ENPM673

Perception

Project 2

ADITYA VARADARAJ

(UID: 117054859)

(SECTION: 0101)

M.Eng. Robotics (PMRO),

University of Maryland - College Park, College Park, MD - 20742, USA.

Date of Submission: 4^{th} April, 2022

LIST OF FIGURES

CONTENTS

I. Problem 1: Histogram Equalization

A. Histogram Equalization

The pipeline I have used is shown in Fig. 1.

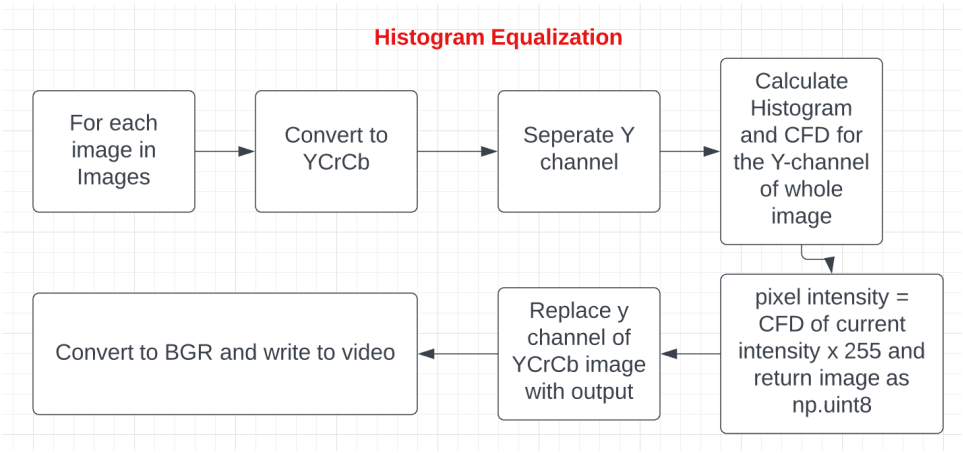

Fig. 1. Q.1.(a) Pipeline

One frame of the output is shown in Fig.2(b).

Output Video: https://drive.google.com/file/d/1r1d25Cu5DeLdzsDP_W7rD2CG0jN8odbf/view?usp=sharing

Fig. 2. (a) Original image (b) After Histogram Equalization

B. Adaptive Histogram Equalization

The pipeline used for Adaptive Histogram Equalization in Q.1.(a) is shown in Fig. 3.

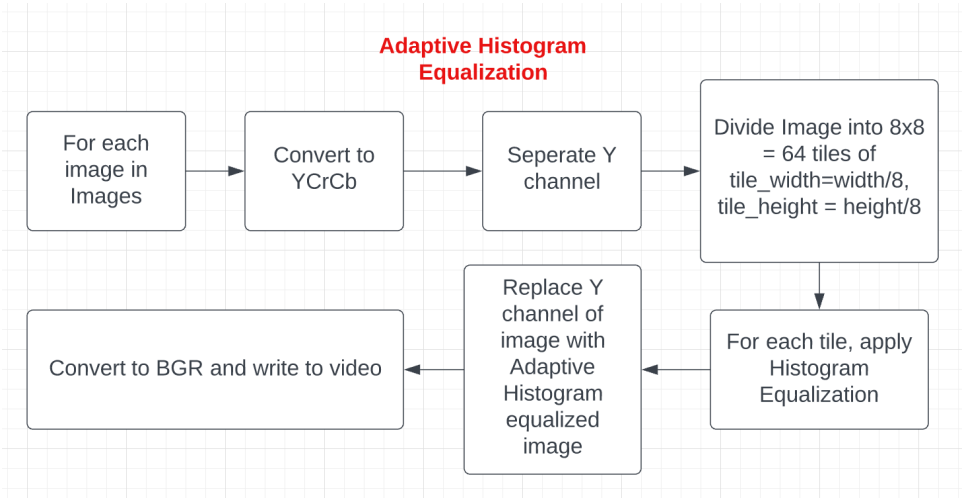

Fig. 3. Q.1.(b) Pipeline

One frame of the output of the adaptive histogram equalization is shown in Fig.4

Output Video: https://drive.google.com/file/d/1vaz-_dWDR3CXwEkVy5Kmbm8fsUIqVyoz/view?

usp=sharing

Fig. 4. Adaptive Histogram Equalization Output

II. Q.2. Straight Lane Detection

The pipeline used for Straight Lane Detection in Q.2. is shown in Fig. 5.

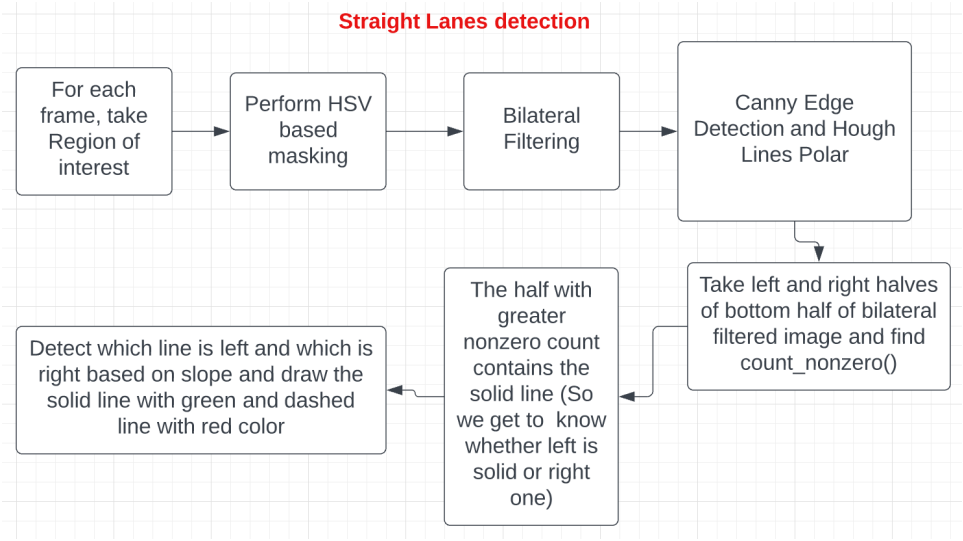

Fig. 5. Q.2. Pipeline

One frame of the output of the straight lane detection is shown in Fig.6

Output Video: https://drive.google.com/file/d/1drocDTWGv5Y8gkNg6oXJBxjQ4hZwYfBx/view?usp=sharing

Fig. 6. Straight Lane Detection Output

III. Q.3. Turn Prediction and Polynomial Fitting

The pipeline used for Turn Prediction, Lane Detection and Polynomial Fitting in Q.3. is shown in Fig. 7.

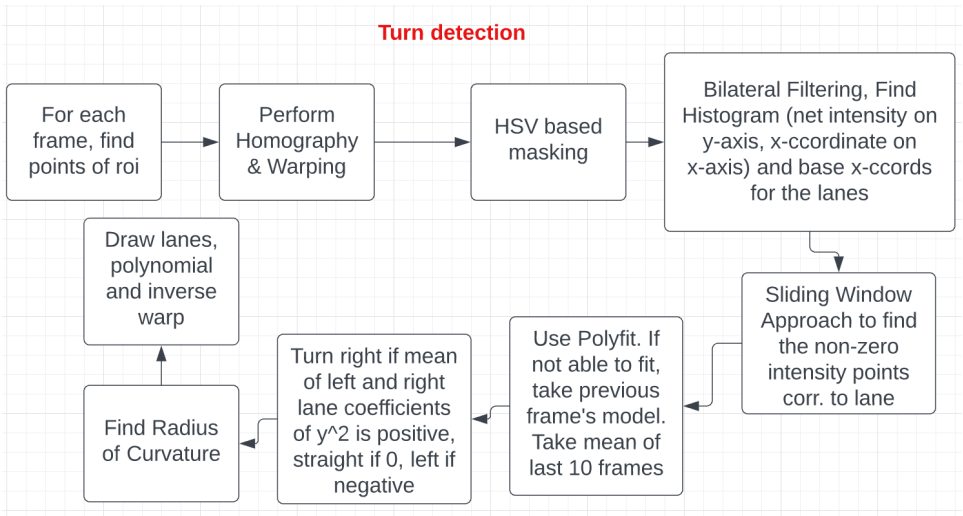

Fig. 7. Pipeline for Turn Prediction and Polynomial Fitting

One frame of the output of Q.3. is shown in Fig. 8.

Output Video: https://drive.google.com/file/d/1umWx7jV6fi6Mz5_9bqCDE_ltdA4mLf_9/view?

usp=sharing

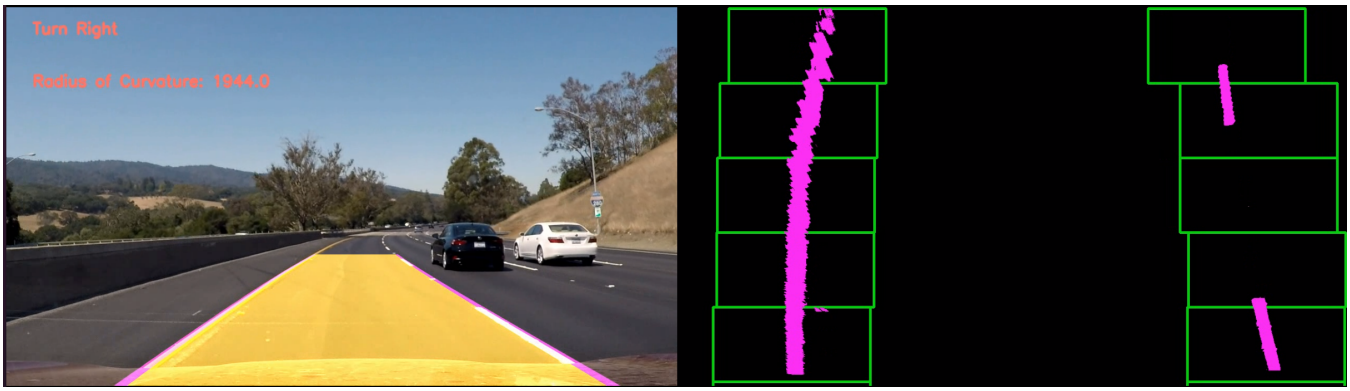

Fig. 8. Output of Turn Prediction and Polynomial Fitting (Left) and Sliding Windows Lane detection (Right)

IV. Homography Understanding and Usage

Homography matrix is a transformation matrix which transforms coordinates from one image plane to another. This matrix is computed using point correspondences. Minimum 4 point correspondences are required to compute the matrix. Once we get the matrix, it can be used to warp / inverse-warp the image (multiply coordinates by H or H^{-1}).

V. HoughLines Understanding

HoughLines Algorithm basically transforms lines from coordinates-space (image space) to parameter-space (a.k.a. Hough Space). A line (x, y) in image space is represented by point / set of parameters (ρ, θ) in Hough Space. Similarly, a point in image space is represented by line in Hough Space.