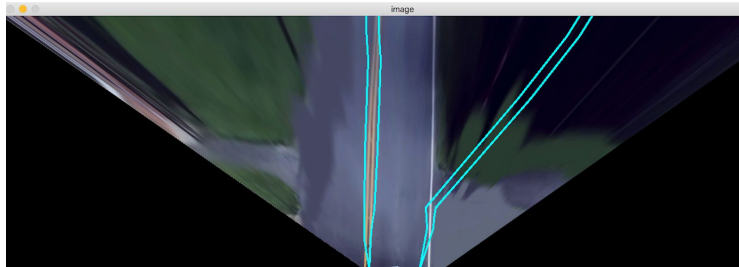


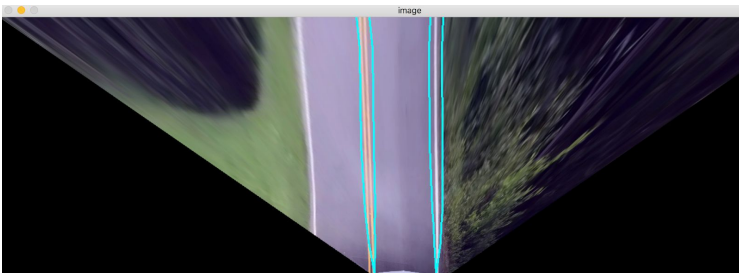
Key Idea:

The idea of this algorithm is to detect lane in the new frame based on the old frame.



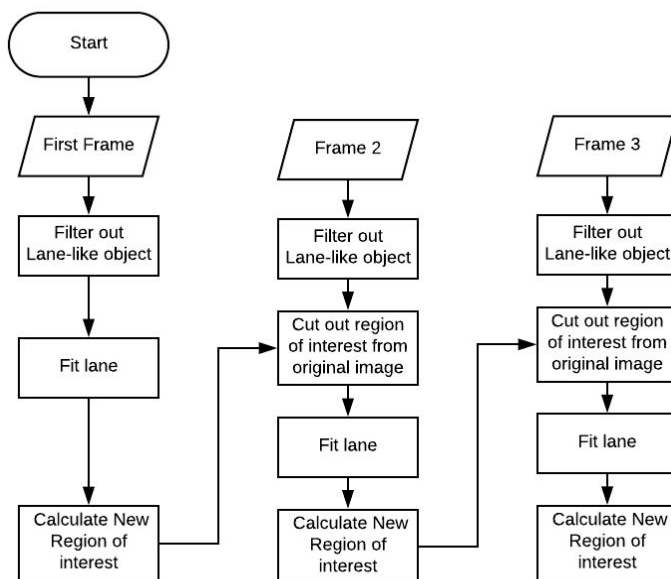
First frame detection is normal detection:

First frame detection might not be accurate. But the lane detector can auto correct itself.



The following frame will only find lane based on the region of interest of the last detection:

Area within blue lines are region of interest.



This is the flowchart of the program

High-Level description

Detect_video.cpp + Detect.hpp:

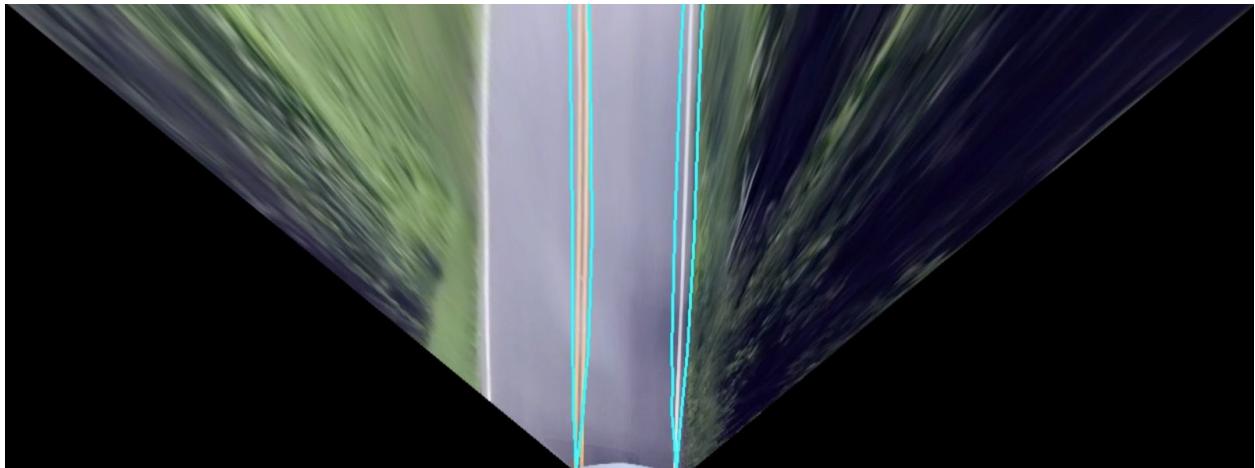
This file contains the main function. It take video input using command line argument.

There are 3 modes: ROBOT_TEST mode, REAL_ROAD mode and RECORD_RST mode. Mode can be changed in detect.hpp.

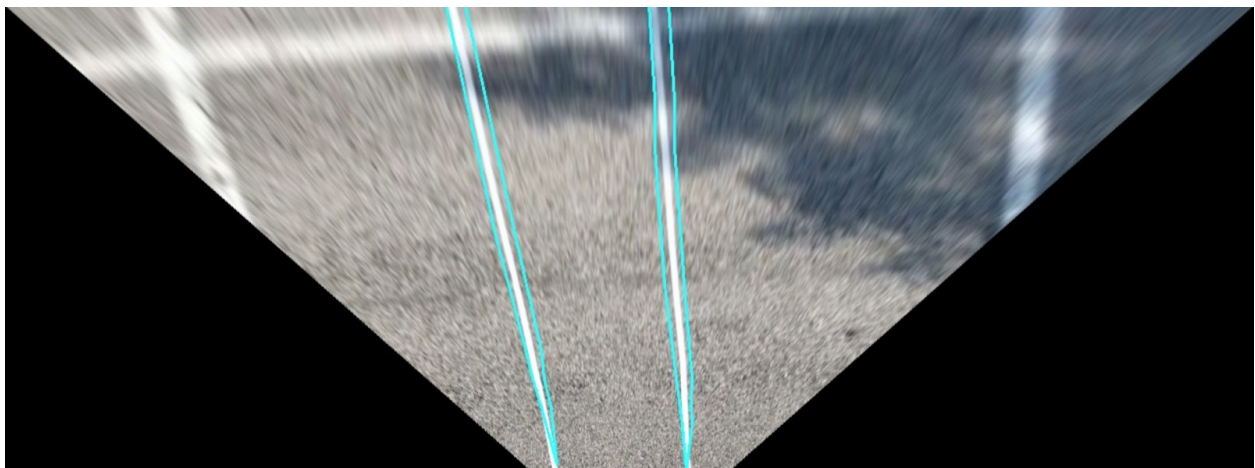
ROBOT_TEST mode is used for test on robot platform.

REAL_ROAD mode is used for test on real vehicle

RECORD_RST mode will take an input video and output the lane detection result as a video.



REAL ROAD mode snapshot



ROBOT TEST mode snapshot

Main:

Read in a video through command line argument. Enter a while loop to loop through each frame.

If consecutively 9 frames are missing, the program will exit on failure.

There are 2 kinds of lane detection algorithms.

For the first frame, run "first frame algo" to detect lane in whole image.

For each frame following, run "normal frame algo" to detect lane in region of interest.

If "normal frame algo" cannot detect lane for consecutively 200 frame. The program will switch back to "first frame algo"

Bool img_proc(Mat src, Mat&filter_frame_L, Mat&filter_frame_R,bool isFirst):

Src: input image

filter_frame_L: left lane area of interest

filter_frame_R: right lane area of interest

isFirst: check if it is the first frame. Different algo is used when it is first frame

Return: Return true when there is a lane detected. Return false when there is no lane detected.

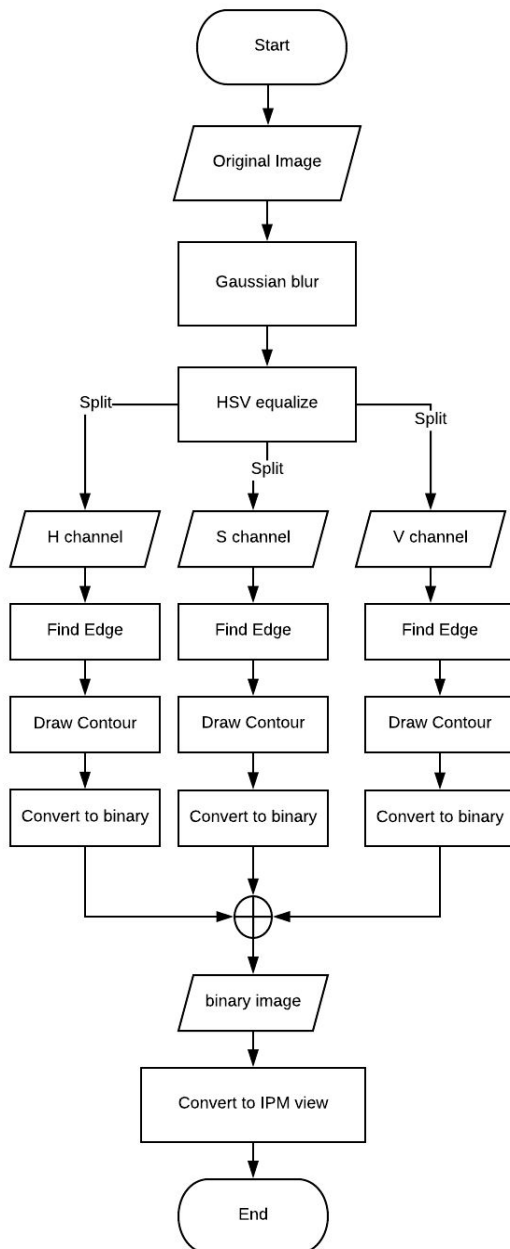
This function call the filter and use the filter result to detect white lane and draw the area of interest.

Preprocess.cpp + Preprocess.hpp:

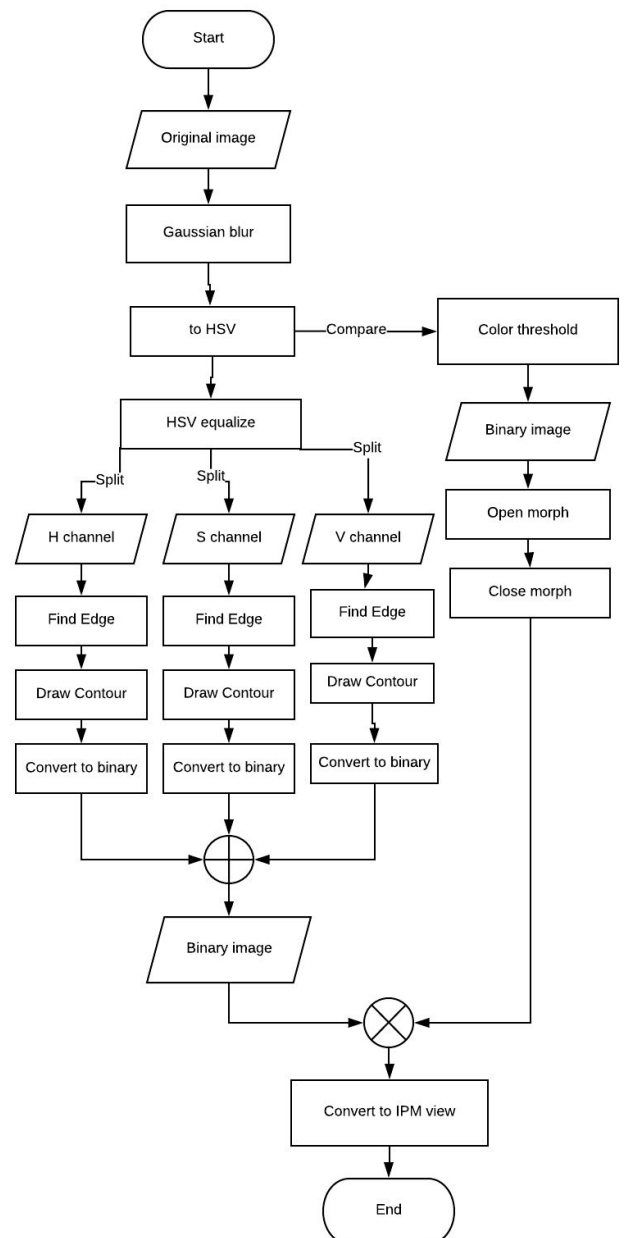
This file contains the filter design. The filter is implemented as a filter class.

For specific design, see diagrams.

Filter1 Design: Filter in REAL WORLD mode



Filter2 Design: Filter in ROBOT TEST mode



Function.cpp:

This file contains the algorithm to fit a line and calculate the region of interest.

Find_white_point:

The white point is the lane in binary image. This function loop through the image and find any white point in given image. Find the median horizontally and return the median value.

new_filter_valid_check:

Check if there is sufficient same area between the area of interest in current frame and last frame.

Draw_line_and_spread_function:

Draw line:

The idea is find 8 critical points and directly connect a line between these points.

The closer to the bottom of the image, the more dense critical points will be.

If the lowest critical point do not touch lower boundary, increase certain distance to lower boundary.

If the highest critical points do not touch high boundary, increase certain distance to higher boundary.

Draw spread function:

Draw spread function based on the 8 critical points. The spread function is a closed region and it will be used as the area of interest for the next frame.