A detailed study is made on the common characteristics of the images in Kitti Dataset[3], this information is used to simplify the previously developed algorithm. The following are a few notable things that appear in over eighty percent of the images available in the Kitti Dataset:

1. Roads have proper solid white lane line markings.
2. Roads have proper broken white lane line markings.
3. A four road junction appears for not more than 8 frames.
4. Only fifty percent of the image frame depicts the road(due to the mounting position of the camera).

Basing on the above-mentioned points, the following techniques or methods are incorporated into the algorithm.

1. Since only fifty percent of the image frame depicts the road, the entire image frame is first truncated into two halves(horizontally).
2. The solid white lane line markings can be used to separate the road from its adjacent surroundings, this in turn greatly reduces the noise that has to be filtered in order to detect the on road objects.
3. The picture frame can be divided into two halves(vertically), and the dominant line in both the halves can be found out using Hough transform, this process is illustrated in Fig. 7a&b.
4. The dominant lines found out in each frame are initially stored and compared with the previous values in the database, a new line is accepted or considered as lane only if it appears in consecutive frames(8 frames). This technique is especially useful to nullify bad and missing lane markings in certain image frames.
5. During every five iterations, the lines with least number of repetitions are discarded.
6. A filter is developed that can reduce the effect of illumination variations and broken white lane line markings.

Problems that were rectified due to the newly adopted techniques:

1. Reduction in the number of filters, thereby increasing the overall computation speed.
2. Reduction in the number of false detections, that were mainly caused due to signboards and other bright objects in image frames.

3. Discontinuity in object detections due to variations in surrounding brightness.
4. Failing to detect objects that have zero relative velocity(moving with the same speed and in the direction similar to the vehicle on which the camera is mounted).

Level wise description and output of the Algorithm:

1. Input image frame to the algorithm is shown in Fig. 2.

Figure 2: Input image frame to the algorithm

2. The truncated form of the input image is shown in Fig. 3.

Figure 3: the truncated form of Fig. 2

3. The truncated image is converted into a two dimensional quantity as shown in Fig 4.

Figure 4: the grayscale output

4. Edge detection filter is applied to find out the dominant edges in Fig. 4, the output of this operation is shown in Fig. 5.

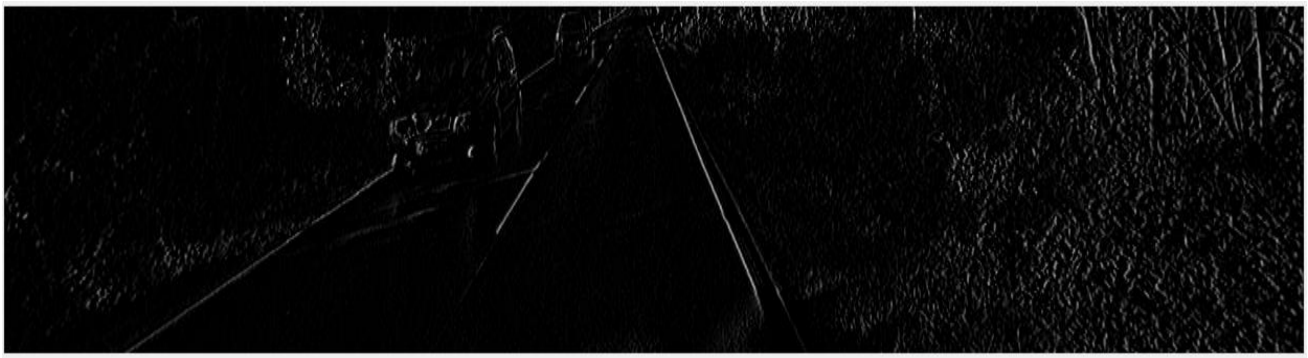


Figure 5: output from edge detection filter.

5. Now, Fig. 5 is subjected to a threshold operation where the values beyond the fixed threshold are treated as ones and the rest are made into zeros. The output of this operation is shown in Fig. 6.

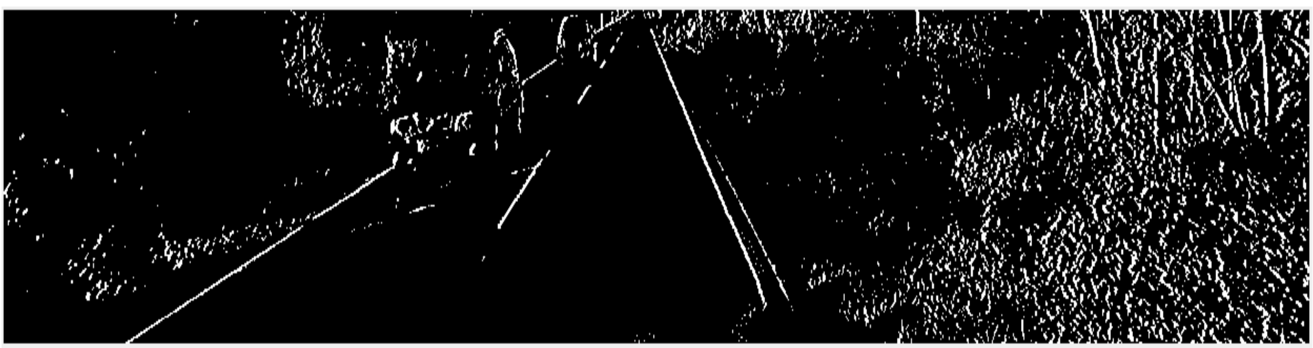


Figure 6: Output from threshold operation

6. Hough Transform is applied on the output in Fig. 6 and the output obtained is shown in Fig. 7a & 7b.

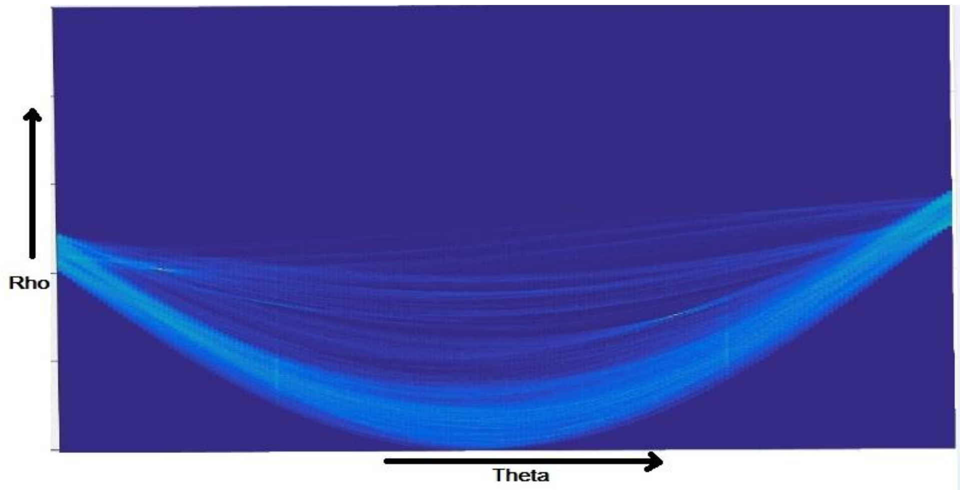


Figure 7a: plot of rho vs theta

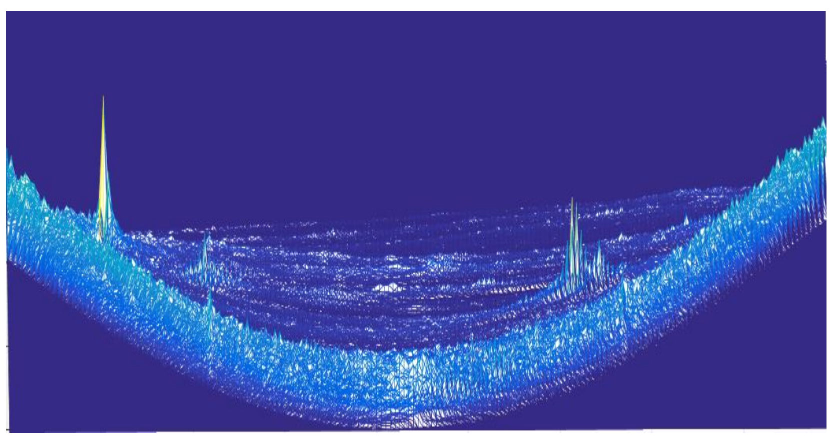


Figure 7b: frequency plot of rho vs theta

7. The rho and theta values having maximum frequency are found out from the peaks in Fig. 7b, these values correspond to the lane markings on the road. Then a polygon is drawn(pink colour) that fits in between the lines as shown in Fig. 8. The area occupied by the polygon corresponds to the area covered by the road.

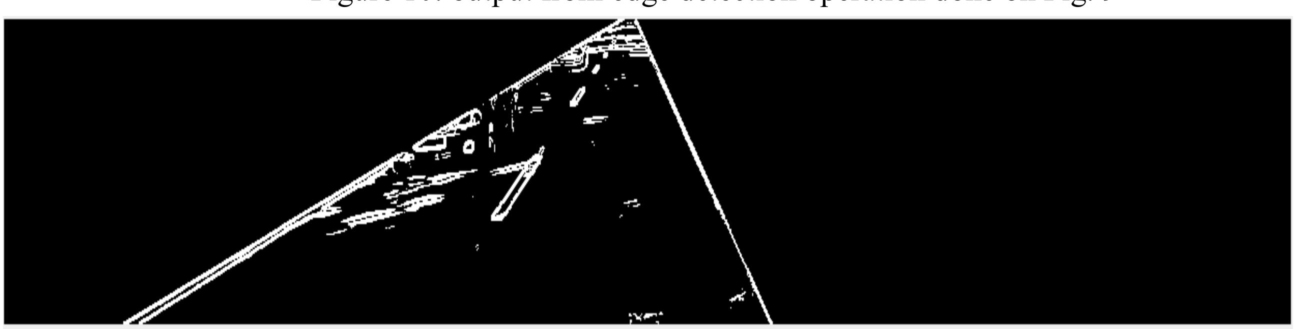
Figure 8: solid white lane line detection output

8. The road detected in Fig. 8, is separated from the rest of the image, the output of this operation is shown in Fig. 9.

Figure 9: road separation from surrounding area

9. Fig. 10 displays the output of the built-in edge detection algorithm, it can be seen that the illumination variations on the road cause a lot of disturbances making it almost impossible to detect the vehicle.

Figure 10: output from edge detection operation done on Fig. 9



10. In order to overcome the problem in step nine, a custom filter is developed to detect only the vehicles irrespective of the changes in illumination variations and broken white lane line markings. The output of the filter can be seen in Fig. 11 .

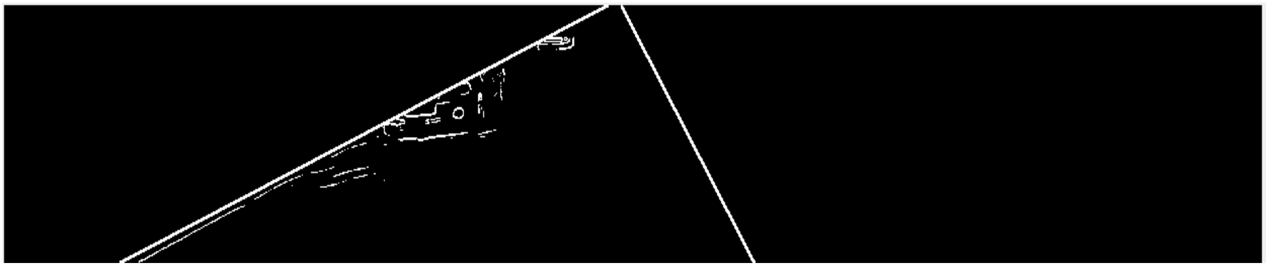


Figure 11: output from the custom designed filter

The designed filter smoothens the steep gradients in the colour patterns and at the same time compresses the values beyond a certain threshold(in this case it is 0.3). This process can be better understand by observing the following Fig. 12 a, b, c & d.

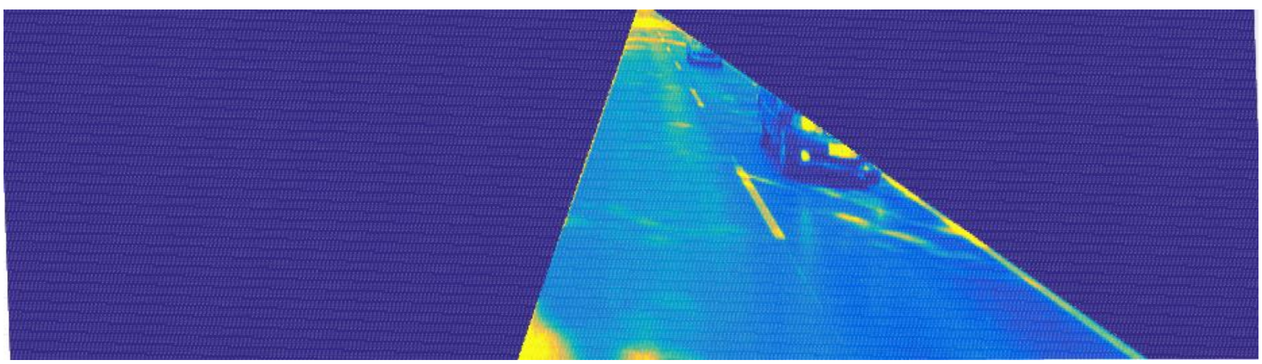


Figure 12a: brighter shades represented in yellow and darker shades are represented in blue.

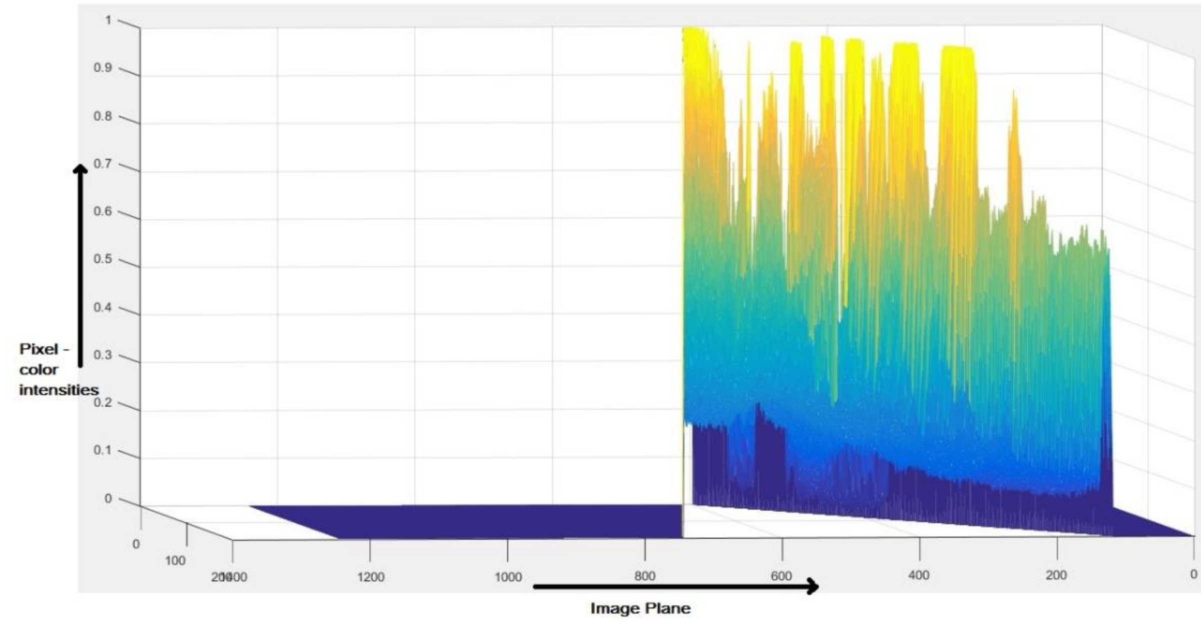


Figure 12b: variation in colour intensities

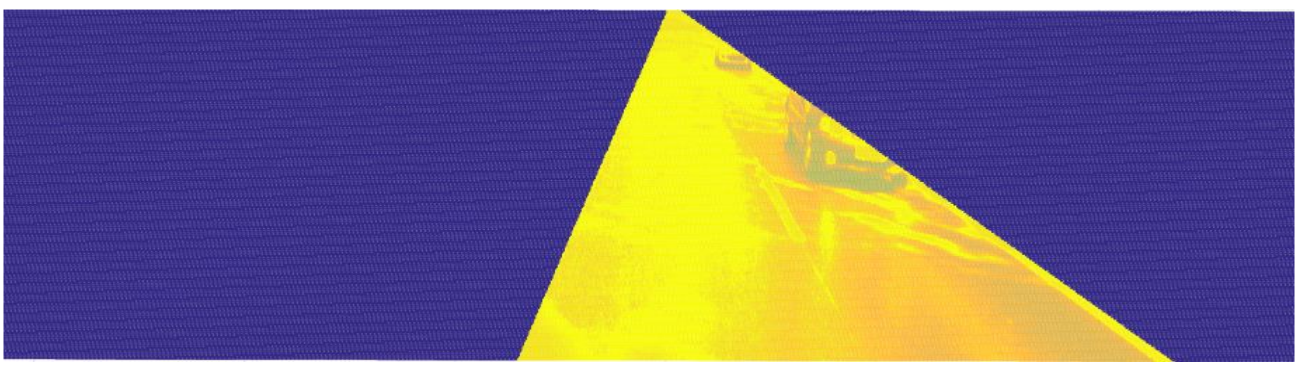


Figure 12c: output from the filter

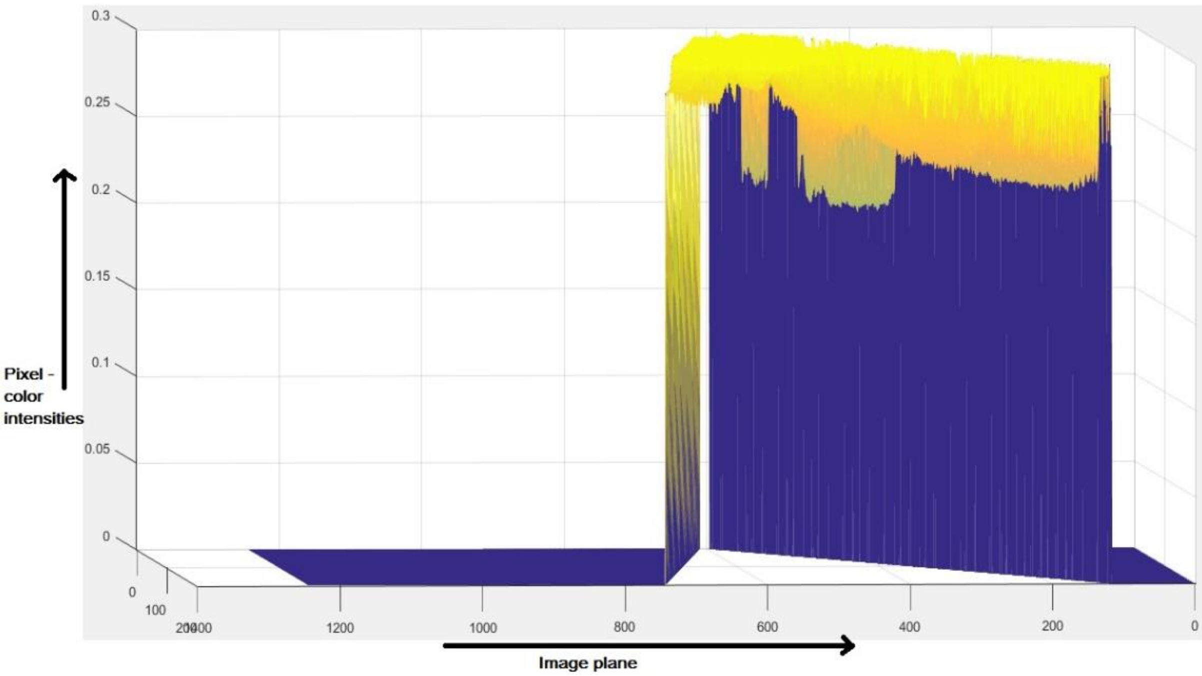


Figure 12d: reduced variation in colour intensities, max intensity is 0.3

11. The output shown in Fig. 11 is processed to remove the lane edges, this is being done in order to facilitate the erosion and dilation operations. The output of this operation can be seen in Fig. 13.

Figure 13: output after removing the lane edges

12. Area filtration(white spaces with an area less than 10 units is removed) and erosion operations are performed on the output in Fig. 13, this is being done in order to enlarge the area covered by the vehicles. The output of this operation can be seen in Fig. 14.

Figure 14: output from area filtration and erosion operations

13. Built in Matlab function called Blob Analysis is being used to find out the area and centre of the white spaces in Fig. 14 . The output of this function is a matrix consisting of centre coordinates and the height and width of the bounding boxes.

14. The output of the function in step thirteen is further processed to track the detected white spaces(vehicles), the detected white spaces are further processed to remove false detections, however, the efficiency of the current method used isn’t as expected. This has to be further improved to filter the unwanted detections. Bounding boxes are generated for the detected white spaces(vehicles), and the output is marked as shown in the Fig. 15.

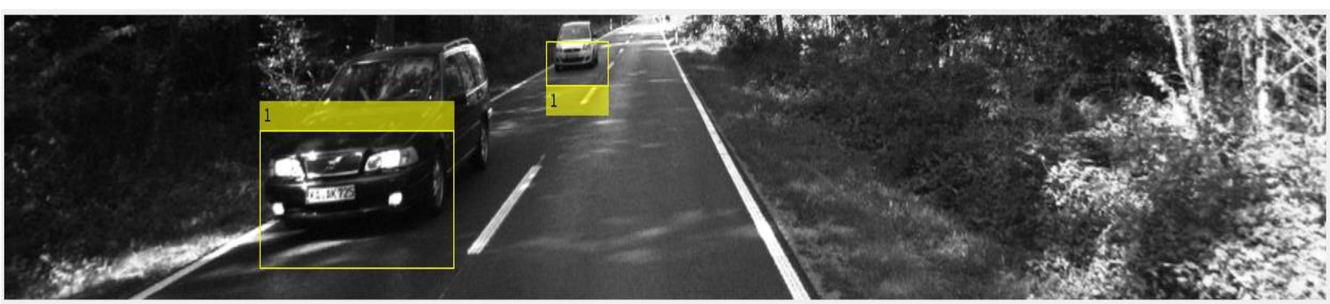


Figure 15: final output from the algorithm