

Thanking for pointing that out.

To get the correct road center and offset difference, I fixed "function: get\_offset\_of\_centerline" first.

However, I found a more serious mistake in "function: get\_list\_img\_sizeYX"

My mistake is to define the inside and reverse width and height.

As a result, I noticed that the conversion rate and image center were shifted.

<Before correction>

img = io.imread('./camera\_cal/calibration1.jpg')

gbl\_img\_sizeX=img.shape[0]

gbl\_img\_sizeY=img.shape[1]

<After>

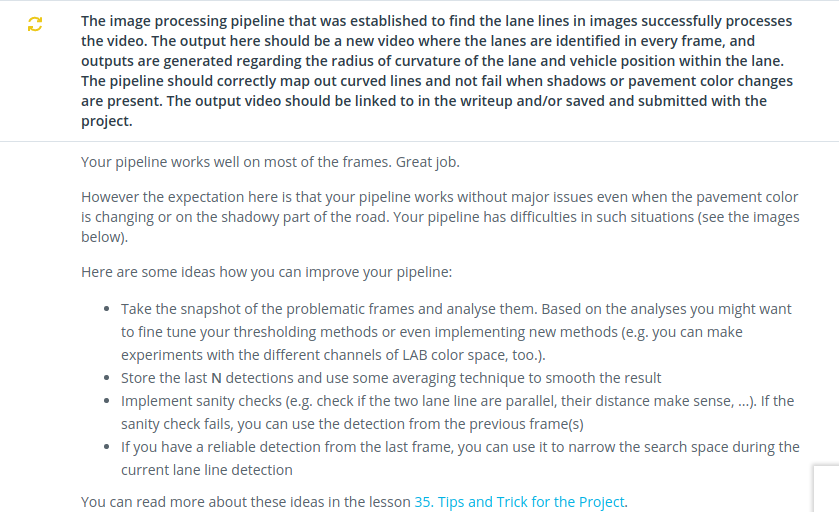
img = io.imread('./camera\_cal/calibration1.jpg')

gbl\_img\_sizeX=img.shape[1]

gbl\_img\_sizeY=img.shape[0]

※I also modified other parts that use get\_list\_img\_sizeYX

By correcting these mistakes, I was able to keep the deviation from the center of the road at a predetermined level.



In the Discussion, I mentioned about " dealing when lane can not be detected with good accuracy" and "dealing with detecting particles to obtain more lanes".

Although slightly, I have taken countermeasures for my two recommendations. As a result, erroneous detection of the running area could be avoided.

[Part\_A] . Added function to "dealing when lane can not be detected with good accuracy"

① Improvement of "the peak of histogram showing left and right lanes"

　→　　at　function of “ans\_project\_move\_function”　

The "peak of the histogram showing the left and right lanes" as the base point of the lowest slide is replaced with the calculated average value together with each of the left and right peaks detected the previous time to improve the accuracy

②Supplementation of detected particles

　→　　at　function of “ans\_project\_move\_function”　

In preparation for cases where binary particles required for lane detection can not be secured due to the influence of external light, the number of particles is supplemented using particles remaining in "binary\_warped" of the previous and last time.

③Correction when one lane can not be detected

　→　　at function of “polynomial2ndfit”

When obtaining the quadratic approximate expression of the lane, when there is a difference of 50 times or more between the left and right binarized particles necessary for detection, in order to improve the reliability of the approximate expression, " Approximate coefficient of lane "is overwritten using the other (approximate coefficient of the lane in which more particles are detected).

[Part\_B] Added function to "dealing with detecting particles to obtain more lanes".

①　Enhancement of lane particle detection processing by kmean method

　→　　at function of “image\_decolor\_debug”

I added a method to combine the detection process of road lane particles by the kmean method which I have done so far in duplicate.

The added window was set at the top of the image where the resolution declined, and the parameters were adjusted so that the number of new labels was changed.

In addition, the image\_decolor\_debug function also adds a masking process to remove particles generated from bumpers displayed at the bottom of the screen.