ENPM673: Project 2 Report
Lane Detection

Smriti Gupta
sgupta23@umd.edu

Varun Asthana
vasthana@umd.edu

Saumil Shah
sshah293@terpmail.umd.edu

***Abstract*— This project aims at detecting lanes from an image taken from the car's perspective. Images were warped such that they look like you are viewing it from the bird's perspective. Usually lanes are yellow or white in color, and thus images were made binary using color segmentation. Points on lanes were detected from this image and using curve fitting, lanes were predicted and highlighted in main image.**

I. Phase 1

Fig. 1: Original image

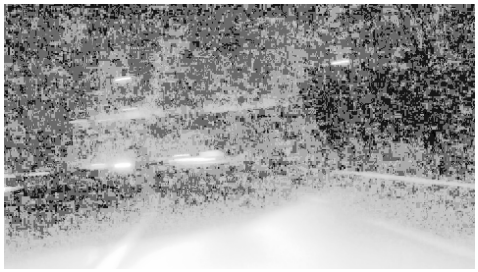

Fig. 2: Image after histogram equalization

Fig. 3: Image after histogram equalization on each color channel

This section gives an idea about histogram equalization of an image. This can be a very important tool to preprocess the image for a certain image processing task.

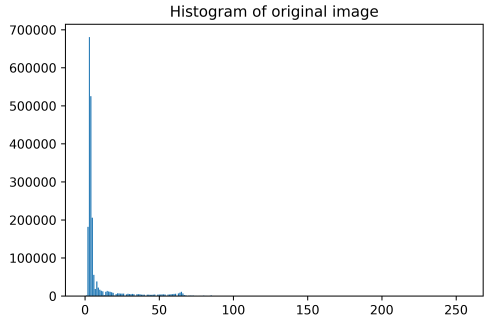

Fig. 4: Histogram of original image

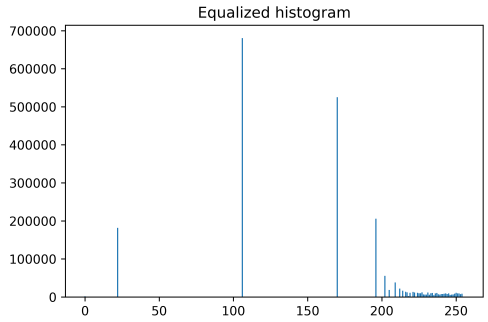

Fig. 5: Histogram of new image

Original image is shown in Fig 1. This image is quite dark and mostly useless for any other processing. Fig 5 shows image after histogram equalization applied to the original image. This newly generated image has too much noise in it, but we can see lanes in this image, which gives slightly better information than original image.

Histograms for both images are shown in Fig 4 and Fig 5. For colored image, histogram equalization can be applied to each color channel differently and new image can be created after merging all the new channels together. Histogram equalization applied to colored image can be viewed in Fig 3.

II. Phase 2

As prepossessing part of image, we have done histogram equalization in order to cover entire color range. This enhances white color which helps in color segmentation at later stage. After that it is necessary to use camera calibration

parameters to undistort the image. We will do everything explained in this section, on this undistorted image.

Fig. 6: Original frame

Using four point homography, all the images were warped. They were warped in such a way that they look from a perspective of a bird flying over the highway. After this perspective transformation of the original image shown in Fig 6, it will look like image shown in Fig 7. It is easy to do computation in this image, because lanes cover the entire top to bottom of image and both the lanes seems parallel in this view.

Fig. 7: Birds eye view of the given scene

In order to detect lanes from this image, we need some information, which we can use. It could be color of lane, shape of lanes or straightness of lanes. We know that lanes can consist of only two colors, white or yellow. This is very important information and can help us to detect lanes with very high confidence. We have used color segmentation using HSV color space. We have taken OR operator between two images, one consisting strongly white pixels and one consisting strongly yellow pixels. The resulting image will be binary image with white pixel value indicating yellow or white color and black pixel value indicating any other color values, which apparently are of no use for us. Generated binary image is shown in Fig 8.

Fig. 8: Color segmentation of image.

Now we have an image where lane pixels are white, but we need to segregate both left and right lanes, and we need to predict their curvature. We have initially taken vertical histogram of the image, which is shown in Fig 9. We can see two high bars in histogram shows left and right lanes. We can use this peaks as staring points for our predictions.This staring points are shown in Fig 10. Red lines shows the peak taken from histogram.

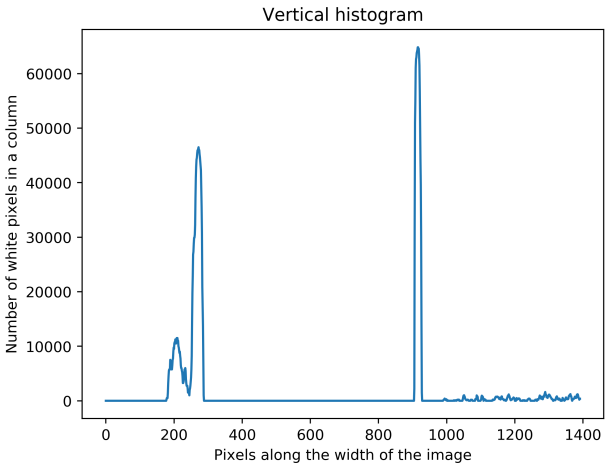

Fig. 9: Vertical histogram of the warped image. This histogram will be used to find the starting point of the lane.

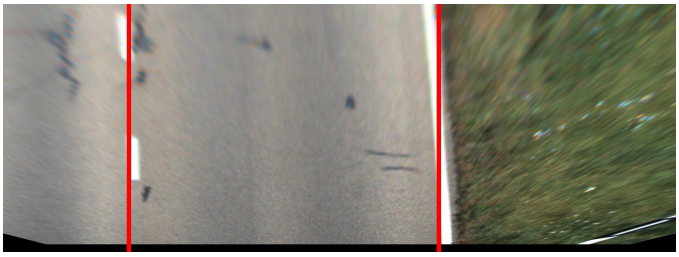

Fig. 10: Highest probability of lane pixels are here.

We have used sliding window technique to detect all the points lying on the lane. We have separated two left and right starting positions from mid point. For every patch, we count the white pixels in binary image shown in Fig 8. The centre point of next patch will be shifted to the mean position of the previous patch. This takes care of the turning road because the new position is constantly following the curvature. Fig 11 gives a clear idea about this algorithm.

We know the homography between original image and this warped image we have been using for detection so far. We can take the inverse of that homography matrix and track back all the lane points detected to the original image. After doing so, we have highlighted the detected lane with green color, with alpha value of 0.4. Fig 12 gives good visualization of the detected lane.

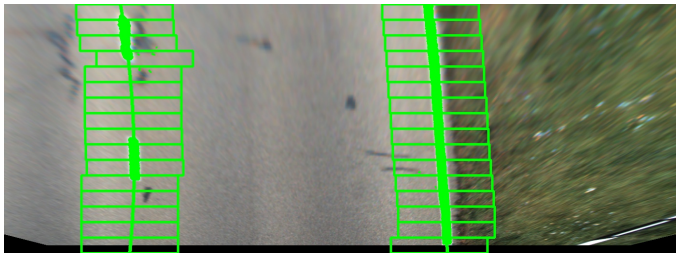

Fig. 11: Finding points lying on lanes using sliding window technique. Use this points for curve fitting.

Fig. 12: Reverse warp lane points to plot them on original image.

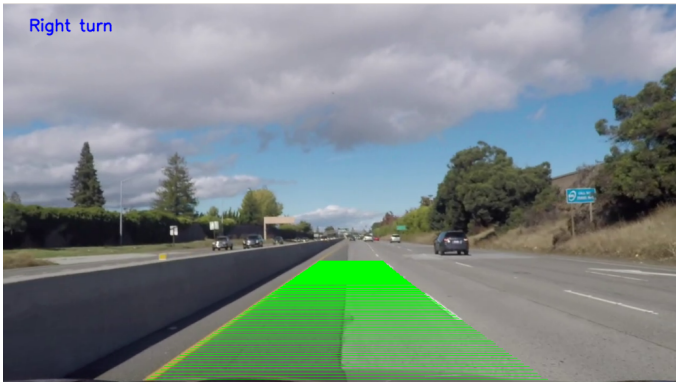

Fig. 14: Detected lane on challenge video

III. Results

We tried running the same algorithm on challenge video provided as the test data. Results were good and it was able to detect the turn as well. One more thing we observed was that our algorithm failed when camera had to detect lane in shadow. Because of shadow, it was unable to pickup the lane color and thus failed miserably.

Fig. 13: Sliding window on challenge video

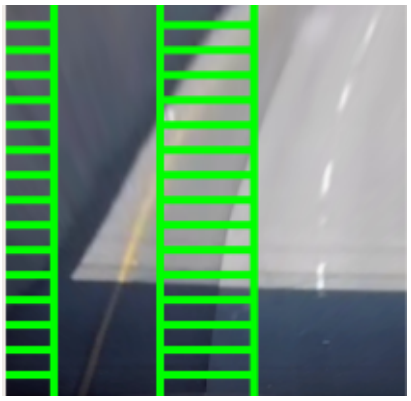

Fig. 15: Failed to find lane because of shadow.

IV. Conclusion

After seeing results of this algorithm, we conclude that it can be used with some modifications. One thing we can add is to use RANSAC based curve fitting for outliers rejection.

A. Output Videos

Videos can be found here.

Videos : `https://drive.google.com/open?id=1m7DoACe9wAIdiG-sPJY2SIj7e0nyzsSG`