Advanced Lane Lines

objective:

The goals is to find and mark the lane lines on the road whatever lane lines are straight or curved using computer vision in order to be used in ADAS features like keep lane assist, also calculate the radius of the curvature

Reflection

1. Pipeline

My pipeline consisted of 4 steps:

1. Calibrate the camera using chessboard and store the calibration parameters into calibration_data.p in order not to calibrate the camera every time we run the program.
2. Read the test images and pass every image to processImage() function for the whole process
3. Read the test videos and pass every frame of every video to processImage() function for the whole process
4. ProcessImage() consists of a lot of stages:
   A. Undistort image using the saved calibration output (mtx,dist) → undistortImage(image,mtx,dist)
   B. Convert image to grayscale →convertImage(undistorted_image,cv2.COLOR_BGR2GRAY)
   C. Apply magnitude and direction gradients →applyMagDirThresholds(gray_image,3,30,100,0,np.pi/2)
   D. Apply color gradient using -→ applyColorThreshold(image,170,255)
   E. Combine all gradients together
   F. Convert the binary image to Bird Eye view using -→ GetBirdEyeView(combined_mag_dir_color_image)
   G. Find curvature of the lane lines using -→findCurvatures(bird_eye_binary_image,nwindows,margin,minpix,True)
   H. Measure the curvature using measureCurvatureReal(ploty,left_fit,right_fit)
   I. Fins the offset which is the same as curvature

All examples can be found in test_images_output

2. Potential shortcomings with your current pipeline

1. I can’t track the frames

3. Possible improvements to your pipeline

Solve my shortcomings