**LANE DETECTION USING FISHEYE DATA**

**A report on**

**Computer Vision Lab Project**

**[CSE-3181]**

Submitted By

**BHAVYA MARUPURA & 210962168**

**MANISH REDDY & 210962164**

**RUSHENDRA REDDY & 210962198**

**SAI KRISHNA VAIBHAV & 210962200**



**[ V – SEM]**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**MANIPAL INSTITUTE OF TECHNOLOGY,**

**MANIPAL ACADEMY OF HIGHER EDUCATION**

**NOVEMBER' 23**

**LANE DETECTION USING FISH EYE DATA**

Bhavya Marupura1, Manish Reddy2, Rushendra Reddy3, Sai Krishna Vaibhav Mogili4

*1CSE AI ML – Manipal Institute of Technology, Manipal*

*2 CSE AI ML – Manipal Institute of Technology, Manipal*

*3 CSE AI ML – Manipal Institute of Technology, Manipal*

*4 CSE AI ML – Manipal Institute of Technology, Manipal*

[1 marupurub@gmail.com](mailto:1%20marupurub@gmail.com), 2 [manishreddy534@gmail.com](mailto:manishreddy534@gmail.com), 3 [rushibommu@gmail.com](mailto:rushibommu@gmail.com),  4 [krishnavaibhav2110@gmail.com](mailto:krishnavaibhav2110@gmail.com)

***Abstract –* This study presents a novel approach that combines fisheye camera calibration with OpenCV-based computer vision algorithms for robust detection of lanes from fisheye data. Lane detection is a critical component of autonomous vehicle navigation. Accurate lane detection is essential for the safety and autonomy of vehicles. While fisheye cameras provide a large field of view, they also introduce distortion, which reduces the precision of conventional lane detection methods. This work provides a viable method for lane detection in data from distorted cameras. Over the past few decades, an increasing number of approaches have been introduced in the literature review mentioned. This project provides two methods for correcting fisheye distortion in images: one is based on image segmentation, which has drawbacks, so we also developed a second method that uses camera calibration to provide precise lane markings.**

***Keywords – Lane Detection, Fisheye Data, Computer Vision, OpenCV, Image Segmentation,***

***Camera Calibration, Rectilinear Projection.***

1. **INTRODUCTION**

In recent years the automotive industry has become one of the most dynamic ones and it is continuously evolving. Therefore, a crucial element of a safe autonomous driving system is road environment perception. The development of new technologies is a major focus for businesses these days, and the majority of them share the primary objective of ensuring road safety for all users. driving. Lane detection is a key component of vehicle automation applications; it must be precise and stable in a variety of challenging circumstances, such as those involving heavy obstructions, obscured lane markings, or night-time. The capacity of fisheye cameras to record a much wider field of view has contributed to enhancing its popularity recently. Thus rendering them particularly beneficial for raising safety and situational awareness.

Fisheye lenses are extremely wide-angle lenses whose main goal is to provide images with a large field of view (FOV). However, this comes with significant radial and tangential image disturbances, the immediate consequence of having such a large field of view is how the distorted images that they generate require the use of specialized lane detection and tracking techniques. Traditionally, lane detection has been performed using standard perspective cameras, which provide a relatively narrow field of view. Unfortunately, these traditional methods of computer vision do not translate well to this kind of information.

In this project, we attempt to create two methods for lane detection using fisheye data obtained from the front and side fisheye cameras. The first method uses image segmentation to identify lanes directly from a fisheye-distorted image without any adjustments. However, this approach has several limitations, including the inability to identify lanes in situations where the input image is noisy, distorted, or contains excessive traffic. Therefore, we devise an additional method of detecting lanes that rectifies the distorted image into a rectilinear projection, which enables accurate lane detections. We achieve this by using camera calibration, hough line transform, and region of interest techniques to identify lane markings, which will be thoroughly covered in the Methodology unit ahead.

1. **LITERATURE REVIEW**

In the realm of driving assistance, camera-based lane detection is a fundamental and significant topic. A growing number of approaches have been published in the literature over the past few decades. But very few of them use lateral fisheye cameras.

**[3]** proposes the use of spatial layers and a U-Net-style architecture in a CNN S-UNet predictive segmentation instance model to efficiently detect and infer lines formed by lane markings from uncorrected side-view fisheye cameras. According to the experimental results, S-UNet can learn continuous prior and spatial relationships between lanes and achieve high accuracy and robustness in various real-world autonomous driving scenarios. It has a 5.57% mean IoU better than the state-of-the-art SCNN model [7], with a significant improvement in line marking detection accuracy.

**[1]** proposes the use of a fisheye side camera installed on the side of a car to observe the side scene in the Four Fisheye Camera Vision System (FFCVS) to detect lane markings. The two vanishing points of the direct lane markings can appear in the fisheye image, because the side fisheye camera has a wider FOV (Field Of View) than the hemispherical, which allows the entire lane markings to be observed on the side. In this paper, we first deploy the spherical stereo method to calculate the epipoles of a pair of fisheye images. The starting points of the vanishing strips are then determined by the epipoles. Finally, the estimated vanishing point locations and the slope values ​​of the edge points of the lane markings are used to identify the direct lane markings.

**[2]** focuses on lane detection problems using fisheye side camera images. It does so by first examining the theory behind the spherical model and the current state of the art, and then develops two approaches to solving this difficult task. While the former uses a convolutional neural network that was originally intended for image segmentation, the latter uses traditional computer vision to find Dark-Light and Light-Dark transitions and match them. Then the two methods are compared, with the second method being found to be better than the first.

**[4]** demonstrates the use of a single omnidirectional fisheye camera to detect voids. In this paper, we propose to use synthetic training data based on Unity3D as a solution. A five-pass algorithm is used to generate the virtual fisheye camera. This artificial training set is evaluated for use in free space detection in various topologies of deep learning networks.

**[8]** proposes a simple technique for calibrating a fisheye camera system while looking at a scene in a parking lot. Initially, a conventional parking spot line pattern is used to estimate each camera's position relative to the ground. Furthermore, the overlapping area of ​​the ground between nearby cameras is used to fine-tune the relative positions of the three cameras. Finally, the interactive interface allows manual adjustment of any slight deviation in the position of the camera system if necessary. The proposed method can reduce the user's workload because the fisheye camera system can be calibrated without having to prepare a special calibration pattern in advance for this specific purpose. The effectiveness of the proposed method is proven by experimental results.

1. **METHODOLOGY**

We developed two approaches for lane detection from fisheye data in this work, and we will go over them in detail as well as the experimental results of each method.

1. **Approach I:**

This is a simple method that detects the lane markings from fish-eyed data without adjusting the distortion by using the fundamental concepts of computer vision such as Gaussian Blurring, Grayscale Conversion, Binary Thresholding, Image Segmentation, etc. It uses the subsequent techniques. These involve the following methods.

1. Image Preprocessing:

* Importing Necessary Packages:

Import required libraries like OpenCV (cv2), Matplotlib (matplotlib), and NumPy (numpy).

* Image Loading and Cropping:

The input image is loaded using the OpenCV (cv2) library.

Slicing is used to crop the image in order to eliminate any disturbances from the original image.

* Grayscale Conversion and Gaussian Blur:

Apply a Gaussian Blue filter on the image using a (5x5) kernel.

Create a gray image by loading the grayscale version of the input image.

1. Image Segmentation:

* Binary Thresholding:

Apply binary thresholding to the original image and gray image using cv2.threshold() with a threshold value of 120 to make the lanes visible.

* Colour Segmentation:

Create a color mask to segment a specific color range.

Create a new colored image based on the segmented color mask.

We obtained the color-segmented image of both the Gray and RGB images to detect the lanes.

This method has several drawbacks, though, including the requirement to modify the thresholding parameters, the size while cropping, and the upper and lower bounds for the color masks according to the image. Each image's intensity has a significant impact on how well lanes can be identified. The image's noise, disruptions, and high traffic volume further impede the system's ability to identify lanes. As lane detection is a complex task, this approach doesn’t include complex techniques like Hough Transform, Edge Detection, Camera Calibration, etc. which would be incorporated in the following approach.

1. **Approach II:**

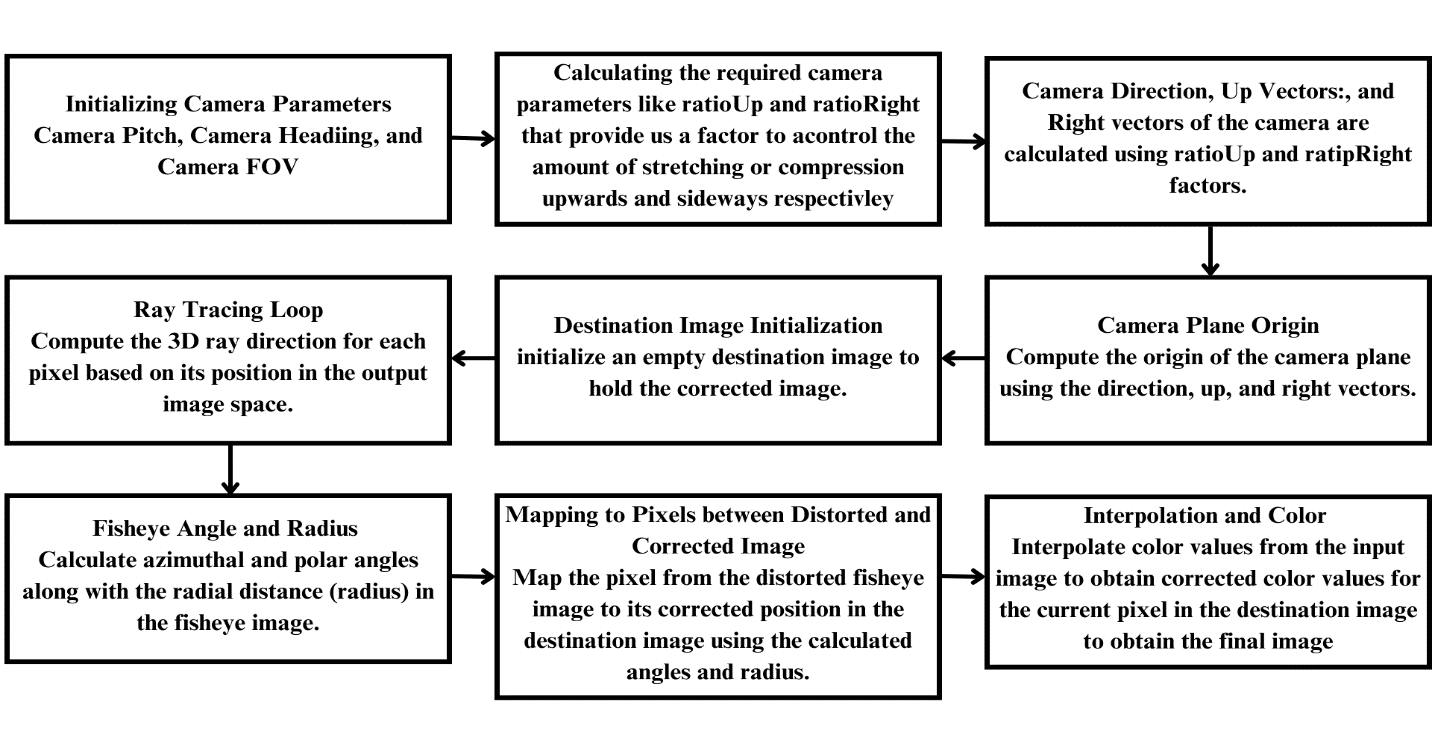
This method uses sophisticated and complex lane detection techniques for distorted input images. After acquiring a fisheye input image, it produces a rectilinear projection of the image to correct for fisheye distortion. Edge detection, region of interest, and hough transform techniques are then applied to further detect the lane markings. The following steps are involved in the methodology.

1. Importing Modules and Image Loading:

* Import required libraries like OpenCV (cv2), Matplotlib (matplotlib), and NumPy (numpy).
* The input image is loaded using the OpenCV (cv2) library.

1. Distortion Correction: Develop a rectilinear projection of the input image. **Fig.1**

* Implement fisheye distortion correction based on camera orientation, field of view, and other camera parameters.
* Initializing and calculating the required camera parameters.
* Create an empty destination image to hold the corrected image.
* For each pixel in the destination image, we have calculated the corresponding pixel in the distorted image using geometric transformations.
* Interpolate color values from the distorted image to the corrected image.
* Give back the input image's rectilinear projection (i.e. the distortion image that is adjusted).



**Fig.1 Algorithm used for correcting a distorted image**

1. Region of Interest (ROI) Masking:

* For highlighting a specific region of the image and suppressing the remaining part of the image.
* Polygon Vertices: Define the vertices that represent the region of interest.
* ROI Mask: Create an empty mask with the input image dimensions. Fill the region defined by the polygon vertices with a white color mask.
* Bitwise ADD: This operation retains the pixel values in the ROI and sets the rest to zero.

1. Canny Edge Detection: Detecting edges from the input image

.

* Grayscale Conversion: Convert input image into grayscale as only image intensity is needed for gradients.
* Gaussian Blur: (5x5) Gaussian blur filter is applied for noise reduction.
* Canny Edge Detection: Perform canny edge detection to detect edges.

1. Hough Line Detection:

* Detects the straight lines from the corrected image.
* Draw the detected lines onto the image.

1. Displaying and Saving Output Image

* Display the corrected image with lane detection.
* Save the corrected image with lane detection to a file.

The technique starts with an input image, corrects fisheye distortion, detects edges, applies a region of interest mask that highlights only the necessary portion of the image, uses the Hough transform to detect lane lines, and then displays the result as an image. However, this might necessitate adjusting the parameters for various photos and camera configurations, as it presumes particular camera and distortion properties.

Both approaches serve as a foundation for further development and optimization, but additional advanced techniques and algorithms are typically needed to create a production-level lane detection system that can reliably handle a wide range of real-world situations.

Future systems should be designed to adapt to different types of roads, regions, and traffic conditions, ensuring they remain effective and safe in diverse scenarios. The future of lane

detection holds exciting prospects for the development of more advanced and robust systems. Collaboration among researchers and open-source development will also be key to accelerating progress in lane detection, making advanced solutions accessible and contributing to the field's rapid advancement.

1. **EXPERIMENTAL SETUP**

Images have been obtained from various research papers and studies that have been conducted on lane detection and fisheye cameras.

1. **Approach I:**

**Input 1**  **Input 2**

|  |  |  |
| --- | --- | --- |
| **Input**  **Image** | Input Image 1 | Input Image 2 |
| **Gaussian**  **Blur** | (5X5) Gaussian Blur applied | (5X5) Gaussian Blur applied |
| **Cropped**  **Image** |  |  |
| **Binary**  **Thresholding** | Binary thresholding was applied to the Gray image    Binary thresholding applied on RGB image | Binary thresholding was applied to the Gray image    Binary thresholding applied on RGB image |
| **Color**  **Segmentation** | Color Segmentation applied on the Gray image    Color Segmentation applied on RGB image | Color Segmentation applied on the Gray image    Color Segmentation applied on RGB image |
| **Color**  **Masks Detected** | Color Masks obtained on Gray image    Color Masks obtained on RGB image | Color Masks obtained on Gray image    Color Masks obtained on RGB image |

1. **Approach II:**

**Front fisheye camera Side fisheye camera**

|  |  |  |
| --- | --- | --- |
| **Input**  **Image**  **With**  **Distortions** | Front View of a fisheye camera | Side View of a fisheye camera |
| **Corrected**  **Image**  **Without**  **Distortion** | Rectilinear Projection of Front view | Rectilinear Projection of Side view |
| **Canny**  **Edge**  **Detection** | Edges detected in Front view | Edges detected in the Side view |
| **Region of**  **Interest**  **Masking** |  |  |
| **Lanes**  **Detected** | Lanes Detected in Front View | Lanes detected in the Side view |

1. **RESULTS AND DISCUSSION**

The first strategy concentrates on fundamental image processing methods like color segmentation, basic binary thresholding, and Gaussian blur. Its robustness and accuracy for lane detection in real-world situations are restricted. For lane detection, the project makes certain assumptions about color and lighting, which might not work in every circumstance. This method does not address perspective correction or lane tracking across multiple frames.

The second method, on the other hand, deals with fisheye distortion correction, which is necessary for precise lane detection in pictures captured by fisheye cameras. It uses the Hough Line Transform for lane detection and the Canny Edge Detector for edge detection. To focus on the pertinent area of the image for lane detection, it uses the region of interest (ROI). Even though it handles lane detection and distortion correction, the code might need its parameters adjusted for particular camera configurations and image qualities. To improve accuracy and robustness, real-world lane detection usually uses more sophisticated methods, frequently combining deep learning and computer vision techniques.

1. **CONCLUSIONS & FUTURE WORK**

In summary, we have presented two strategies, for detecting lanes using fisheye data to overcome the challenges posed by distortion in imagery. The first approach, although simple highlights the limitations of relying on image processing techniques and emphasizes the need for more advanced methodologies. On the other hand, the second approach incorporates fisheye distortion correction and advanced computer vision methods showing promise in improving accuracy. May require fine-tuning for specific camera setups.

The lane detection systems outlined in this project serve as a foundation, for lane identification and tracking. While the current implementation meets expectations there are areas where future work and enhancements can be explored. The following features in conjunction with incorporating algorithms have the potential to contribute to the advancement of a sophisticated and adaptable lane detection system:

1. Dynamic Region of Interest Selection: Implement an algorithm that dynamically adjusts the region of interest (ROI) based on road geometry and vehicle dynamics. This would enhance the system’s ability to adapt to varying road conditions and lane configurations.
2. Lane Departure Warning System: Integrate a warning system that alerts drivers when their vehicle deviates from the detected lane. This can be achieved by analyzing lane positions and comparing them to the vehicle’s trajectory.
3. Real-time Camera Calibration: Developing a mechanism for real-time calibration of cameras to account for changes, in their parameters. This can improve distortion correction accuracy and overall system performance.
4. Using Machine Learning for Lane Detection: Incorporation of machine learning methods like networks (CNNs) to detect lanes. Enhancing the model through training, on datasets could enhance its capability to handle situations and difficult lighting conditions.
5. Applying Semantic Segmentation for Lane Recognition: Explore the utilization of semantic segmentation algorithms to classify pixels that belong to lane markings. This approach aims to enhance the understanding of the road environment and increase the system’s ability to handle situations.
6. Enhancing Lane Tracking and Prediction: Develop tracking and prediction algorithms to estimate lane positions accurately. This would be particularly useful, in scenarios involving obstructions or sudden changes in lane configuration.
7. Experimenting with Adaptive Thresholding Techniques: Thresholding techniques that can improve the detection of lane markings under varying lighting conditions and on different road surfaces. The system should be able to adjust these thresholds based on factors.

1. Integrating Multiple Cameras for a Comprehensive View: Expanding the capabilities of the system by incorporating cameras enabling a view of the surroundings. By fusing information from different perspectives, we can achieve a precise representation of the road leading to improved reliability.
2. Lane Type Classification: Functionality that enables classification of types of lanes such as solid lines dashed lines or double lanes. This information is crucial for making higher-level decisions, in driving scenarios.
3. Performance Optimization: Optimization to enhance the real-time performance of the system. Utilizing hardware acceleration and parallel processing to achieve lane detection and tracking.

Additionally, creating a dataset specifically designed for fisheye lane detection can facilitate benchmarking and comparative evaluations of methodologies. By addressing these aspects, we can advance the development of accurate lane detection systems. The future of lane detection holds exciting prospects for the development of more advanced and robust systems

**REFERENCES**

[1] Shigang Li, and Yuta Shinmomura, "Lane Marking Detection by Side Fisheye Camera", IEEE/RSJ International Conference on Intelligent Robots and Systems Acropolis Convention Center Nice, France, Sept, 22-26, 2008

[2] Alex Valente, Stefano Ghidoni, Francesca Ghidini, and Pietro Cerri, "Lane Detection System for Intelligent Vehicles using Lateral Fisheye Camera", Department of Computer Engineering, University of Padova, Italy, Dec 09, 2018-2019

[3] Salma Moujtahid, Rachid Benmokhtar, Amaury Breheret, and Saif-Eddine Boukhdhir, "Spatial-UNet: Deep Learning-Based Lane Detection Using Fisheye Cameras for Autonomous

[4] Tobias Scheck, Adarsh Mallandur, Christian Wiede, and Gangolf Hirtz, " Where to drive? Free Space Detection with one Fisheye Camera", Chemnitz University of Technology, Nov 11, 2020

[5] <https://learnopencv.com/getting-started-with-opencv/>

[6] Yu, B. and Jain, A. K., “Lane boundary detection using a multiresolution hough transform,” in [Proceedings of International Conference on Image Processing], 2, 748–751, IEEE (1997).

[7] Pan, X., Shi, J., Luo, P., Wang, X., Tang, X.: Spatial as deep: spatial CNN for traffic

scene understanding. Computing Research Repository (CoRR) (2017)

[8] [Shigang Li](https://ieeexplore.ieee.org/author/37279611900), [Ying Hai](https://ieeexplore.ieee.org/author/37542161000), "Easy Calibration of a Blind-Spot-Free Fisheye Camera System Using a Scene of a Parking Space", [IEEE Transactions on Intelligent Transportation Systems](https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=6979), Nov 11, 2010

[9] K. Kluge, “Extracting road curvature and orientation from image edge points without perceptual grouping into features”, Proc. Of IEEE Symposium on Intelligent Vehicle, pp.109-114, 1994