

# ENPM 673 - FINAL PROJECT

## TURTLEBOT CHALLENGE

### GROUP 14 MEMBERS:

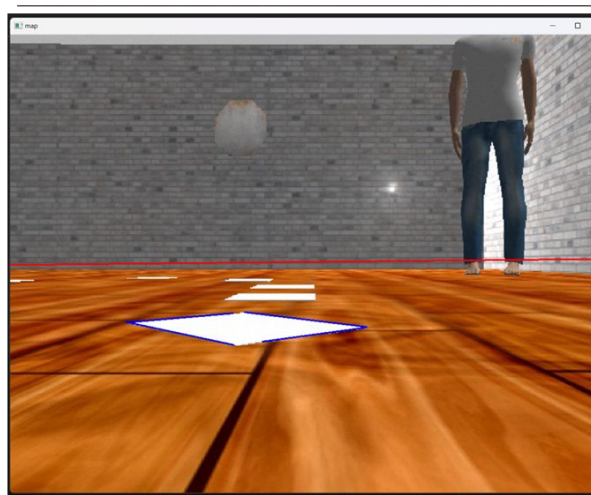
Anbarasan Kandasamy (120270697)

Hariharasudan Muralidaran (120172656)

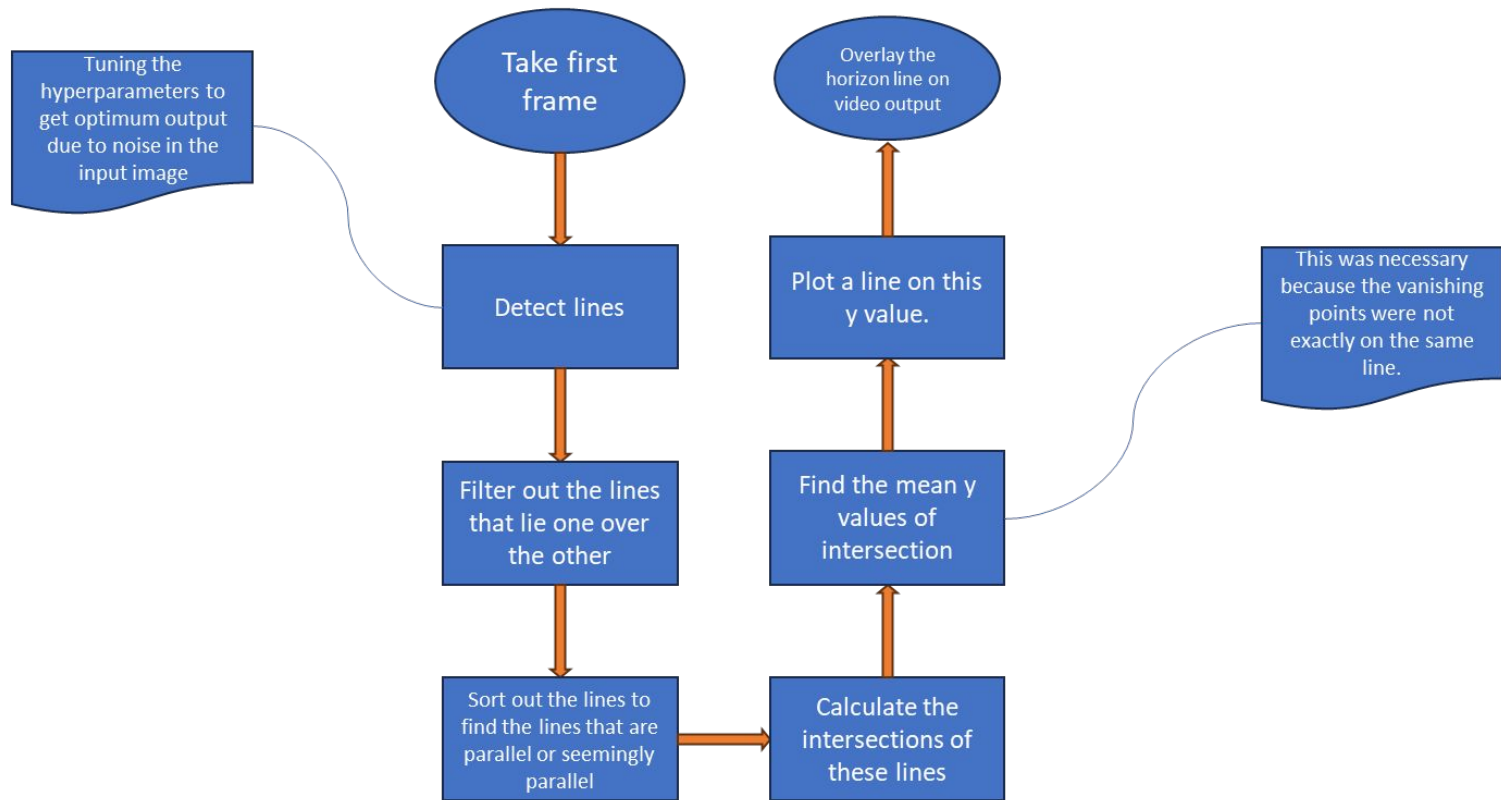
Manoj Kumar Selvaraj (120511257)

Swaraj Mundruppady Rao (120127007)

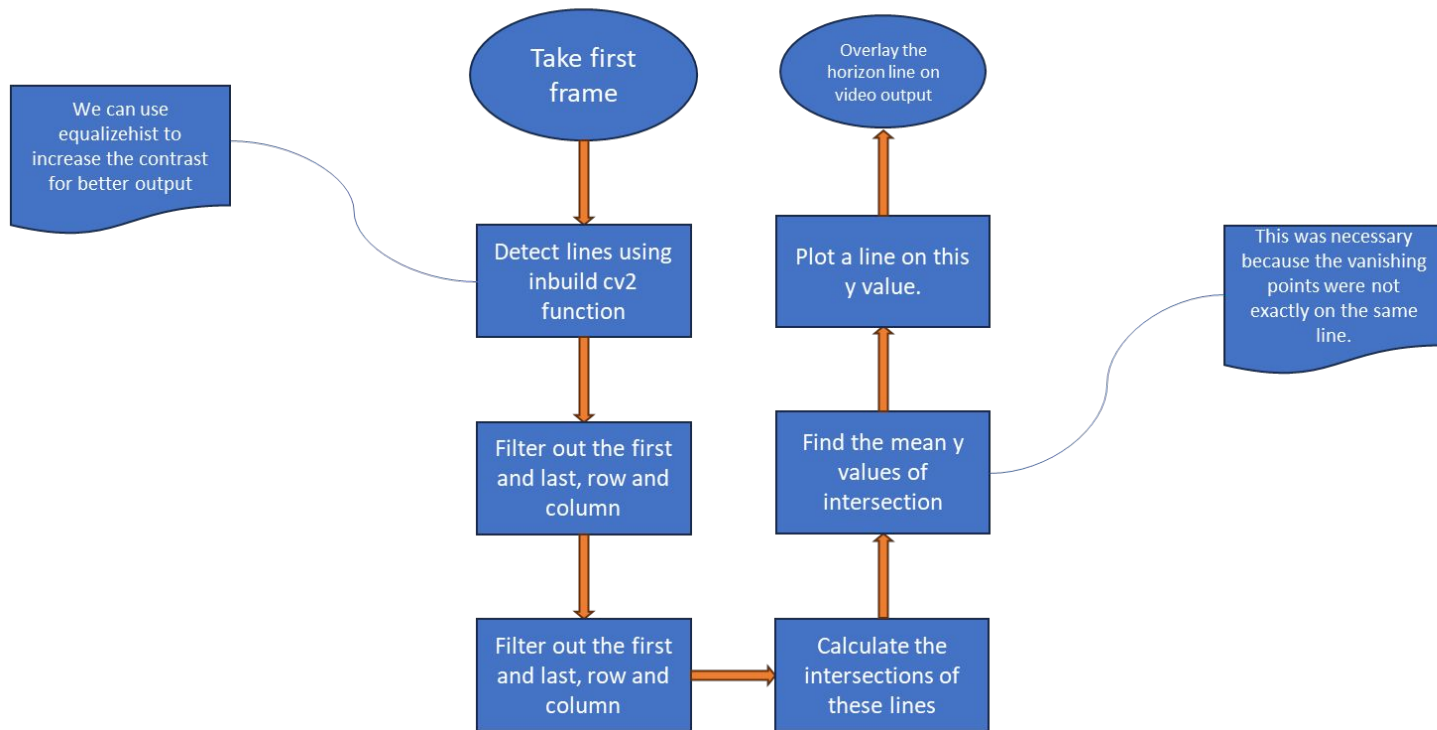
# HORIZONTAL LINE DETECTION

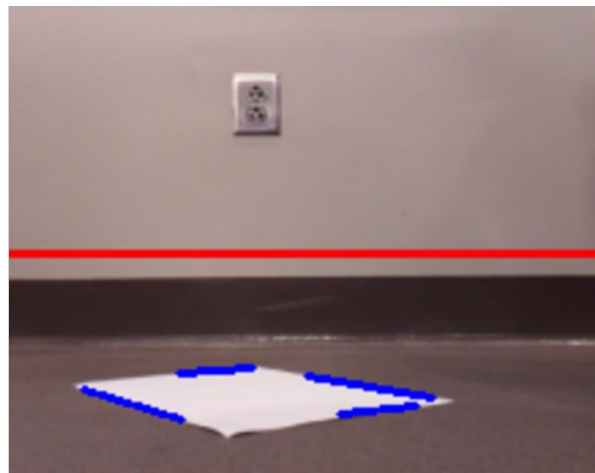
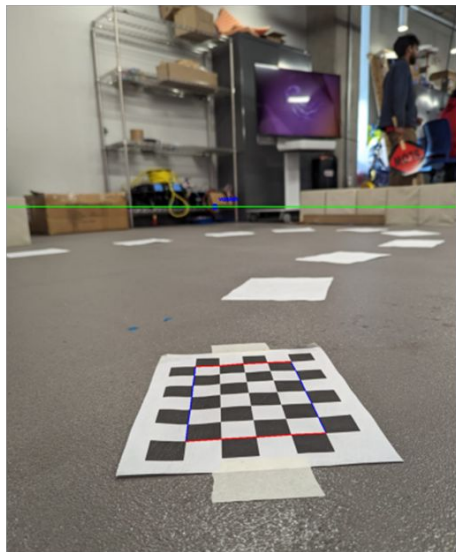


# Horizon line detection using paper

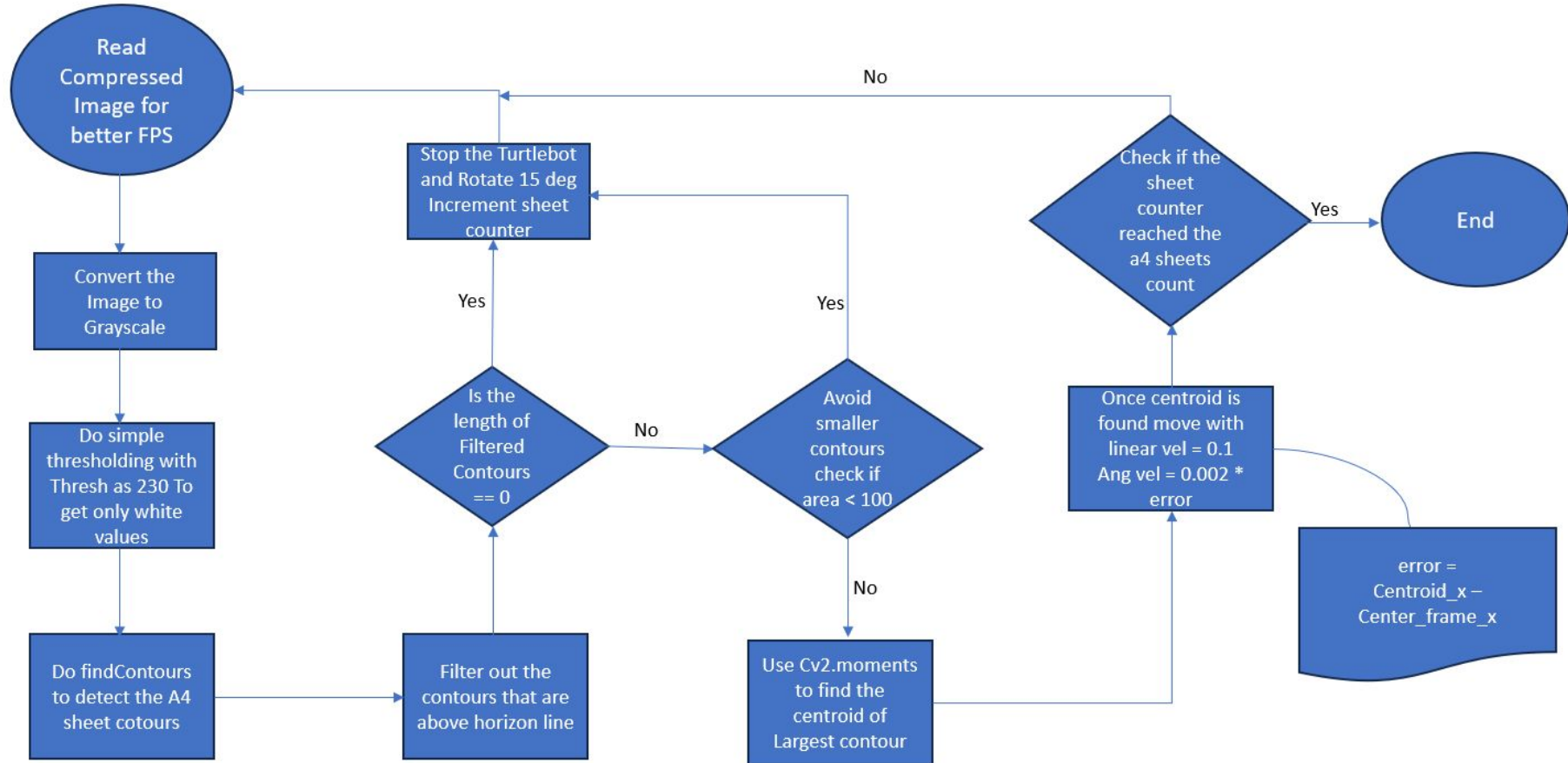


# Horizon line detection using chessboard pattern



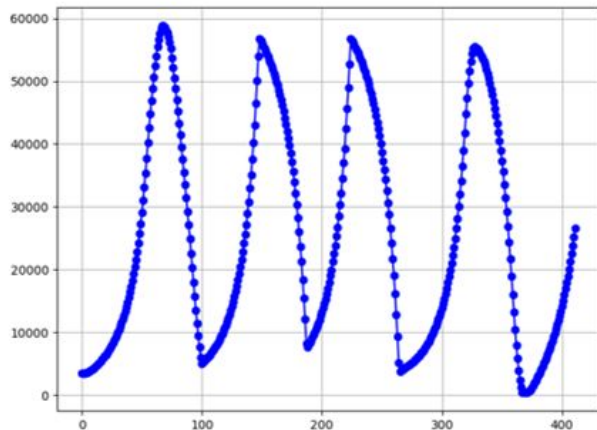


# Centroid Follower:

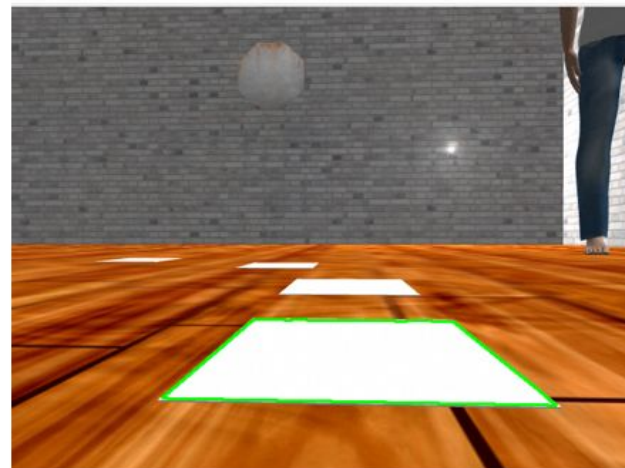


# Challenges:

1. Smaller contours made the Turtlebot follow unwanted centroid points.
  - Resolved this using a check to filter out contours based on area and horizon line.
2. Initially, area-based data were used to detect the final A4 sheet for the robot to stop, but inconsistent sampling led to unreliable results.
  - Instead, we implemented a counter system that increments when A4 sheets are not detected. This made the robot stop at the final A4 sheet.



# Result:





# STOP SIGN DETECTION





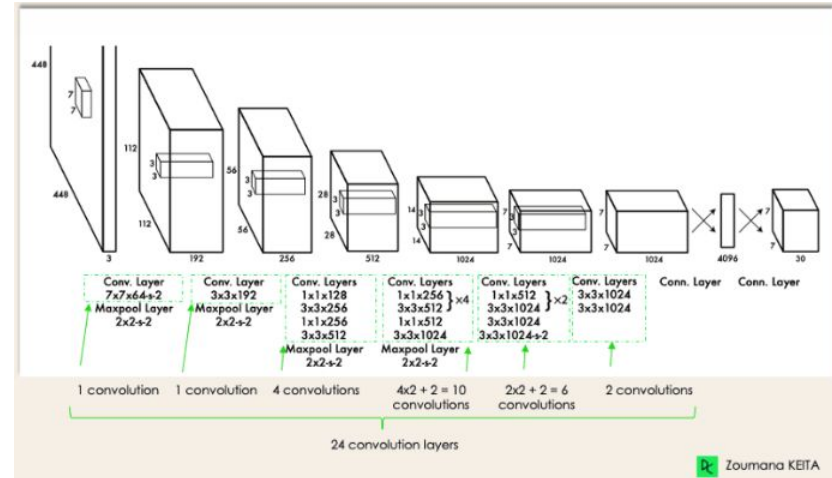
# YOLO(You only Look Once)

## A bit about Yolo:


- Uses Deep CNN network to detect objects
- Divides image into a grid and each grid cell predicts certain number of bounding boxes and confidence scores for those boxes. Also predicts class probability for each box

## Why YOLO was used ?

- Speed (45 frame/sec)
- Good Accuracy (Has fewer background errors)
- Easier to train
- Good Generalization



```
swaraj@swaraj: ~/ENPM673_turtlebot_perception_challenge
```



A screenshot from a 3D environment. On the left is a dark brick wall. In the center, a small green light source is visible on the ground. To the right, a grey grid floor extends into the distance. A blue vertical line is visible on the left side of the image.

I

1

1111

49

1

1

1

1

1

1

1

1

1

1

1



1

1

1



1

Reset Time

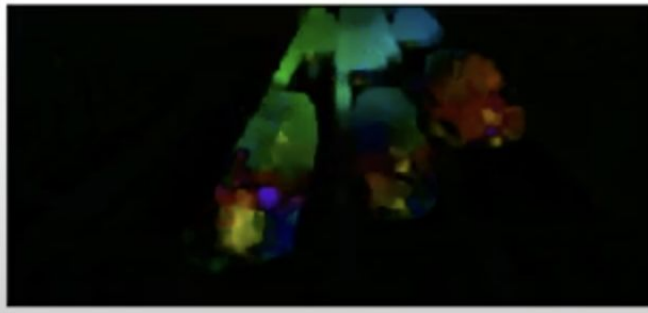
# Problems Faced

- Using Feature matching : Slow and was not always giving the best results
- Haar Cascade :
  - Not Always accurate
  - Performance drop in complex variable scenarios
  - Better suited for low resolution images rather than video processes
  - Struggles with variations in object scale, pose and orientation

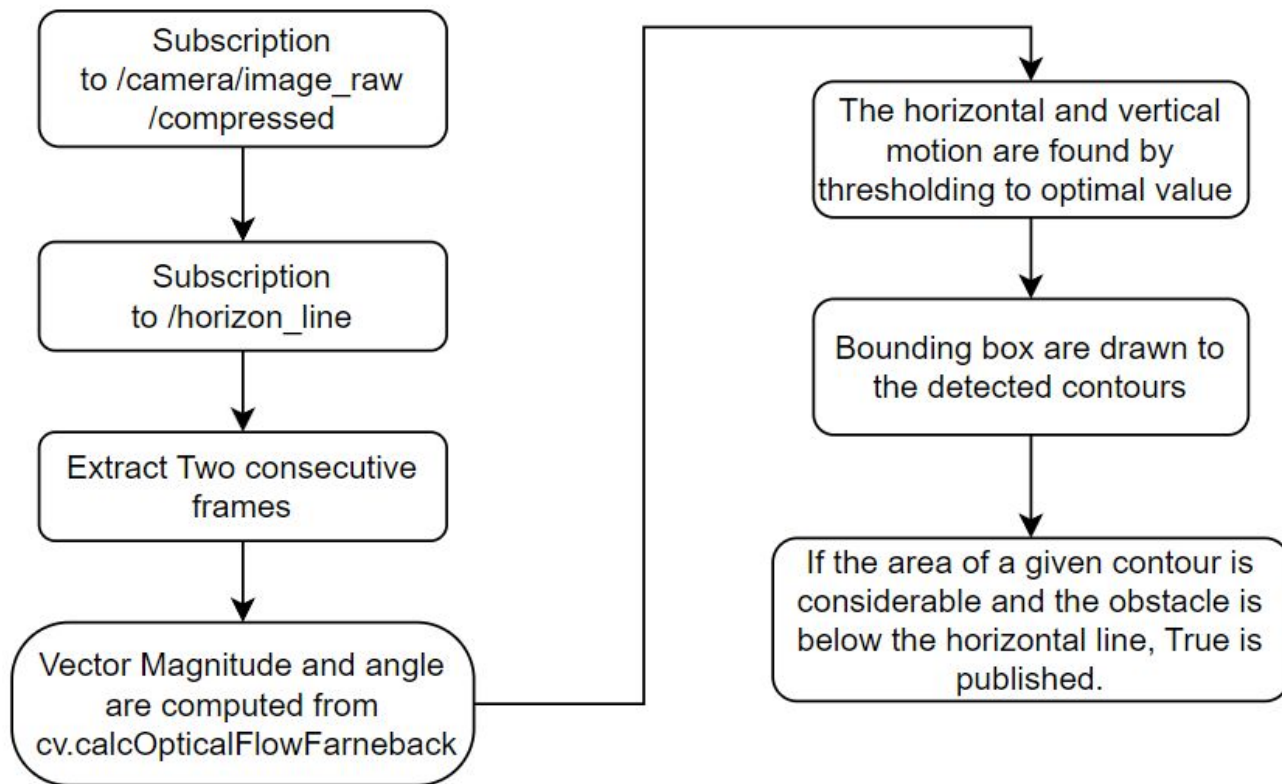
# DYNAMIC OBSTACLE DETECTION



optical flow



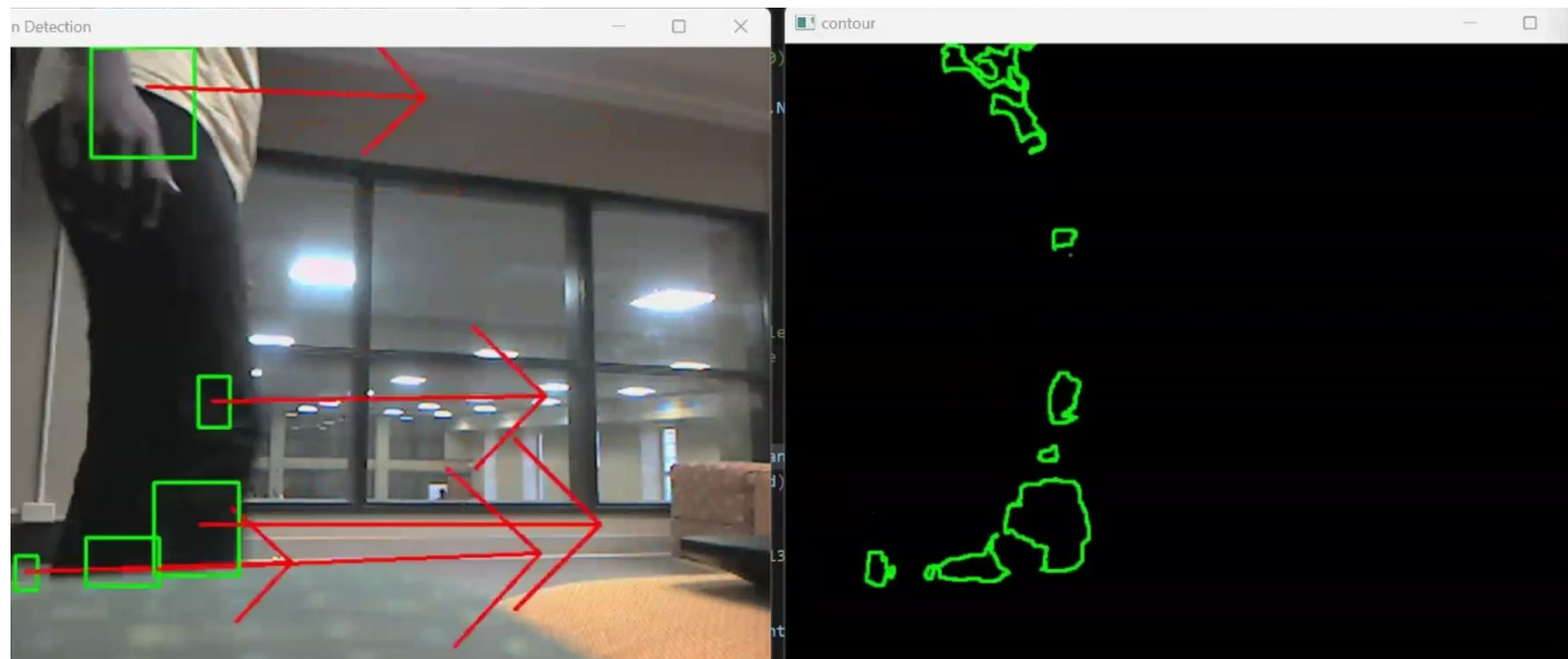
# PIPELINE



# Problems Faced

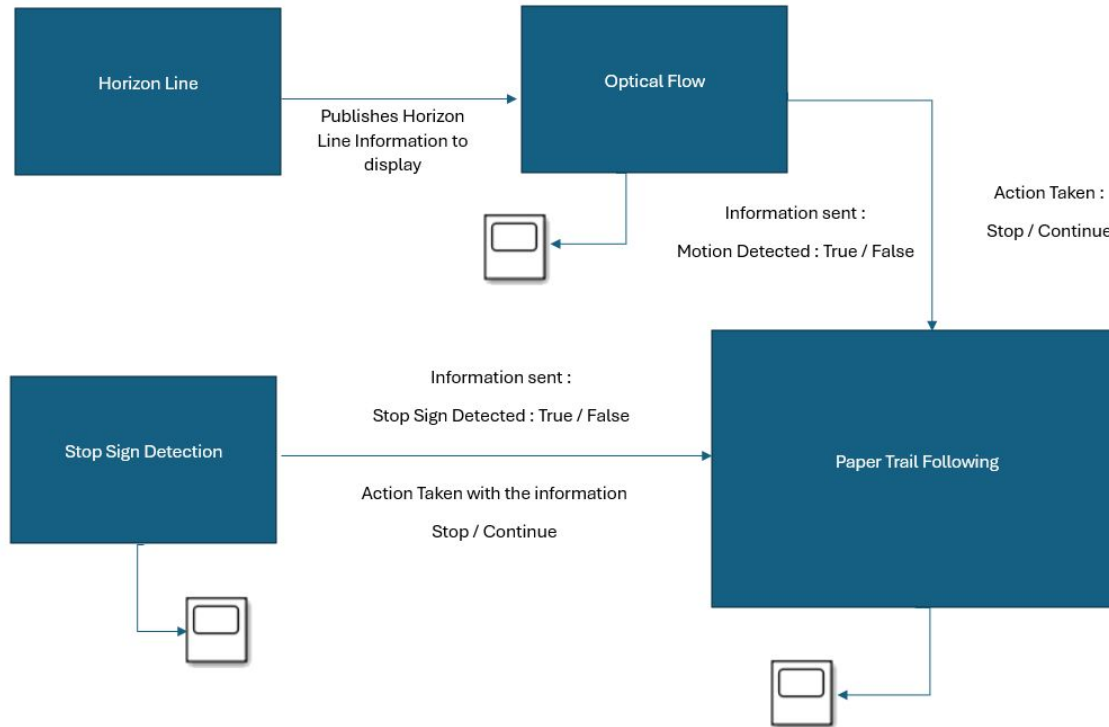
- Filtering the flow vectors that are caused due to the motion of the turtlebot itself.
- Tuning the threshold for real time application

# Demonstration





# ROS2 (Humble) FRAMEWORK





THANK YOU  
Q&A ?