the small 10% difference may not be worth the additional $100 USD to purchase the ZED.

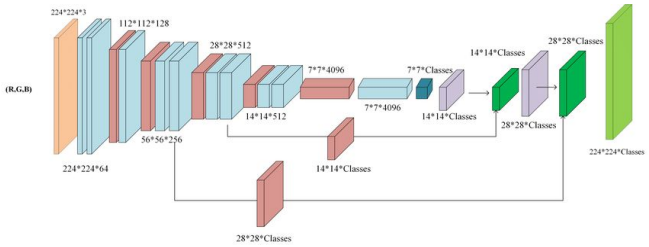
LANE SEGMENTATION

One of the key functions needed for an autonomous vehicle is the ability to detect the road and see where it can and cannot maneuver. Our autonomous vehicle will process camera input images live time to detect the lanes in each image and then using the processed information, determine what the next maneuver will be. To achieve such a task, a pixel wise classification of every pixel in each image must be made. To do so the team will use deep learning. There are two major categories of segmentation; semantic segmentation and instance segmentation. In semantic segmentation, objects in the same category are classified with the same values whereas in instance segmentation each unique instance of an object is classified with a different value. Semantic segmentation is generally easier to achieve a higher accuracy and for the team's purposes, segmentation of only the roads is necessary, so the team will implement semantic segmentation.

MODEL

A common structure used in semantic segmentation is the Fully Convolutional Network. The general idea of a FCN is to extract the features of an image using a

convolutional neural network and then upsample via deconvolution then add the deconvolved features and CNN features. The idea is that the deconvolved features will have the features of an image while fusing the CNN features will preserve the location features.

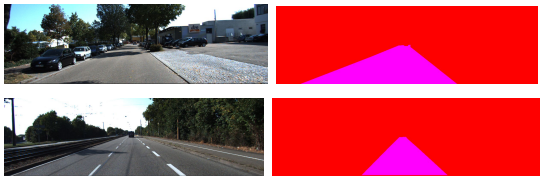


(FCN Architecture [2])

A Fully Convolutional Network performs well on semantic segmentation and is relatively fast. However, when the team implemented the model and ran it in the Jetson single board computer, the team were only able to get results at a speed of only five frames per second. The team chose to implement the U-Net model instead, for its faster speed and relatively high accuracy. Some key differences of U-Net and FCN are that FCN only upsamples once while the U-Net has multiple upsampling layers and U-Net uses skip connections and concatenates the upsampled features and the CNN features instead of adding them together.

DATASET/TRAINING

To train a neural network to be able to segment roads, a good amount of data is needed. However, collecting large amounts of the team's segmented data for roads can be time consuming. The team decided to utilize publicly available datasets to train the model and further tune the model with the team's dataset. The KTTI-Road dataset contains 289 training images and 290 test images. Each training image is an image of a road and contains a label masking the road.



[KITTI dataset]

[U-Net predictions on test set (blue: road, green: non-road)]

After training the model on the KITTI road dataset, the team further trained the model

on Thunderhill Raceway track dataset that the team manually annotated. Due to COVID-19 travel restrictions, the team was not able to go to the track in person so the team extracted images from a dash camera video of a race in the Thunderhill Raceway Track.

[Sample results of model prediction on Thunderhill track]

CONCLUSION

When trained on the KITTI dataset, the model performed well on the KITTI test set. The train set and test set images were different images, but still in similar environments. For higher performance, the team trained the model further with the dataset extracted from the dashcam video. After training, the model is able to segment lanes well even in the dash camera videos. However, one thing to note here is that the dash camera video was used as the training set so the model may be overfitting to it (although it seems to do a good job generalizing when checking the results on the KITTI dataset). The model will have to be tested out on the real track, but because of pandemic travel restrictions, the team could not test the model in a real track. Further improvements will be made once the model is tested on a real track.

REFERENCES

[1] “Allen Berg Racing Schools - Thunderhill.mov.” Edited by Allen Berg, *YouTube*, 16 May 2011, www.youtube.com/watch?v=jLmtuveVzrs.

[2] Piramanayagam, Sankaranarayanan & Saber, Eli & Schwartzkopf, Wade & Koehler, Frederick. (2018). Supervised Classification of Multisensor Remotely Sensed Images Using a Deep Learning Framework. Remote Sensing. 10. 1429. 10.3390/rs10091429.