A.D.A.S. LANE DETECTION USING CLASSICAL COMPUTER VISION

EECE-5554: ROBOTIC SENSING AND NAVIGATION GROUP 6

PRESENTED BY:-

ABHINAV GUPTA

KAUSHIK VENKATESH KRISHNAMURTHY

SARTHAK GUPTA

SENTHIL MANICKAVASAGAM

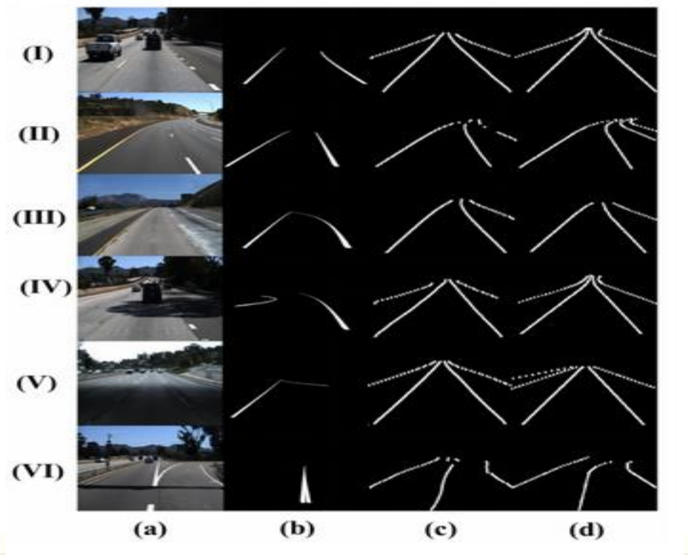
PROJECT OBJECTIVES

- Detect ego and adjacent lane markers using Computer Vision
- Be able to track lane lines when lane lines are dashed/occluded
- Warn user when lanes are not visible
- Provide warning during unintended lane change with help of U/I
- Indicate occurrence and direction of intended lane change with the help of U/I
- Handle Lane merge and divergence cases

A SURVEY ON LANE DETECTION

Traditional Computer Vision based methods

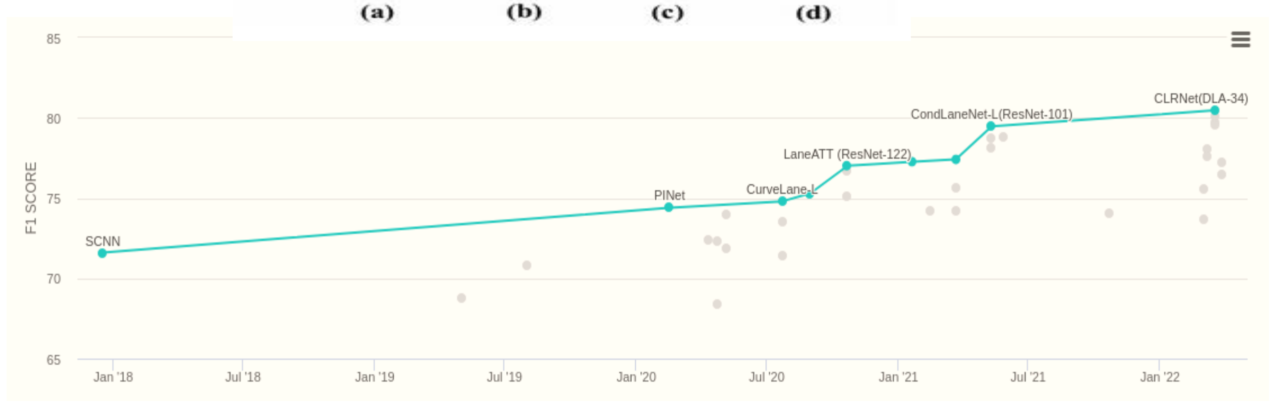
- Gradient Based edge detection methods:
  - Canny Edge detection
  - Hough Transforms
  - Sobel Filtering
- ED lines detection
- Maximally Stable Extremal Regions



A- RGB frame
B-CV methods
C- SCNN
D-Ground Truth

Deep Learning Methods

- Spatial Convolutional Neural Networks
- Cascading Neural Nets
- Vanishing Point detectors



Why are Neural Nets the state of the art?

Because they handle extreme conditions much better compared to regular Computer Vision Methods. They also handle reflections, occlusions and lane tracking.

[1] https://www.researchgate.net/profile/Xuechen-Zhang-4/publication/350714815_Deep_Learning_in_Lane_Marking_Detection_A_Survey/links/6097bfad92851c490fc7d7ce/Deep-Learning-in-Lane-Marking-Detection-A-Survey.pdf?origin=publication_detail

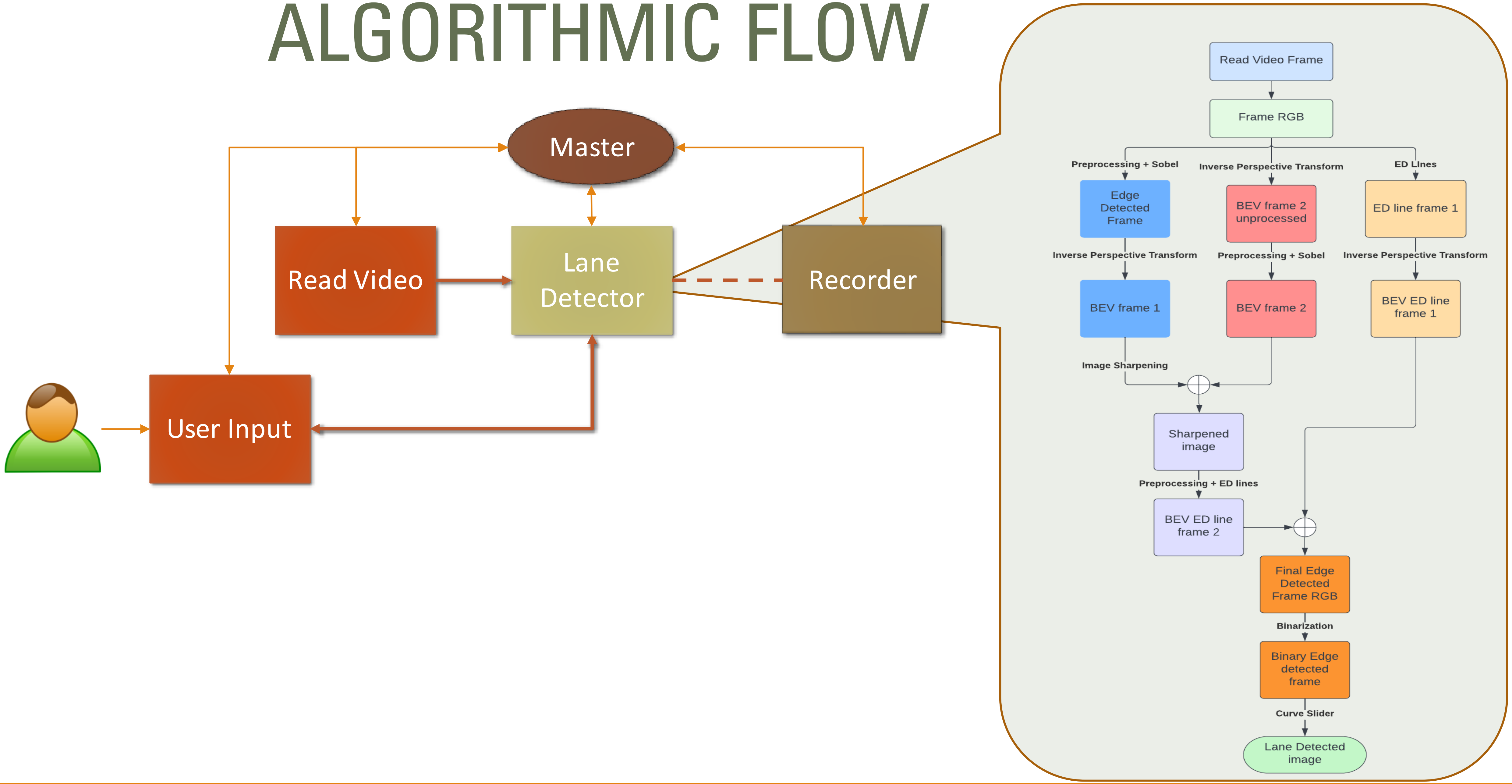
POPULARIZED METHODOLOGY

Widely used methods include the following base processes [1]

- Obtaining an image frame from a mounted camera
- Color space Conversion, thresholding and Image smoothing
- Vanishing point detection or hand calibration of sky view mapping points
- Detect lane markers with edge detection and line drawing
- Curve fitting in sky view
- Remap to original view

Ref:
1-https://arxiv.org/abs/2104.04755

ALGORITHMIC FLOW

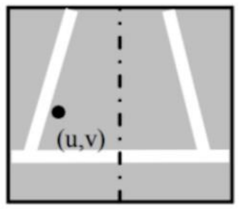


PREPROCESSING STAGE

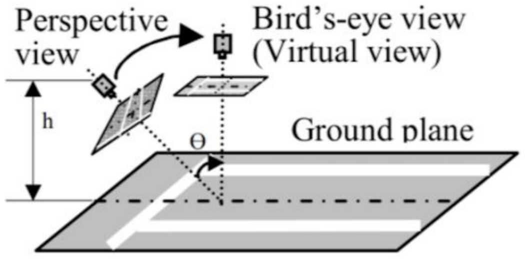
BEV PERSPECTIVE TRANSFORM

- Four region of interest points are identified on ego lane.
- Transformation to parallel line points in BEV.
- The Zone of Interest (ZoI) has 4 different points which cover the drivable region of the road.
- BEV is applied on the frame and further computation is done only on the transformed ZoI.

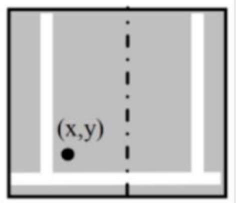
Ref:https://developer.ridgerun.com/wiki/index.php?title=Birds_Eye_View/Introduction/Research



(a) Perspective view image



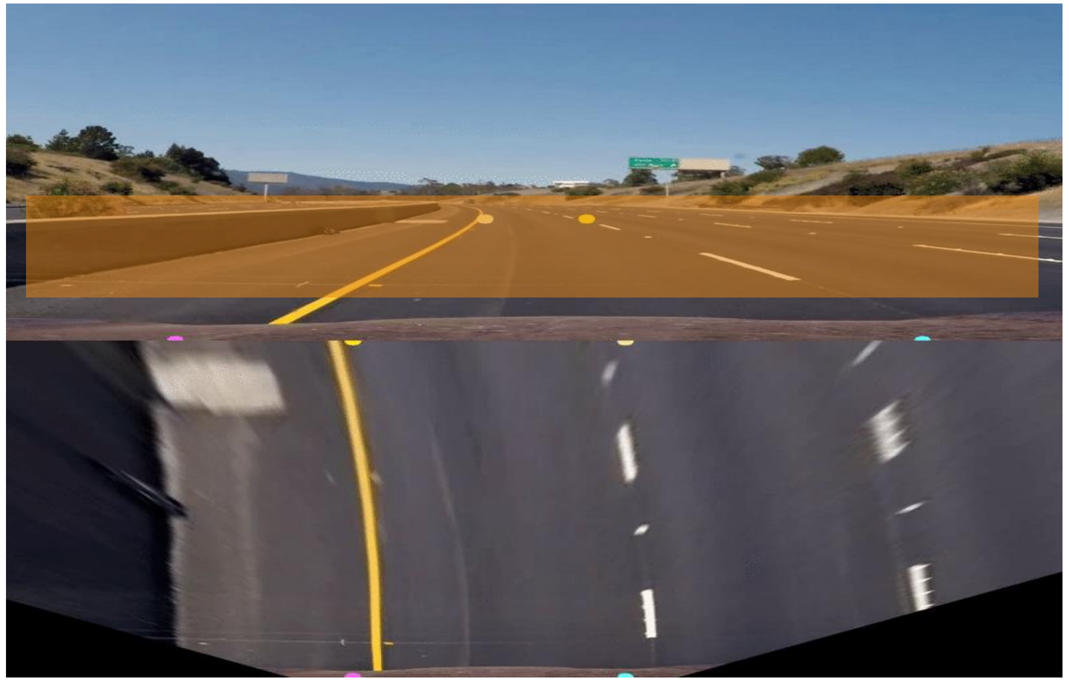
(b Perspective transformation

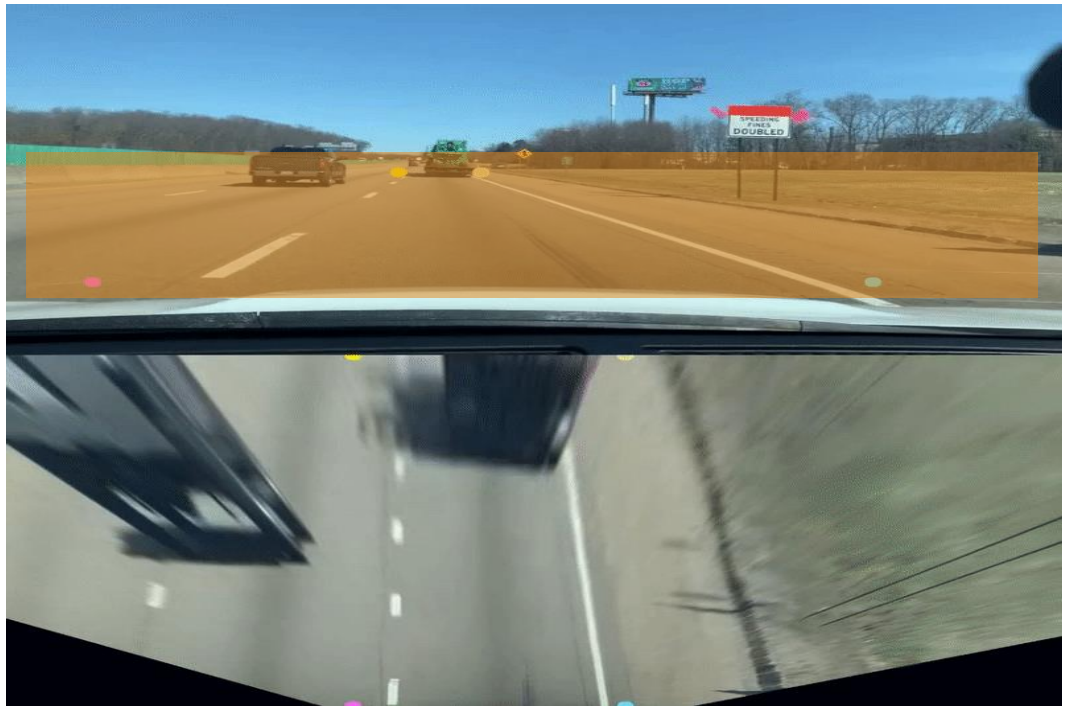


(c)Bird's-eye view image

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

$$x = \frac{x'}{w'} \quad and \quad y = \frac{y'}{w'}.$$



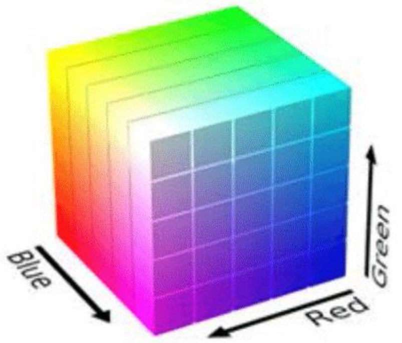


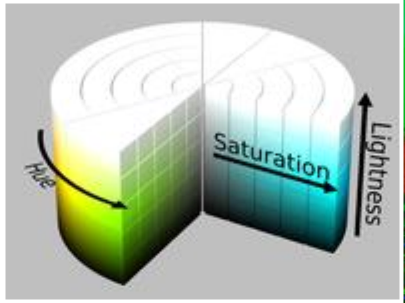
PREPROCESSING STAGE

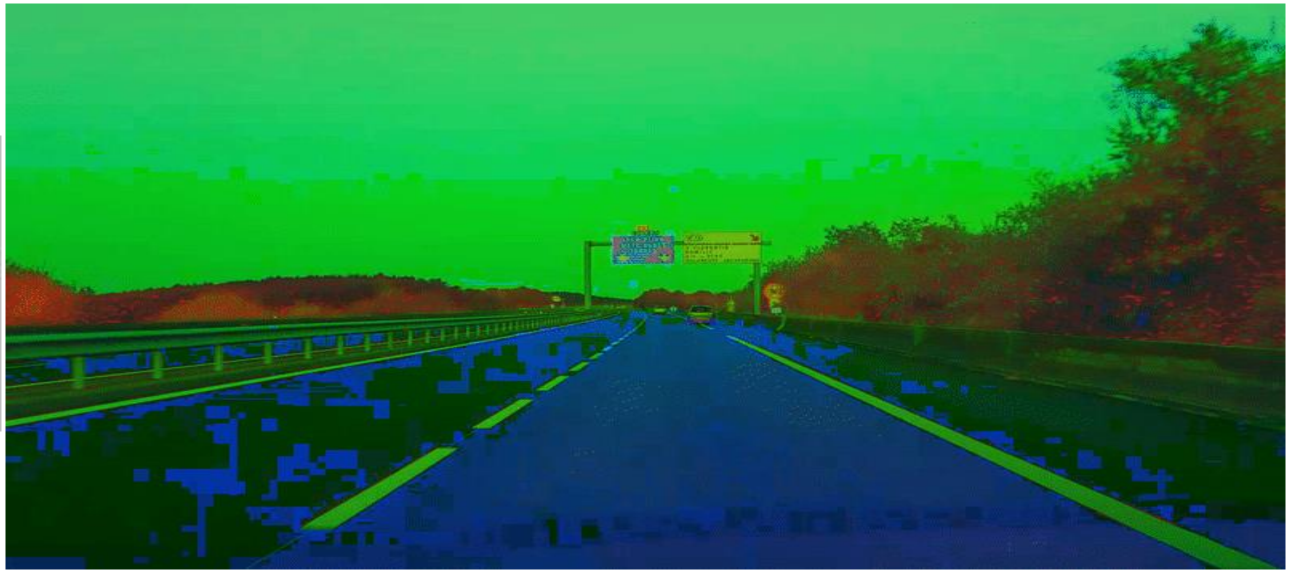
COLORSPACE REMAPPING

- RGB frame is transformed to HSL.
- HSL : Hue, Saturation and Lightness.
- We perform preprocessing only on H and S frames as they are invariant to light conditions.
- Preprocessing includes exposure balance, highlight balance and shadow removal to remove unwanted artifacts which might add false edges.

Ref:https://developer.ridgerun.com/wiki/index.php?title=Birds_Eye_View/Introduction/Research







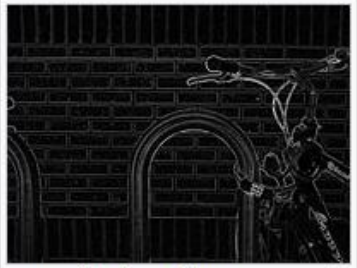
EDGE DETECTION

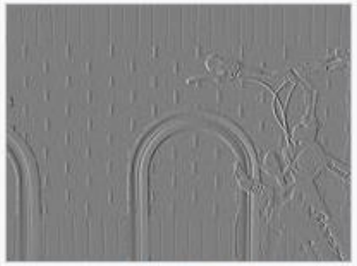
SOBEL FILTERING

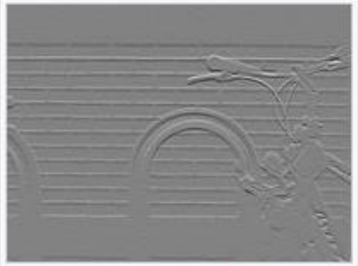
- Sobel operator is used mainly for edge detection in computer vision applications.
- The operator uses a kernel that convolves over an entire image and calculates the change in gradient in the x and y directions.
- These resultant of these two images can be taken and the all the edges can be detected along with direction of the gradient change.

Ref: https://en.wikipedia.org/wiki/Sobel_operator

Grayscale test image of brick wall and bike rack


Normalized gradient magnitude from Sobel–Feldman operator


Normalized *x*-gradient from Sobel–Feldman operator


Normalized *y*-gradient from Sobel–Feldman operator

X – Direction Kernel

-1	0	1
-2	0	2
-1	0	1

Y – Direction Kernel

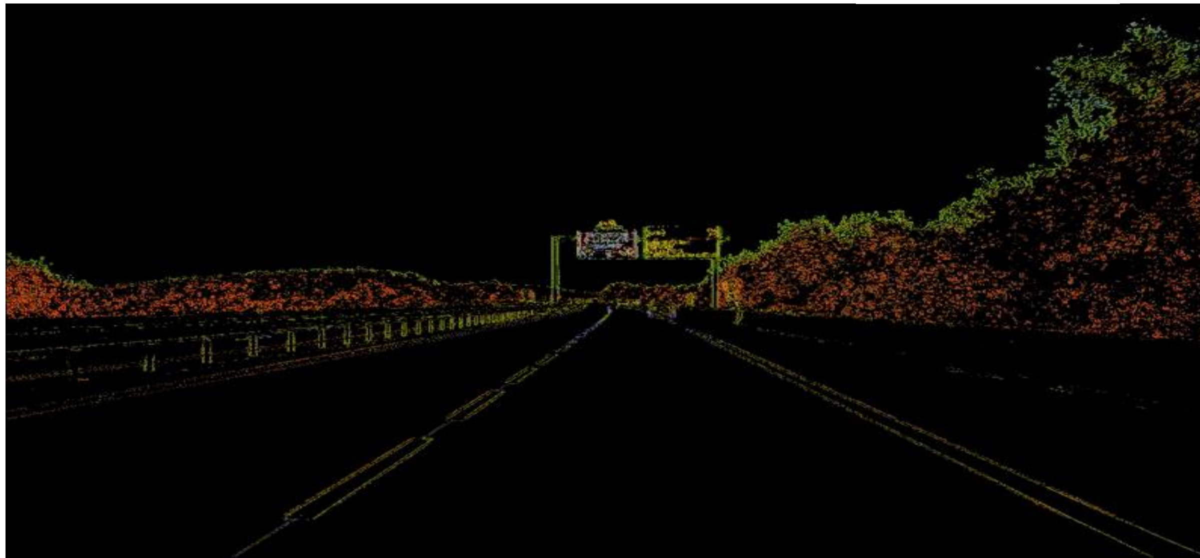
-1	-2	-1
0	0	0
1	2	1

$$\mathbf{G} = \sqrt{\mathbf{G}_x{}^2 + \mathbf{G}_y{}^2}$$

Magnitude

$$\mathbf{\Theta} = \operatorname{atan}\left(\frac{\mathbf{G}_y}{\mathbf{G}_x}\right)$$

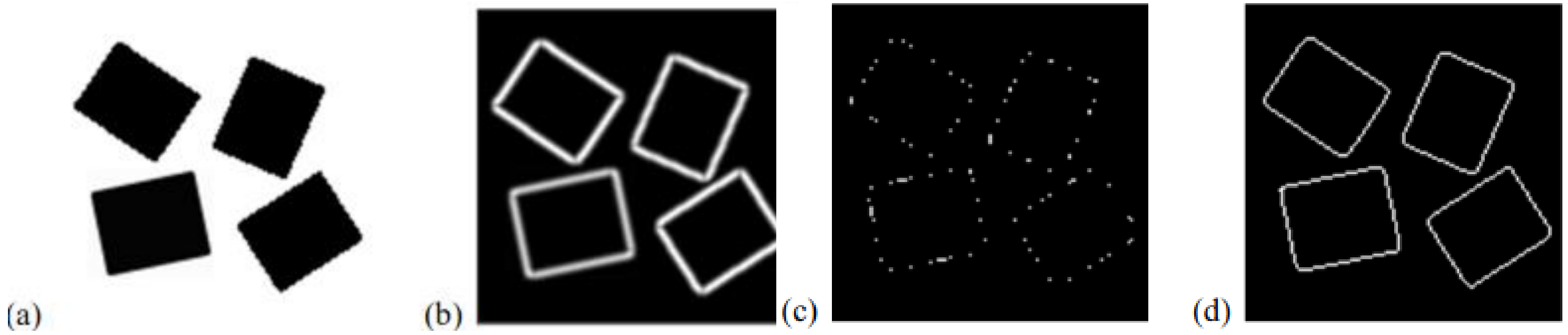
Direction

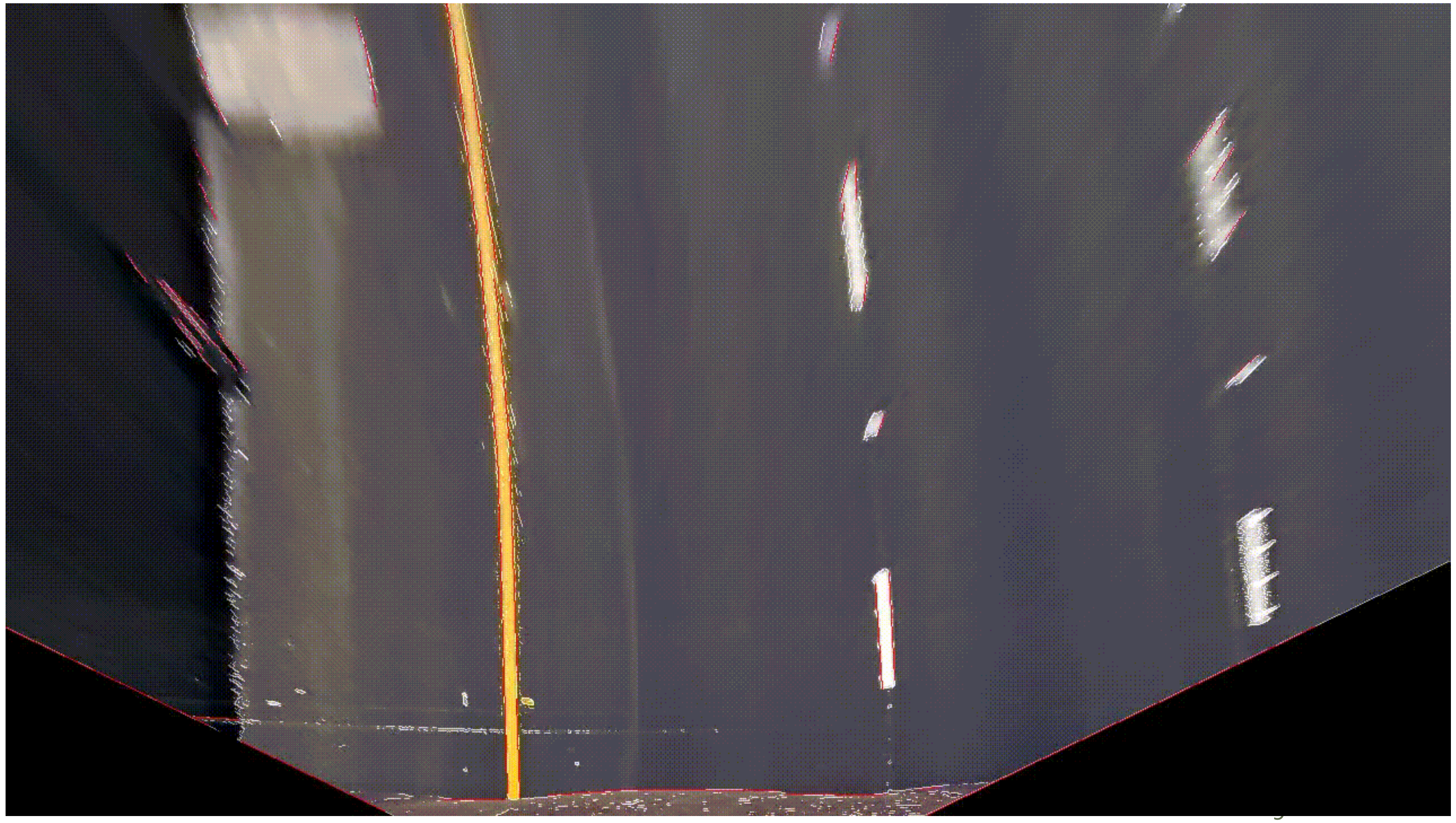


LINE DETECTION:

THE ED LINE ALGORITHM

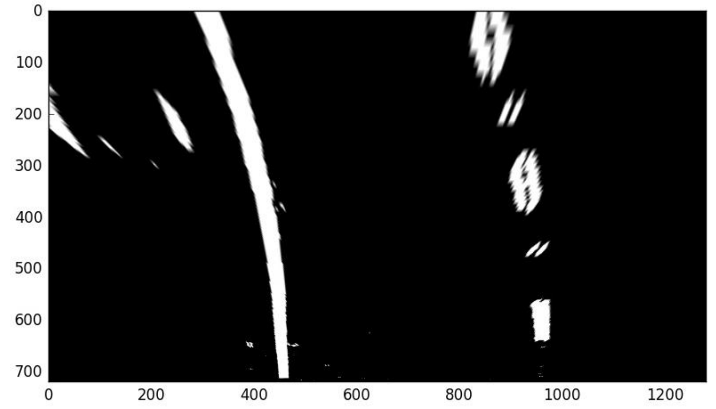
- A very fast algorithm to detect edges in images.
- Calculates anchor points in an input image and fits a straight line through these anchor points using minimum Least Square error.
- The anchor points are defined at peaks of the gradient map which are derived from an edge detected mask which can be calculated using any standard edge detection algorithm like Sobel, Prewitt filtering etc.
- Detected lines are thresholded on angle argument to filter horizontal edges

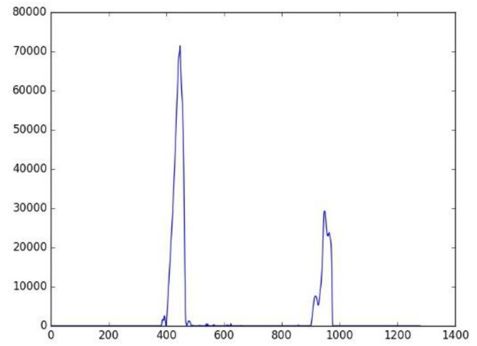


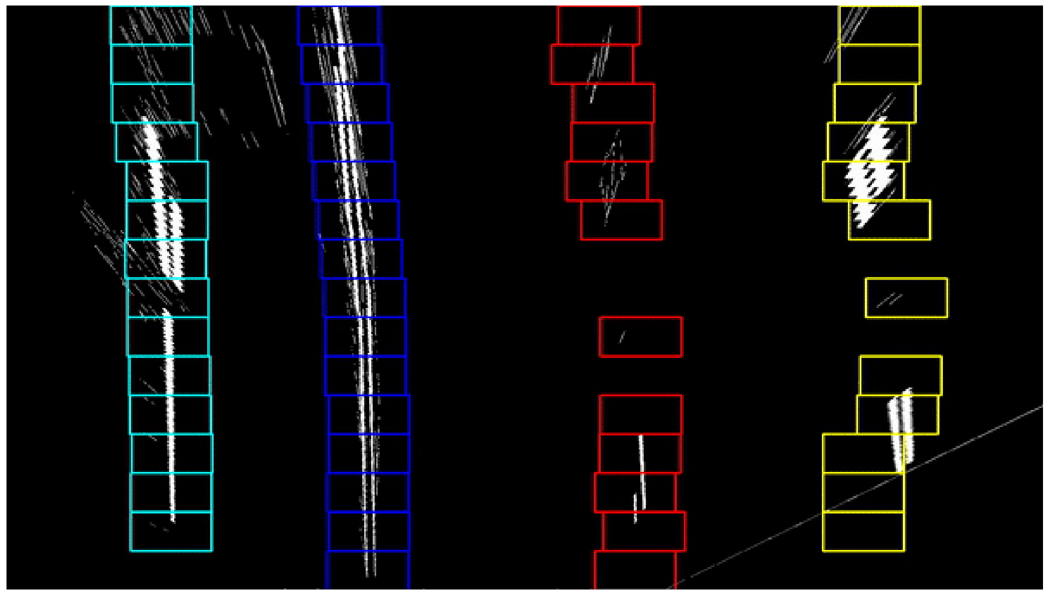


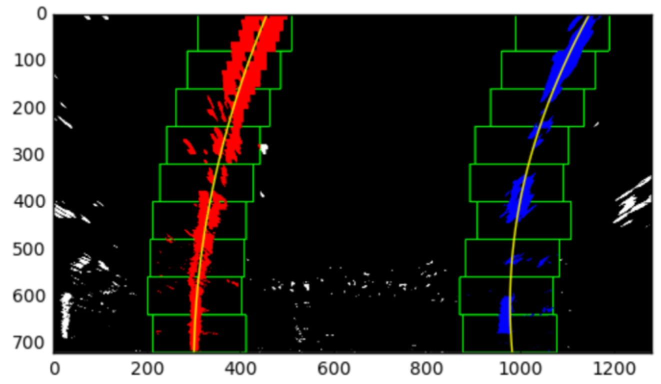
SLIDING WINDOW BASED CURVE FITTING & LANE DETECTION

- The histogram of the detected edge map is created.
- The algorithm then iterates over different column regions in descending order of histogram values.
- A rectangular window traverses in bottom-up direction of the image in these column regions. If there is a minimum no. of pixels inside the window, then it is considered part of the lane.
- For n lanes, n blocks of sliding windows are chosen. Then pixels are extrapolated within these windows. Finally, a 3rd degree polynomial is fit on these pixels to get the curve.









CODEBASE MODIFICATION SUMMARY

Primary:

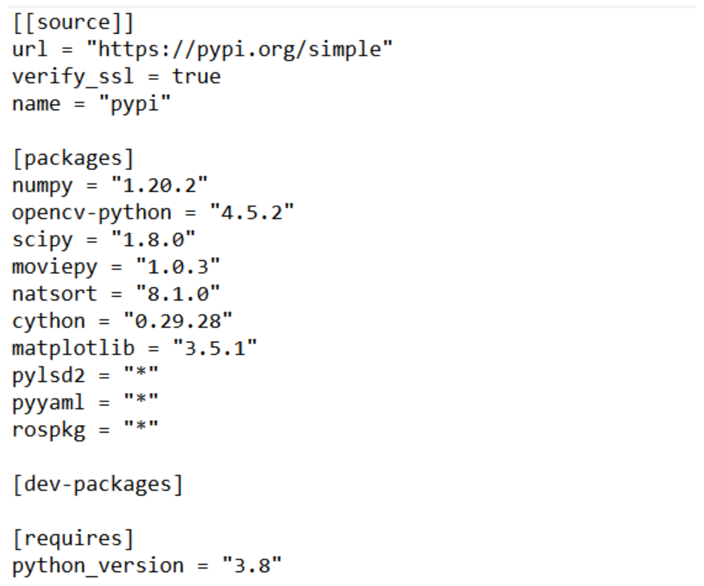
OpenCV2

Utilized Python wrapper over edlib.so files (pylsd2)

Utilized Udacity lane detection codebase from Nguyen

Modifications

- Additional dependencies encapsulated in a pipenv
- Restructuring for an OOP based approach and reading live video feeds
- Preprocessing stage modified to incorporate fused views
- Backend changes for
  - Adding CF
    - A complimentary filter is applied on coefficients of previous frame and current frame for each lane. The confidence value of current frame is the alpha weighting parameter between the two curves.
    - CF = $\alpha C_1 + (1-\alpha)C_2$
  - Handle lane merging divergence
  - Handle lane change
  - ROS support and UI decluttering



```
[[source]]
url = "https://pypi.org/simple"
verify_ssl = true
name = "pypi"

[packages]
numpy = "1.20.2"
opencv-python = "4.5.2"
scipy = "1.8.0"
moviepy = "1.0.3"
natsort = "8.1.0"
cython = "0.29.28"
matplotlib = "3.5.1"
pylsd2 = "*"
pyyaml = "*"
rospkg = "*"

[dev-packages]

[requires]
python_version = "3.8"
```

TESTING SCENARIOS

3 Primary Quests

Multilane highway with marked lanes and no occlusion

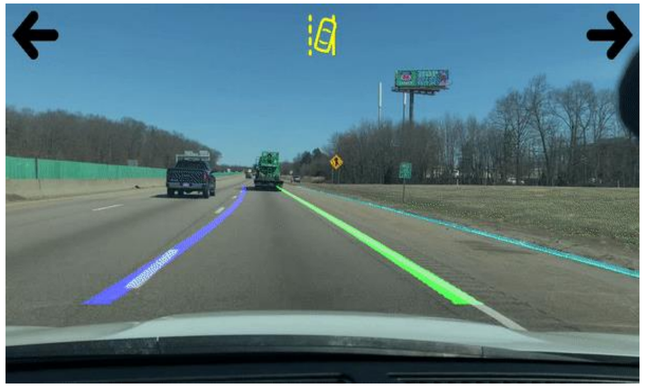
Multilane highway with marked lanes and vehicles in close proximity

Night scenes with poor visibility of lane markings, locally bright or dark spots and mounting vibration

3 Side Quests

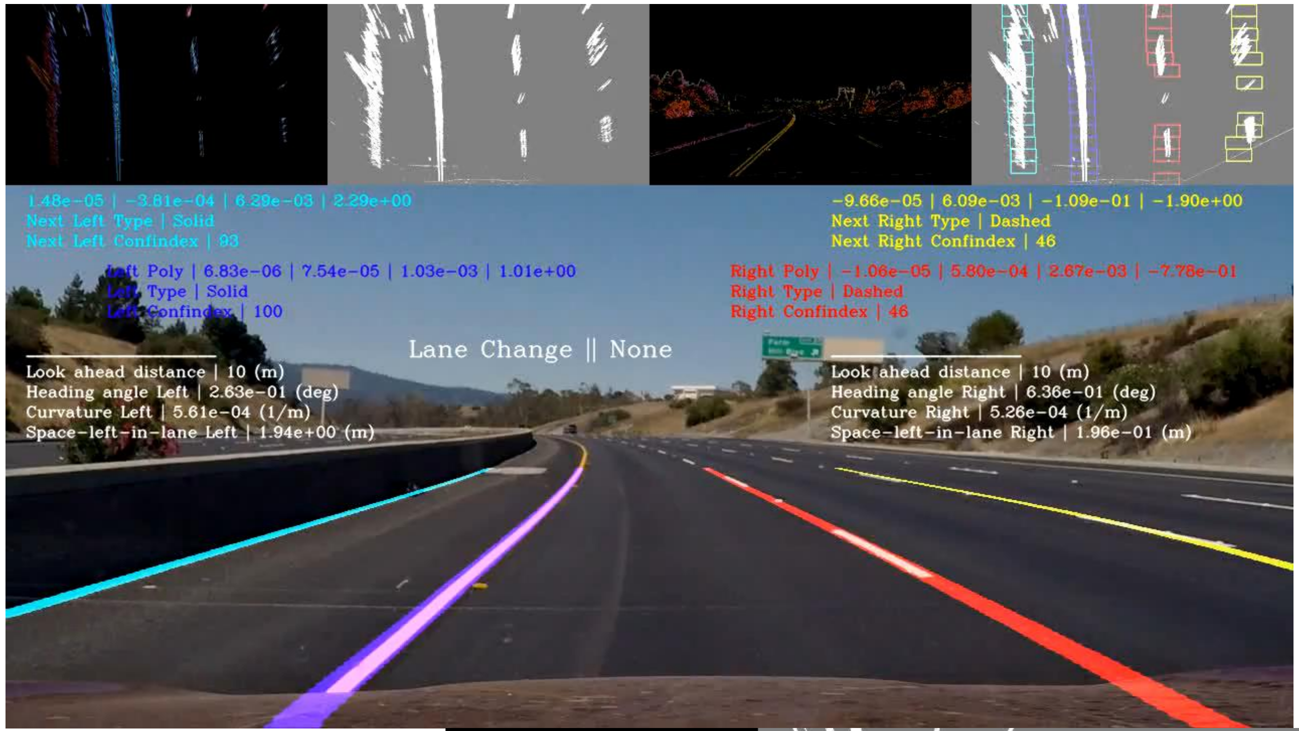
Lane Change Detection

Lane Merge/Divergence Detection


Handling manual turn inputs

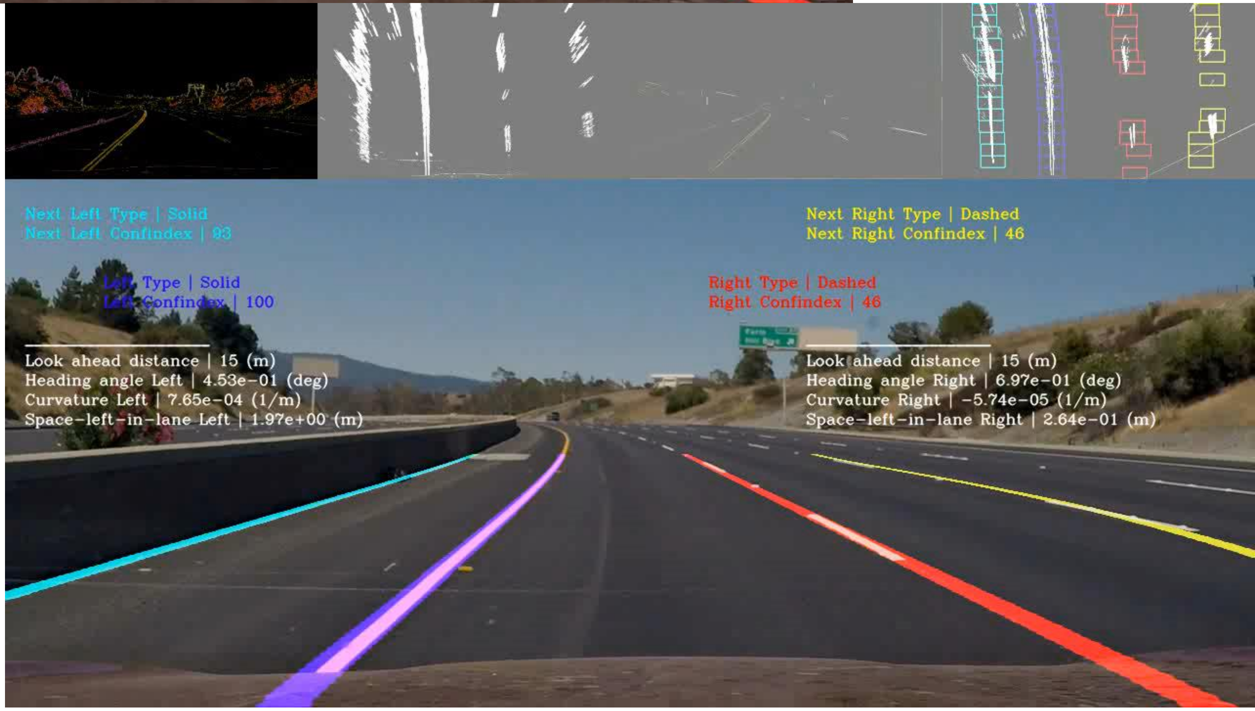
MULTILANE HIGHWAY (UDACITY DATASET)

- The Udacity dataset composes of a multilane highway with well defined features and less traffic occlusion
- Lighting is consistent and mount vibration is low
- Key Observation:
  - Local terrain changes affected ego lane estimation.
  - Complimentary filter able to handle continuity.



```
1.48e-05 | -3.81e-04 | 6.29e-03 | 2.29e+00
Next Left Type | Solid
Next Left Confindex | 93

Left Poly | 6.83e-06 | 7.54e-05 | 1.03e-03 | 1.01e+00
Left Type | Solid
Left Confindex | 100

-9.66e-05 | 6.09e-03 | -1.09e-01 | -1.90e+00
Next Right Type | Dashed
Next Right Confindex | 46

Right Poly | -1.06e-05 | 5.80e-04 | 2.67e-03 | -7.78e-01
Right Type | Dashed
Right Confindex | 46

Lane Change || None

Look ahead distance | 10 (m)
Heading angle Left | 2.63e-01 (deg)
Curvature Left | 5.61e-04 (1/m)
Space-left-in-lane Left | 1.94e+00 (m)

Look ahead distance | 10 (m)
Heading angle Right | 6.36e-01 (deg)
Curvature Right | 5.26e-04 (1/m)
Space-left-in-lane Right | 1.96e-01 (m)
```

Complimentary Filter disabled

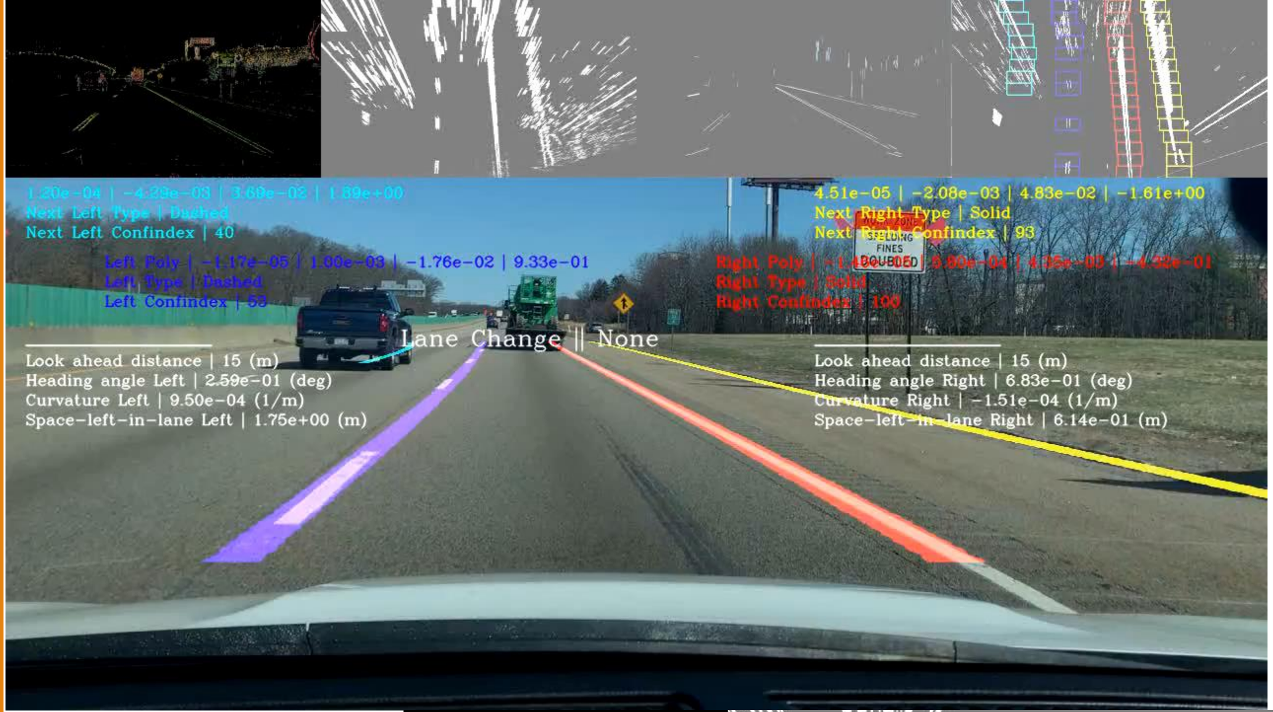


```
Next Left Type | Solid
Next Left Confindex | 93

Left Type | Solid
Left Confindex | 100

Next Right Type | Dashed
Next Right Confindex | 46

Right Type | Dashed
Right Confindex | 46

Look ahead distance | 15 (m)
Heading angle Left | 4.53e-01 (deg)
Curvature Left | 7.65e-04 (1/m)
Space-left-in-lane Left | 1.97e+00 (m)

Look ahead distance | 15 (m)
Heading angle Right | 6.97e-01 (deg)
Curvature Right | -5.74e-05 (1/m)
Space-left-in-lane Right | 2.64e-01 (m)
```

Complimentary Filter enabled

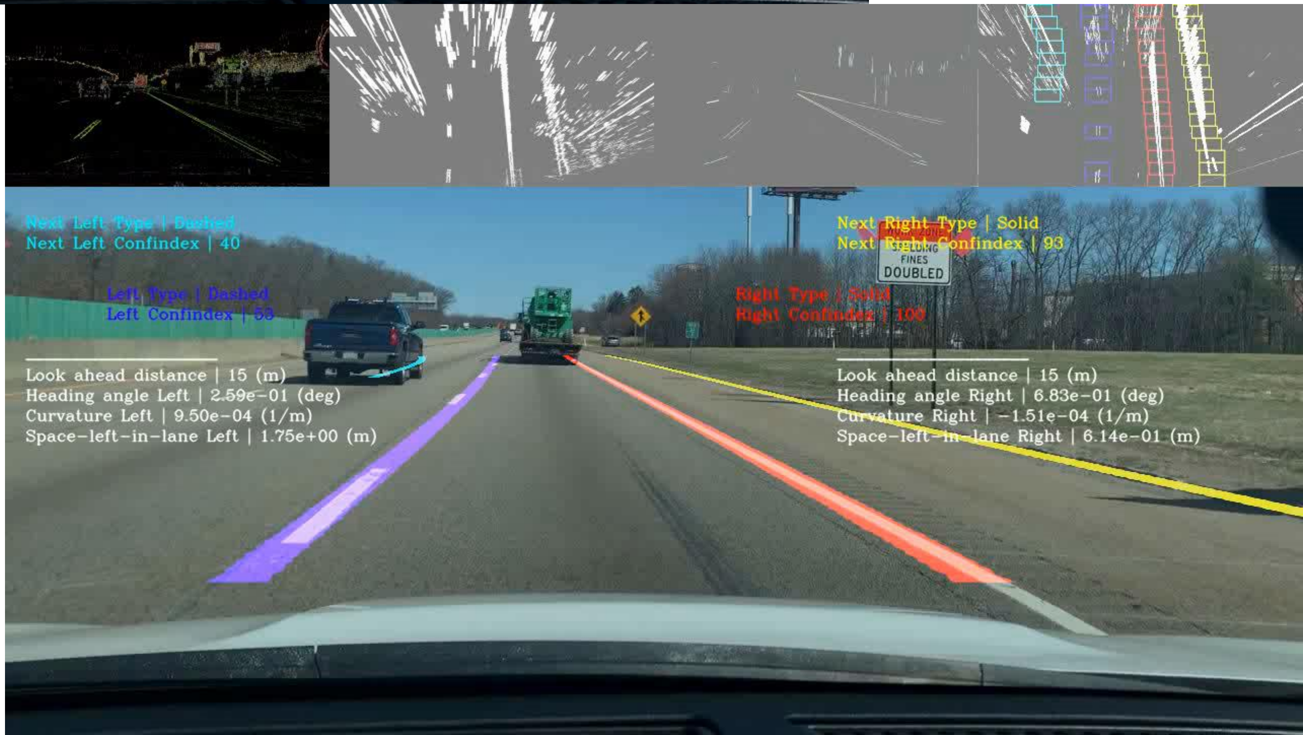
OCCLUDED MULTILANE HIGHWAY (OUR DATASET)

- Videos were collected using an iPhone camera mounted to the dashboard of a car
- Recorded data includes lane merging and vehicle occlusion on a stable lighting multilane scene
- Key Observation:
    - Vehicle occlusion within the ZOI causes poorly formed curves



```
1.20e-04 | -4.29e-03 | 3.69e-02 | 1.89e+00
Next Left Type | Dashed
Next Left Confindex | 40
Left Poly | -1.17e-05 | 1.00e-03 | -1.76e-02 | 9.33e-01
Left Type | Dashed
Left Confindex | 53
Lane Change || None
Look ahead distance | 15 (m)
Heading angle Left | 2.59e-01 (deg)
Curvature Left | 9.50e-04 (1/m)
Space-left-in-lane Left | 1.75e+00 (m)

4.51e-05 | -2.08e-03 | 4.83e-02 | -1.61e+00
Next Right Type | Solid
Next Right Confindex | 93
Right Poly | -1.48e-05 | 5.80e-04 | 4.35e-03 | -4.32e-01
Right Type | Solid
Right Confindex | 100
Look ahead distance | 15 (m)
Heading angle Right | 6.83e-01 (deg)
Curvature Right | -1.51e-04 (1/m)
Space-left-in-lane Right | 6.14e-01 (m)
```

Complimentary Filter disabled

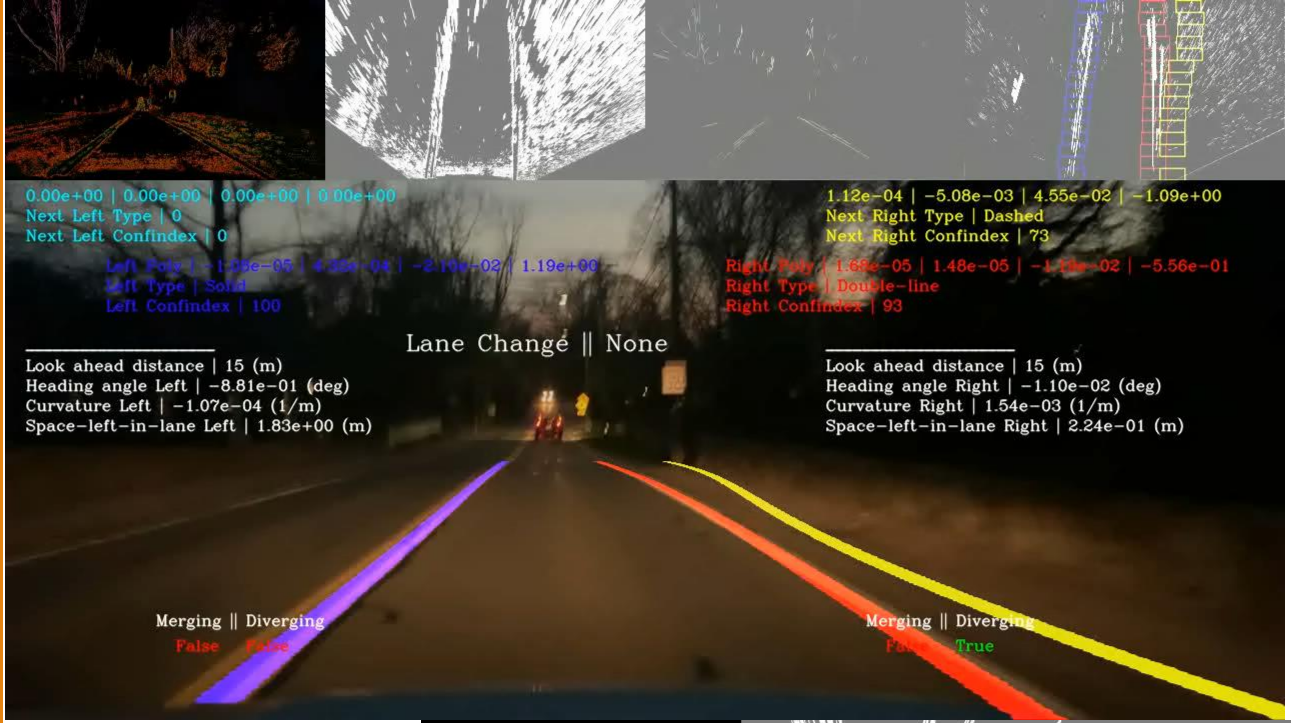


```
Next Left Type | Dashed
Next Left Confindex | 40
Left Type | Dashed
Left Confindex | 53
Look ahead distance | 15 (m)
Heading angle Left | 2.59e-01 (deg)
Curvature Left | 9.50e-04 (1/m)
Space-left-in-lane Left | 1.75e+00 (m)

Next Right Type | Solid
Next Right Confindex | 93
Right Type | Solid
Right Confindex | 100
Look ahead distance | 15 (m)
Heading angle Right | 6.83e-01 (deg)
Curvature Right | -1.51e-04 (1/m)
Space-left-in-lane Right | 6.14e-01 (m)
```

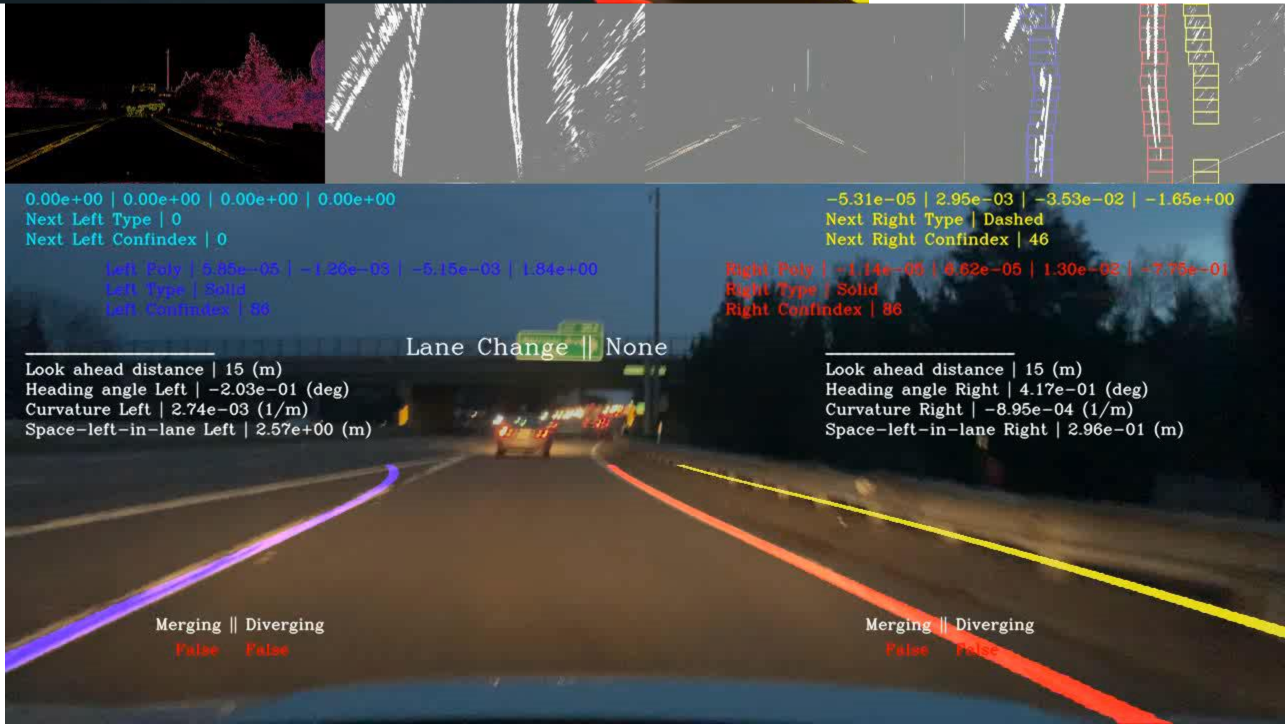
Complimentary Filter enabled

NIGHT SCENE (OUR DATASET)

- Similar recording setup
- Terrain is bumpy with highly shaky frames and poor lighting and locally bright regions
- Key Observation:
  - Too much jitter in curve formation due to locally dark or bright spots interfering with window sliding



```
0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00
Next Left Type | 0
Next Left Confindex | 0
      Left Poly | -1.08e-05 | 4.32e-04 | -2.10e-02 | 1.19e+00
      Left Type | Solid
      Left Confindex | 100

Look ahead distance | 15 (m)
Heading angle Left | -8.81e-01 (deg)
Curvature Left | -1.07e-04 (1/m)
Space-left-in-lane Left | 1.83e+00 (m)

Lane Change || None

      1.12e-04 | -5.08e-03 | 4.55e-02 | -1.09e+00
      Next Right Type | Dashed
      Next Right Confindex | 73
Right Poly | 1.68e-05 | 1.48e-05 | -1.10e-02 | -5.56e-01
Right Type | Double-line
Right Confindex | 93

      Look ahead distance | 15 (m)
      Heading angle Right | -1.10e-02 (deg)
      Curvature Right | 1.54e-03 (1/m)
      Space-left-in-lane Right | 2.24e-01 (m)

Merging || Diverging
  False     False

Merging || Diverging
  False     True
```

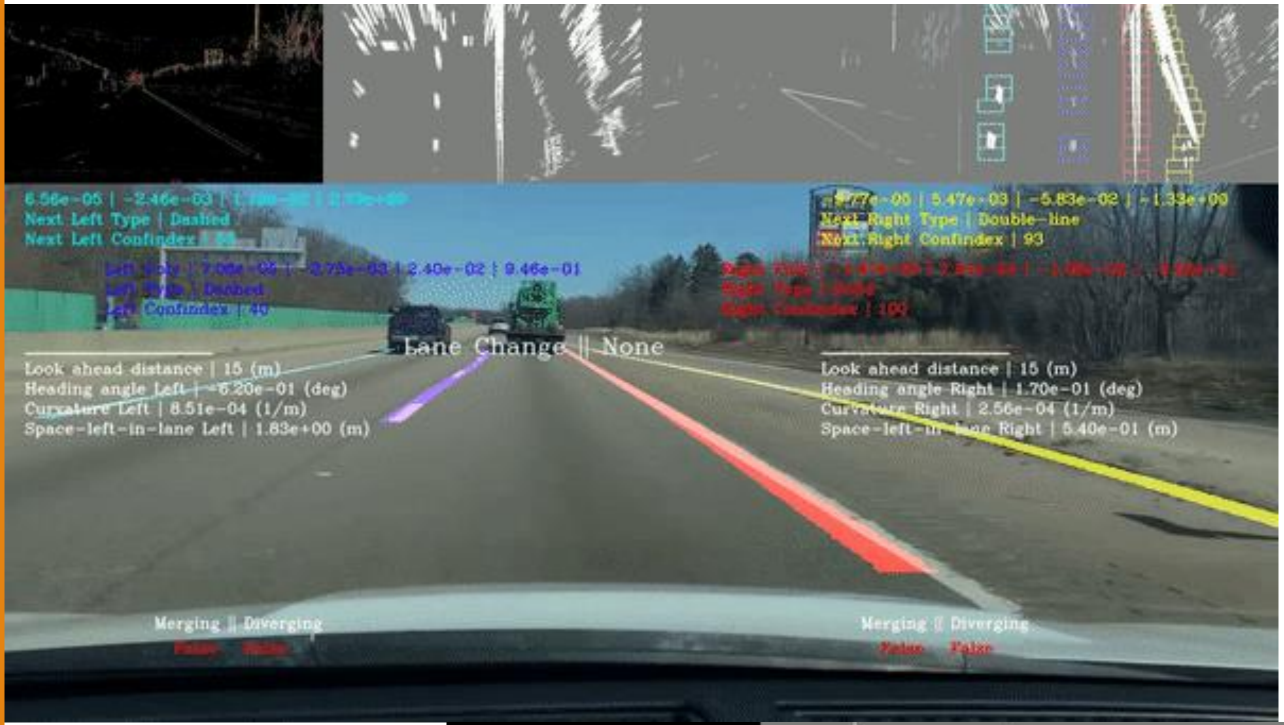


```
0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00
Next Left Type | 0
Next Left Confindex | 0
      Left Poly | 5.85e-05 | -1.26e-03 | -5.15e-03 | 1.84e+00
      Left Type | Solid
      Left Confindex | 86

Look ahead distance | 15 (m)
Heading angle Left | -2.03e-01 (deg)
Curvature Left | 2.74e-03 (1/m)
Space-left-in-lane Left | 2.57e+00 (m)

Lane Change || None

      -5.31e-05 | 2.95e-03 | -3.53e-02 | -1.65e+00
      Next Right Type | Dashed
      Next Right Confindex | 46
Right Poly | -1.14e-05 | 6.62e-05 | 1.30e-02 | -7.75e-01
Right Type | Solid
Right Confindex | 86

      Look ahead distance | 15 (m)
      Heading angle Right | 4.17e-01 (deg)
      Curvature Right | -8.95e-04 (1/m)
      Space-left-in-lane Right | 2.96e-01 (m)

Merging || Diverging
  False     False

Merging || Diverging
  False     False
```

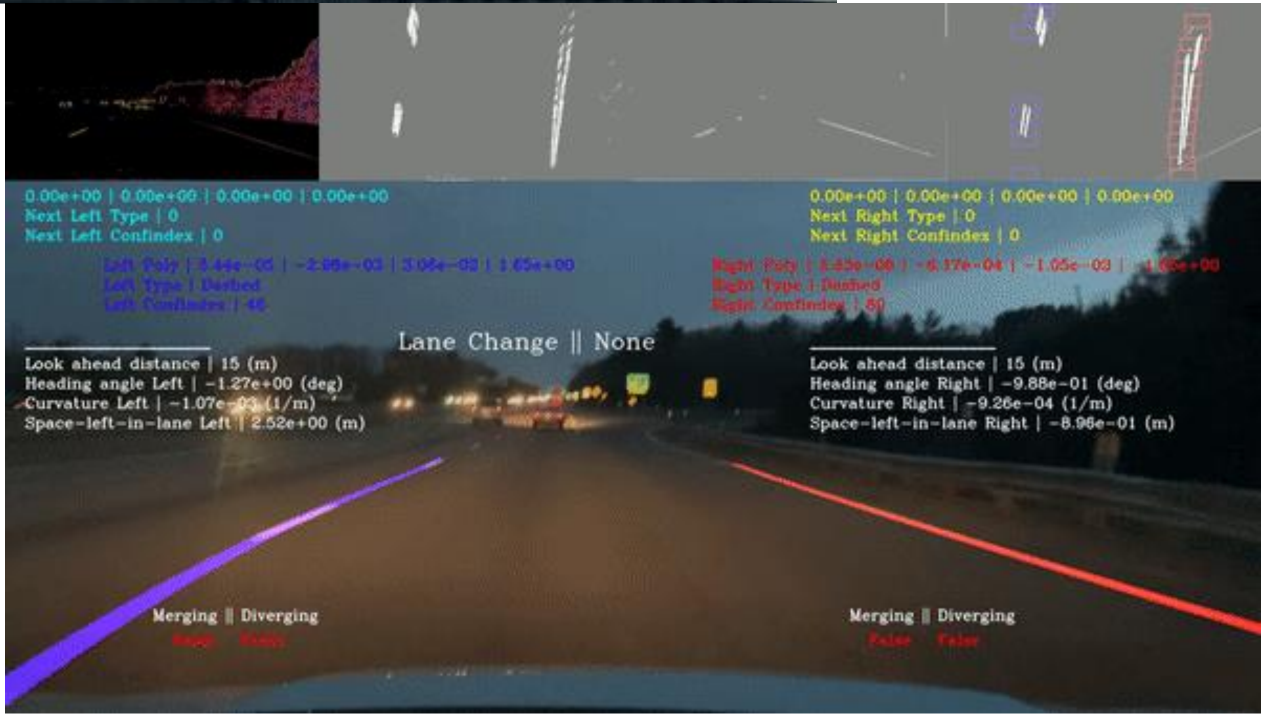
ADDITIONAL BEHAVIOR:

LANE MERGER OR DIVERGENCE

- Utilize ratio of distance curve endpoints and start points to determine if curves are converging or not
- Fails in cases when polynomials are poorly formed and their extrapolated projections intersect



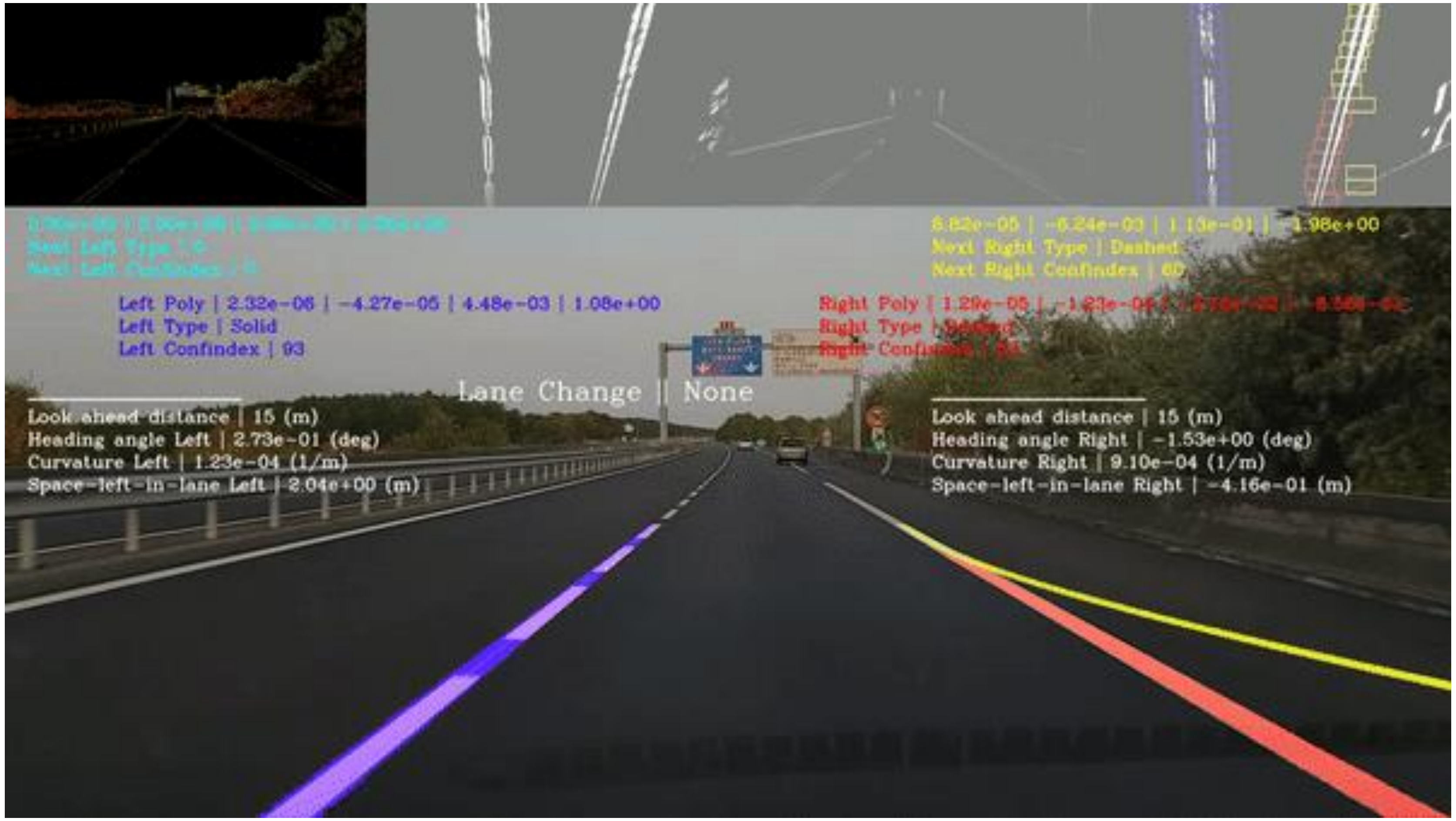
```
6.56e-05 | -2.46e-03 | [illegible] | [illegible]
Next Left Type | Dashed
Next Left Confindex | [illegible]
    Left Poly | [illegible] | -2.73e-03 | 2.40e-02 | 9.46e-01
    Left Type | Dashed
    Left Confindex | 40

Look ahead distance | 15 (m)
Heading angle Left | -6.20e-01 (deg)
Curvature Left | 8.51e-04 (1/m)
Space-left-in-lane Left | 1.83e+00 (m)

Lane Change || None

[illegible]7.77e-06 | 5.47e-03 | -5.83e-02 | -1.33e+00
Next Right Type | Double-line
Next Right Confindex | 93
Right Poly | [illegible]
Right Type | Dashed
Right Confindex | 100

Look ahead distance | 15 (m)
Heading angle Right | 1.70e-01 (deg)
Curvature Right | 2.56e-04 (1/m)
Space-left-in-lane Right | 5.40e-01 (m)

Merging || Diverging
False     False

Merging || Diverging
False     False
```



```
0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00
Next Left Type | 0
Next Left Confindex | 0
    Left Poly | [illegible] | -2.89e-03 | 3.06e-03 | 1.65e+00
    Left Type | Dashed
    Left Confindex | 46

Look ahead distance | 15 (m)
Heading angle Left | -1.27e+00 (deg)
Curvature Left | -1.07e-03 (1/m)
Space-left-in-lane Left | 2.52e+00 (m)

Lane Change || None

0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00
Next Right Type | 0
Next Right Confindex | 0
Right Poly | [illegible] | -6.17e-04 | -1.05e-03 | [illegible]
Right Type | Dashed
Right Confindex | 50

Look ahead distance | 15 (m)
Heading angle Right | -9.88e-01 (deg)
Curvature Right | -9.26e-04 (1/m)
Space-left-in-lane Right | -8.96e-01 (m)

Merging || Diverging
False     False

Merging || Diverging
False     False
```

ADDITIONAL BEHAVIOR:
LANE SWITCHING

Compares center axis distance from the closest two curves and generate a switching behavior in case of a lane change

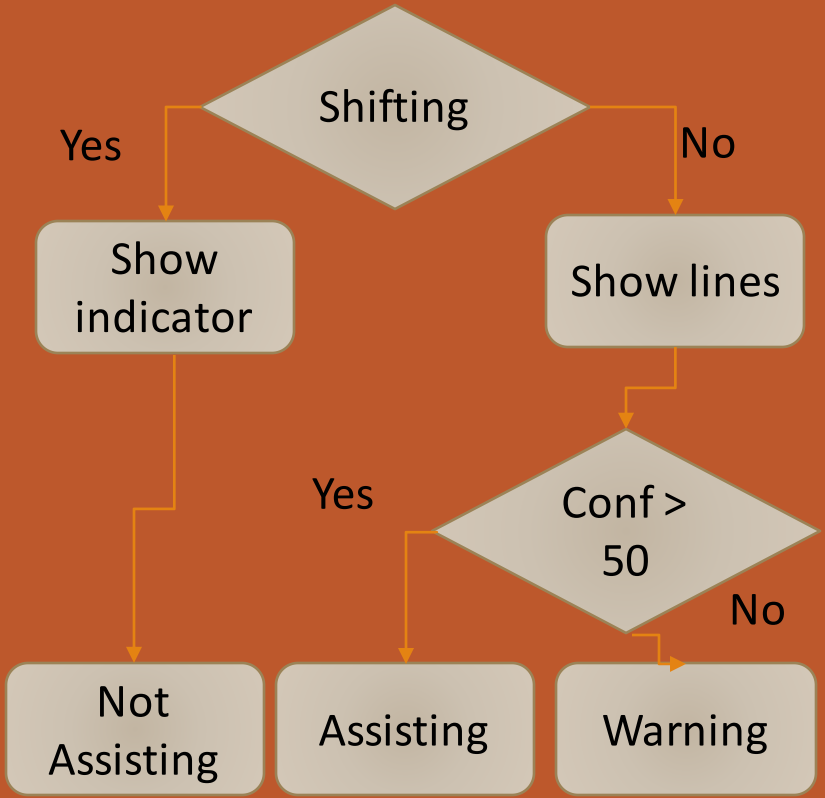


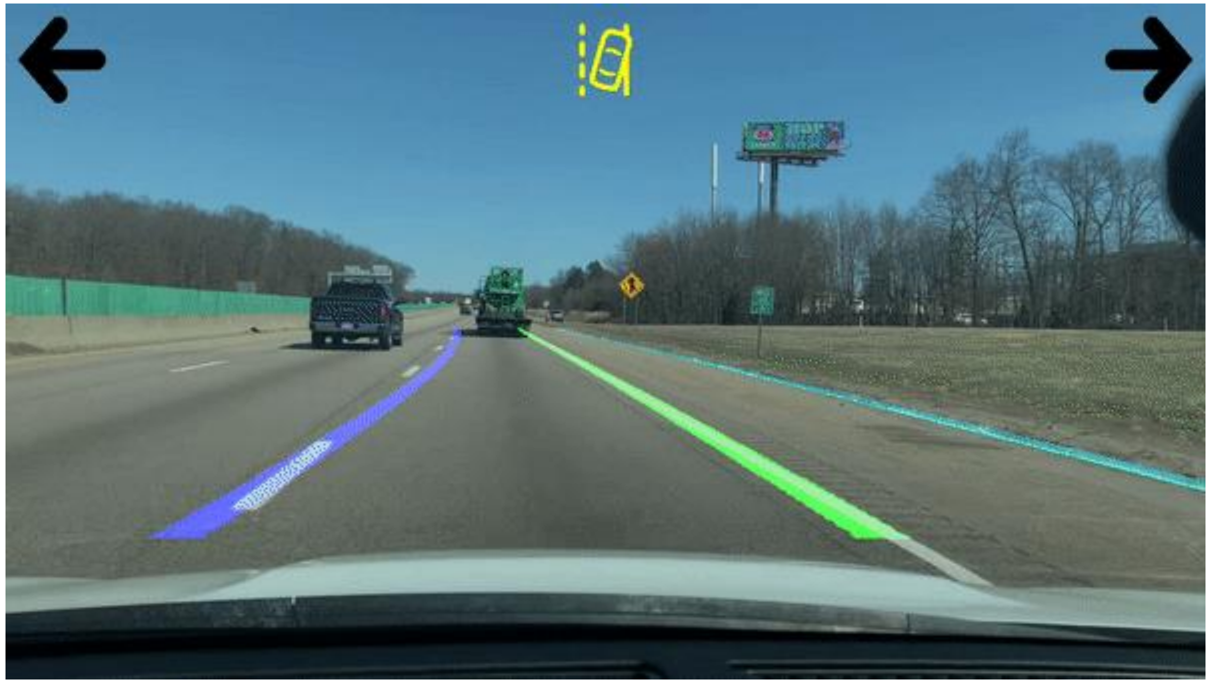
```
0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00
Next Left Type | 0
Next Left Confindex | 0

          Left Poly | 2.32e-06 | -4.27e-05 | 4.48e-03 | 1.08e+00
          Left Type | Solid
          Left Confindex | 93

Lane Change || None

Look ahead distance | 15 (m)
Heading angle Left | 2.73e-01 (deg)
Curvature Left | 1.23e-04 (1/m)
Space-left-in-lane Left | 2.04e+00 (m)
```

```
          8.82e-05 | -6.24e-03 | 1.13e-01 | 1.98e+00
          Next Right Type | Dashed
          Next Right Confindex | 60

Right Poly | 1.29e-05 | -1.23e-04 | [illegible] | [illegible]
Right Type | [illegible]
Right Confindex | [illegible]

          Look ahead distance | 15 (m)
          Heading angle Right | -1.53e+00 (deg)
          Curvature Right | 9.10e-04 (1/m)
          Space-left-in-lane Right | -4.16e-01 (m)
```

ADDITIONAL BEHAVIOR:

ROS UI DEVELOPMENT





BREAKDOWN OF FAILURE SOURCES

- INCORRECT CAMERA CALIBRATION
  - Poor radial distortion correction
  - Incorrect pixel to meter mapping for ego vehicle pose estimation
  - Noisy lane change, lane merge detection
- HIGH ROAD CURVATURE (SHARP TURNS AND SLOPED PATHS)
  - ROI failure due to manual calibration on a per scene basis (Remedy with adaptive ROI estimation)
- POOR LIGHTING OR LOCALLY BRIGHT SPOTS
  - Loss of discernible lighting gradient changes
- UNSTABLE MOUNTING
  - Shaky frames cause blurring and poor edge detection for low resolution
- VEHICLE OCCLUSION
  - False edge detection along the closer edge of the vehicle (being within the line angle threshold)
- NO LANE MARKINGS
  - Deep learning or IR based road detection to the rescue?
- COMPUTATIONAL MODULES
  - Sobel filtering is a costly affair and we do it twice per frame (could look into CUDA parallelization)
  - ROS Master – Single point failure

STARTUP CHECKLIST

- **HARDWARE:**
- Ensure camera is mounted properly as close to the center of the car as possible (if not update offset distance in settings.py)
- Ensure Camera lens is cleaned before start of every trip

- **SOFTWARE:**
- Make sure repo is cloned and virtual environment is set up properly
- Auto calibrate camera from a checkerboard before start of trip.
- Ensure pixel to meter mapping is accurate
- Make sure parameters such as car width, look-ahead distance is properly set in settings.py

FUTURE SCOPE

- Generating a custom wrapper for ED Lines with concurrency to speed up processing
- Adaptive ROI generation based on MSER
  - Extending to more lanes
- Develop low light pre-processing pipeline
- Work with hyperspectral IR data to extract regions of differing emissivity
- Sub-feature integration

THANK YOU FOR TUNING IN!

APPENDIX

RADIUS OF CURVATURE CALCULATION:

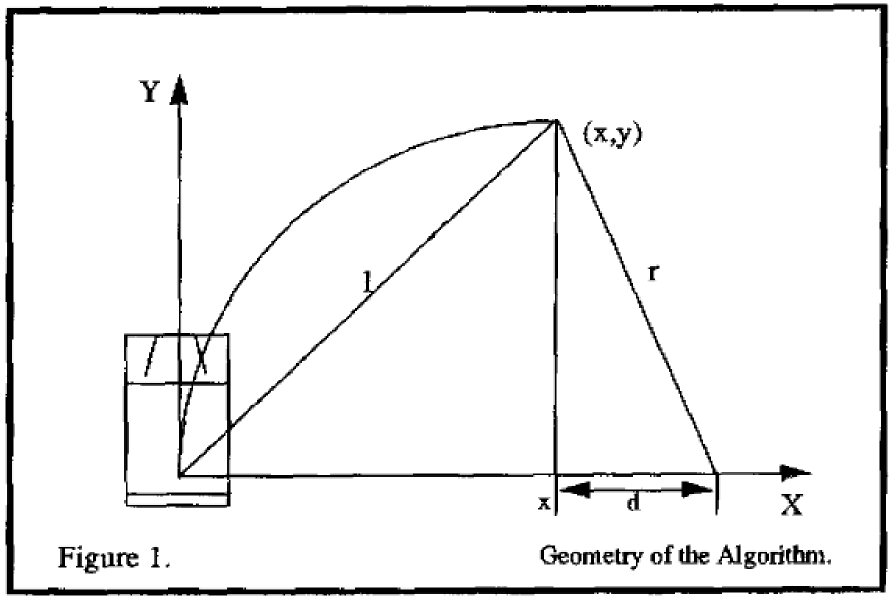


Figure 1. Geometry of the Algorithm.

$$d = r - x$$

$$(r - x)^2 + y^2 = r^2$$

$$r^2 - 2rx + x^2 + y^2 = r^2$$

$$2rx = l^2$$

$$r = \frac{l^2}{2x}$$

$$\gamma = \frac{2x}{l^2}$$

HEADING ANGLE: arcsec(y/x)