



# Drivable Road Corridor Detection using Flood Fill Road Detection Algorithm

Karthik Shetty, Pratik Kanani

**Abstract:** Current image processing techniques for drivable road detection make use of lane markings. However, most roads lack lane markings which make such techniques obsolete. For such conditions, an image processing technique is required which identifies the boundaries of the road based on the color differences between the road and the surroundings. This paper proposes a flood fill road detection approach in which we first analyze a sample of the road and compute its RGB pixel distribution. The pixel range is used to detect the other road pixels in the image. Edge detection algorithms are then applied on the detected road to give road edge. It classifies the road on the basis of the visible differences between the road and its neighborhood. It allows for subtle color differences on the road surface, and unlike a color mask, due to the inherent growing nature of a flood fill algorithm, it does not detect neighborhood elements beyond the boundary having features similar to the road. This technique also manages to detect any obstructions on the road as opposed to other edge detection algorithms. We also propose methods to enable quick computation of otherwise expensive flood-fill algorithm. The method was tested on both marked and unmarked lanes and produced satisfying results for both images and videos.

**Keywords:** road detection, image processing, flood-fill algorithm

## I. INTRODUCTION

Convolution neural networks have proven to be a reliable method for road edge detection in self-driving cars. CNN architectures like VGG-16 and AlexNet have been able to label roads with high accuracy in the past[2]. They rely on training with large datasets of labeled road data. However, so far, they only perform reliably on roads with well-defined lane markings. As of right now, sufficient datasets are available only for city roads with lane markings, such as the Kitti dataset, which makes CNN's viable for road detection in cities. However, for unmarked country roads, there simply isn't enough labeled data to train our CNN's to work reliably on diverse terrain. This underlines the need for image processing techniques that can be universally used for both marked and unmarked lanes. Conventional image processing techniques such as canny and hough transform edge detectors again use lane markings and perform poorly on curved roads. This paper suggests a method that distinguishes the intrinsic visible differences between the road and its surroundings.

Revised Manuscript Received on December 30, 2019.

\* Correspondence Author

**Karthik Shetty\***, Department of Computer Engineering, Dwarkadas J. Sanghvi College of Engineering, University of Mumbai, India.

**Pratik Kanani**, Department of Computer Engineering, Dwarkadas J. Sanghvi College of Engineering, University of Mumbai, India.

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](http://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

The proposed algorithm implements a flood fill approach to find all the pixels of the image belonging to the road. We calculate the average and standard deviation of the pixel intensity of a sample of the road and label pixels of the image based on whether they satisfy the conditions. Since this is a color-based segmentation algorithm, it does not require lane markings and can work on any terrain as long as there is a visible difference between the road and its surroundings. It detects both straight and curvy roads accurately and does not detect the neighborhood having features similar to the road due to the inherent growing nature of the flood fill algorithm.

## II. PROPOSED METHODOLOGY

The paper follows the following steps to detect the road surface and extract edge information-

### A. Image Acquisition

Self-driving cars have multiple cameras that provide a continuous unobstructed feed of the road in front of the car. The drivable road is detected within these images which prompt the car to move accordingly. In this paper, we have set the resolution of the images to 350\*200. The reduced resolution facilitates the quick flooding of the road pixels which is explained later.

### B. Selection of Sample Space of image

We select a rectangular (or any other shape) sample of the road. This represents the distribution of the pixel intensities throughout the road. The placement of the sample must be such that throughout the video feed, no part of the surroundings should enter the sample. In our case, our rectangular sample was placed right in front of the hood. We selected a rectangular sample of size (125\*75) located at the bottom of the image.

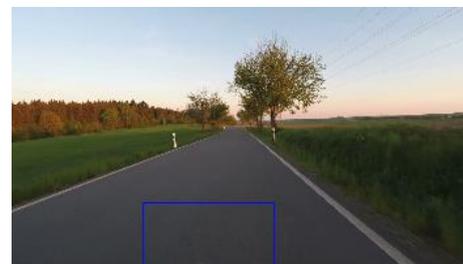


Fig 1. Selected sample space representing the pixel distribution of the road

### C. Calculation of the mean and standard deviation

We calculate the mean() and standard deviation() of all the pixel values in the sample for each channel (R-G-B).

We then set the satisfying condition for road detection of pixels for each channel(R-G-B) as-

$$\mu - \alpha \times \sigma \leq \text{Intensity}(\text{pixel}) \leq \mu + \alpha \times \sigma$$

Where,

$\mu$  is the mean of the sample's pixel intensities,

$\sigma$  is the standard deviation of the pixel intensities in the sample, and

$\alpha$  is the standard deviation factor

If the tested pixel's intensity values for all three channels lie within this range, the algorithm labels that pixel as the road.  $\alpha$  is taken as 2.5 based on experimentation.

### D. Flood-Fill Algorithm

It is a recursive algorithm that fills all the neighboring pixels of the seed point based on the given condition and then recursively performs the same operation with the neighboring pixels as seed points. The initial seed point lies inside the sample space.

```

flood_fill(seed_x, seed_y)
{
    if(condition is true)
    {
        set_colour(seed_x, seed_y)
        flood_fill(seed_x + 1, seed_y)
        flood_fill(seed_x, seed_y + 1)
        flood_fill(seed_x - 1, seed_y)
        flood_fill(seed_x, seed_y - 1)
    }
}
    
```

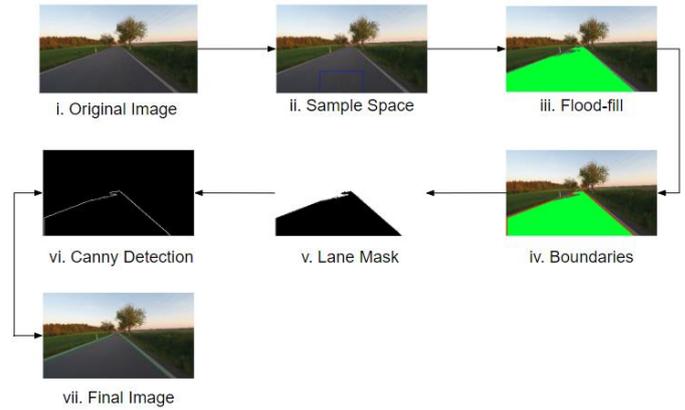
The advantage of using this algorithm instead of a simple mask is that a mask would label all the pixels of the image that lie within the permissible intensity range as a drivable road. This means that even the neighborhood of the road might get labeled as a road which is unacceptable.

The flood fill algorithm, on the other hand, starts from the seed point and works outward recursively. As soon as the intensity condition fails along the boundary, the algorithm stops spreading in that direction. This contains the spread of drivable road pixels within the confines of the road and provides reliable results in all types of terrain.

### E. Edge Detection

We then find the left and right boundaries of the drivable road. We can then fit a curve using a polyfit function or by using a canny and hough space edge detection algorithm.

In this paper, we first applied a lane mask to extract the entire road in the image. We then applied canny and hough transformation to get consistent road edges along the road. Other edge detection approaches like polyfit curve, bezier spline curve or Sobel edge detection algorithm can be used.



**Fig 2. Transformation Progression**

## III. OPTIMISATION

### A. Reduced resolution

Flood fill algorithm is a recursive algorithm that is slow in normal conditions as a large number of seed points are stored in the system stack which gets processed iteratively. To enable faster running time of the algorithm, we can reduce the resolution of the image. The algorithm does not depend on the minute details of the image and hence the details lost in the image by reducing resolution do not affect the accuracy. Reduced resolution means faster filling of the drivable road region.

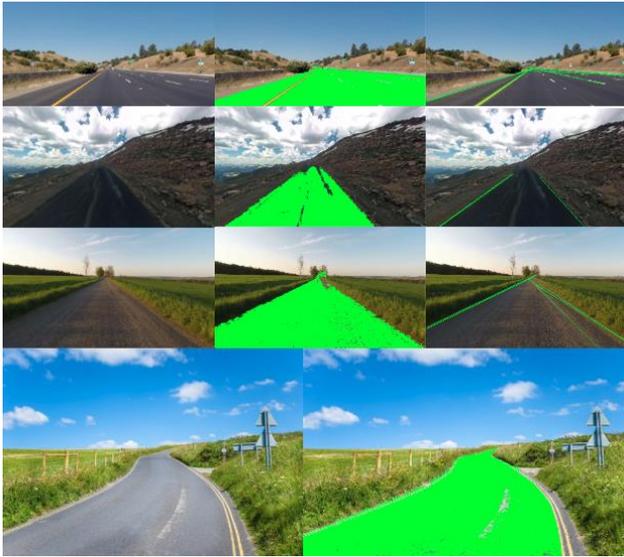
### B. Safety Nets

Sometimes in a video feed, an obstacle might enter the sample space and affect the mean and standard deviation values of the road sample. The sudden increase in standard deviation values would fill a large portion of unnecessary pixels as the sample space is no longer a true representation of pixel distribution of the road. To avoid this, we implement a safety net. A sharp increase in standard deviation values is assumed to be caused by an obstacle in the sample space. We store the previous mean and standard deviation values so that as soon as a sudden spike in standard deviation values is identified, we drop the new values and instead work with the old values until the sample space is uncontaminated again. This ensures the consistent working of the algorithm in all conditions.

## IV. RESULTS

The above algorithm was tested on both images and videos and produced desirable results. The technique produces desirable results on a wide range of different images and videos. It does not require lane markings and works well on any terrain. It produces accurate results for both straight and curved lanes.

The growing nature of the flood fill algorithm prevents the detection of the neighborhood having similar features as roads. It can accommodate subtle color differences and noise in the road surface, unlike contour detection algorithms.



**Fig 3. Results on different road surfaces. The technique uses visual clues for road detection.**

However it comes with certain drawbacks,

- A. Flood fill uses recursion which requires a lot of processor computation. This leads to a slight delay in the detection of road edges. This is especially problematic in case of a video feed. The algorithm cannot cope with the high frame rate of the video and hence does not produce real-time results. This can, however, be solved using a dedicated GPU.
- B. Due to the dynamic nature of driving, even through careful placement of the sample space, we can't guarantee the absence of obstacles in the sample space. Also, once set, the sample space cannot be changed.
- C. It performs poorly in low light conditions due to the absence of visible cues.

## V. CONCLUSION

The lack of lane markings is a serious hurdle for autonomous cars that current methods fail to address. This paper has hence showcased an efficient technique that manages to detect road corridors for both marked and unmarked lanes. The sample proves to be a satisfactory representation of the pixel distribution of the road. The growing nature of the flood fill algorithm enables it to differentiate the visible differences without labeling the noise present in the surroundings. The technique also manages to detect obstacles on the road and performs detection in real-time for a video feed.

## VI. FUTURE WORK

The flaws of the algorithm can be addressed through further study.

- A. The recursive approach of the flood fill algorithm makes it computationally expensive and slow. We need another algorithm which performs the same function but with lower time complexity. Perhaps an iterative version of a flood fill algorithm which does not require recursion since recursive functions make heavy use of the system stack.

- B. Furthermore, the static nature of the sample space makes it prone to contamination by an obstacle. Implementation of a smart dynamic sample space that automatically converges to a region that is surely part of the road.
- C. Further image processing techniques like contrast enhancement or transformation to another color space to further emphasize the differences between the road and its surroundings.
- D. A curve-fitting algorithm that accurately approximates the edges of the road. For Eg. a bezier spline curve can be used to fit the edges.
- E. Perspective transformation and vanishing point estimation to better localize the region of interest. Vanishing point refers to the point on the image where the road edges are seen to converge.
- F. Different Machine Learning Techniques can also be used to identify other objects and different patterns present in the driving scenarios. Goolge Colab can be used to accomplish such tasks. [18-19]

## REFERENCES

1. Annika Meyer, N. Ole Salscheider, Piotr F. Orzechowski, and Christoph Stiller, "Deep Semantic Lane Segmentation for Mapless Driving", 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, October 1-5, 2018.
2. Yichao Cai, Dachuan Li, Xiao Zhou, and Xingang Mou, "Robust Drivable Road Region Detection for Fixed-Route Autonomous Vehicles Using Map-Fusion Images", *Sensors*, 18(12):4158, November 2018
3. Chan Yee Low, Hairi Zamzuri, Saiful Amri Mazlan, "Simple robust road lane detection algorithm", 2014 5th International Conference on Intelligent and Advanced Systems (ICIAS)
4. Xingang Pan, Jianping Shi, Ping Luo, Xiaogang Wang, and Xiaoou Tang, "Spatial as Deep: Spatial CNN for Traffic Scene Understanding", The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18).
5. Vipul H. Mistry, and Dr. Ramji Makwana, "Survey: Vision-based Road Detection Techniques", (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (3), 2014, 4741-4747
6. Lamia Jaafar Belaid and Walid Mourou, "Image Segmentation: A watershed transformation algorithm", *Image Anal Stereol* 2009;28:93-102
7. Zuo-Quan Li, Hui-Min Ma, Zheng-Yu Liu, "Road Lane Detection with Gabor Filters", 2016 International Conference on Information System and Artificial Intelligence (ISAI)
8. Hui Kong, Jean-Yves Audibert, and Jean Ponce, "Vanishing point detection for road detection", 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2009.
9. Mingfa Li, Yuanyuan Li, and Min Jiang, "Lane Detection Based on Connection of Various Feature Extraction Methods", *Advances in Multimedia*, Volume 2018, Article ID 8320207
10. Yinghua He, Hong Wang, Bo Zhang, "Color-based road detection in urban traffic scenes", *IEEE Transactions on Intelligent Transportation Systems ( Volume: 5, Issue: 4, Dec. 2004 )*
11. Hendrik Dahlkamp, Adrian Kaehler, David Stavens, Sebastian Thrun and Gary Bradski, "Self-supervised Monocular Road Detection in Desert Terrain", *Robotics: science and systems*. Vol. 38. 2006.
12. Wijesoma, Wijerupage Sardha, KR Sarath Kodagoda, and Arjuna P. Balasuriya. "Road-boundary detection and tracking using ladar sensing." *IEEE Transactions on robotics and automation* 20.3 (2004): 456-464.
13. Fritsch, Jannik, Tobias Kuehnl, and Andreas Geiger. "A new performance measure and evaluation benchmark for road detection algorithms." 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013). IEEE, 2013.

14. Jochem, Todd M., Dean A. Pomerleau, and Charles E. Thorpe. "Vision-based neural network road and intersection detection and traversal." Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human-Robot Interaction and Cooperative Robots. Vol. 3. IEEE, 1995.
15. Unsalan, Cem, and Beril Sirmacek. "Road network detection using probabilistic and graph theoretical methods." IEEE Transactions on Geoscience and Remote Sensing 50.11 (2012): 4441-4453.
16. Caltagirone, Luca, et al. "Fast LIDAR-based road detection using fully convolutional neural networks." 2017 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2017.
17. Brust, Clemens-Alexander, et al. "Convolutional patch networks with spatial prior for road detection and urban scene understanding." arXiv preprint arXiv:1502.06344 (2015).
18. Kanani P., Padole M. (2018) Recognizing Real Time ECG Anomalies Using Arduino, AD8232 and Java. In: Singh M., Gupta P., Tyagi V., Flusser J., Ören T. (eds) Advances in Computing and Data Sciences. ICACDS 2018. Communications in Computer and Information Science, vol 905. Springer, Singapore
19. P. Kanani and M. Padole, "Deep Learning to detect skin Cancer using Google Colab", International Journal of Engineering and Advanced Technology (IJEAT), Volume-8 Issue-6, August 2019.

### AUTHORS PROFILE



#### **Karthik Shetty**

He is currently a Computer Engineering student at Dwarkadas J. Sanghvi College of Engineering, Mumbai. He has interests in the fields of Machine Learning, Deep Learning, and Computer Vision. He has gained experience through his R&D internship as an algorithm developer for autonomous cars and will be pursuing his masters in the future.



#### **Pratik Kanani**

Pratik received his Bachelors's and Master's Degree from the University of Mumbai. He has more than 28 research contributions in International and National Journals as well as Conferences. He is an active researcher in the field of Network Security, IoT, Machine Learning and Fog Computing.