2024 Tenth Indian Control Conference (ICC)
9 - 11 December, 2024
IISER Bhopal, India

Performance Comparison of Edge Detection Filters for Lane Detection in Scaled-down AVs

Aayush Ranawat, Vasumathi R, Subhadeep Kumar
Indian Institute of Technology, Madras
ee24e008@smail.iitm.ac.in, ee21d063@smail.iitm.ac.in, subhadeep.kumar@alumni.iitm.ac.in

***Abstract*—Autonomous lane detection is essential for self-driving vehicles, enabling accurate road boundary identification under various conditions. In scaled-down autonomous platforms, real-time lane detection introduces challenges due to limited computational resources and the constraints of edge computing. This study evaluates the performance of three established edge detection algorithms—Sobel, Canny, and Gabor filters—for lane boundary detection on a scaled-down electric vehicle platform. A custom indoor track with colour-coded lanes was developed to improve lane visibility and segmentation, providing a controlled environment for performance comparison. Using the DEFT electric vehicle platform, we applied these edge detection methods to images captured by a front-mounted ZED Mini stereo camera processed on a Jetson Nano. The images undergo colour filtering and masking to isolate relevant lane regions, with affine transformations enhancing lane geometry and accurately extracting inner lane points. Key performance metrics, including edge clarity, processing speed, and detection accuracy, were measured at different vehicle speeds to identify the optimal method for lane detection on resource-limited edge devices. The results reveal distinct strengths and weaknesses of each algorithm, offering insights into their suitability for real-time lane detection in edge-computing applications. Our study recommends an effective approach to lane detection in autonomous vehicles, supporting the development of reliable, resource-efficient navigation systems using classical computer vision methods.**

***Index Terms*—Lane detection, Autonomous Vehicles, Scaled-down Electric Vehicle**

I. Introduction

Autonomous vehicles (AVs) represent a sophisticated application of automated control systems, requiring precise and dynamic interaction between sensing, decision-making, and actuation processes.This is achieved through a carefully coordinated control architecture that balances perception, planning, and control actions in real time. The Perception layer plays an important part in the control of an autonomous vehicle. It is responsible for extracting information from the vehicle's local environment through onboard sensors like LiDAR, radar and cameras. various types of camera systems such as monocular cameras, stereo cameras, and 3D structured light cameras assist in providing visual perception to the vehicle. Visual images are extremely informative compared with other sensor outputs; a lot of information can be extracted, especially from colour images. They contain not only the distance information of the object but also the texture and depth information. Lane boundaries are marked by sharp contrasts in pixel intensity, making edge detection an essential method for extracting lane information from camera images. This study focuses on comparing the performance of three edge detection filters — Sobel, Canny, and Gabor- on a scaled-down electric vehicle platform named DEFT, which is designed to test autonomous algorithms in controlled environments. The Sobel filter, a gradient-based filtering method, is often used for its simplicity and ability to detect well-defined edges in environments with clear lane markings. The Canny edge detector, with its multi-stage process, offers refined edge detection by minimizing noise and improving precision as it incorporates both Sobel and Laplacian operators to find suitable edges in image. Gabor filter, which captures both frequency and orientation information, excels in detecting lane edges even when markings are worn or textured. In addition to these three methods, other techniques like the Hough Transform and Prewitt filters are also widely used for lane detection. However, we chose Sobel, Canny, and Gabor to explore how different edge detection approaches perform in real-time on a scaled-down vehicle as sobel is one of the best gradient-based operators among Robert and Prewitt operators, while canny edge detector is a multistage algorithm which incorporates sobel and laplacian operators to better define the edge boundaries, Gabor filter is a band pass filter used for edge detection, texture classification it is a very computational intensive algorithms which can extract low-level features from an image. We compare the computational performance of these filters by implementing them on DEFT and assessing the real-time capabilities and efficiency of each technique. Our aim is to provide insights into how these filters impact lane detection accuracy and overall performance on scaled-down autonomous electric vehicles.

II. system design

The one-tenth-scale electric vehicle utilized in our experiments, DEFT, has been meticulously engineered to prioritize measuring system states, achieving precise control, alongside other crucial aspects like signalling, perception capabilities,and robust communication Figure 1 shows the block diagram of the one-tenth-scale EV. The dimensions of the scaled-down Electric Vehicle, along with the sensors mounted on the vehicle, are as follows: length = 0.40 meters, width = 0.22 meters and height = 0.20 metres. It comprises the following systems: (i) powertrain, (ii) wheels and axles, (iii) steering, (iv) chassis and suspensions, (v) electronic control units (ECUs), (vi) sensors, (vii) microcontroller, and (viii) single board computer (SBC)

979-8-3315-1721-2/24/$31.00 ©2024 IEEE

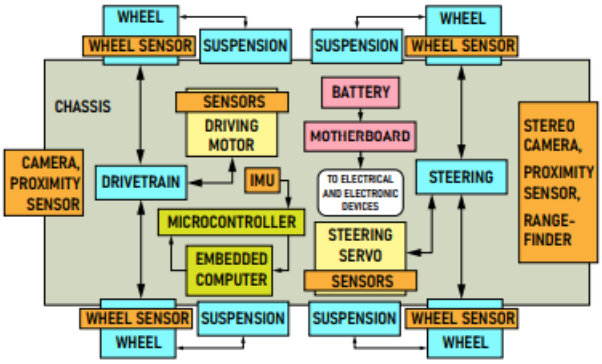


Fig. 1. Block Diagram of DEFT System [1]

DEFT is meticulously designed to prioritize measuring system states, achieving precise control alongside other crucial aspects such as signaling, perception capabilities, and robust communication. This newly designed DEFT encompasses the following key features and capabilities:

noitemsep

- **Physical Structure:** Robust aluminum chassis, carbon fiber base, rigid mounting structures, and links.
- **Actuation:** Precision actuators with smooth control.
- **Sensing Capabilities:** Sensors for measuring the steering angle, acceleration, orientation, angular velocities of wheels, and battery cell voltages. Perception sensors for detecting objects in the vehicle's proximity and visualizing the environment in the front and rear of the vehicle.
- **Control:** Embedded systems capable of computing control inputs onboard utilizing sensor data, with the flexibility to upgrade the platforms as needed.
- **Communication:** Equipped with Bluetooth and WiFi for V2V and V2X communication. Features a real-time clock for time synchronization and time-triggered operations.



Fig. 2. side view of deft with zed camera

III. Experimental Setup

Our experimental setup consists of a custom lane map created using coloured tape in a room with dimensions of 10.45 meters in length and 6.95 meters in breadth. The inner lanes are delineated with green tape, while the outer lanes are marked with blue tape, effectively defining a clear navigation area for our autonomous vehicle. These distinct coloured tapes facilitate the accurate filtering of lane points, which are captured by the ZED camera mounted on the vehicle, ensuring precise lane detection and navigation.

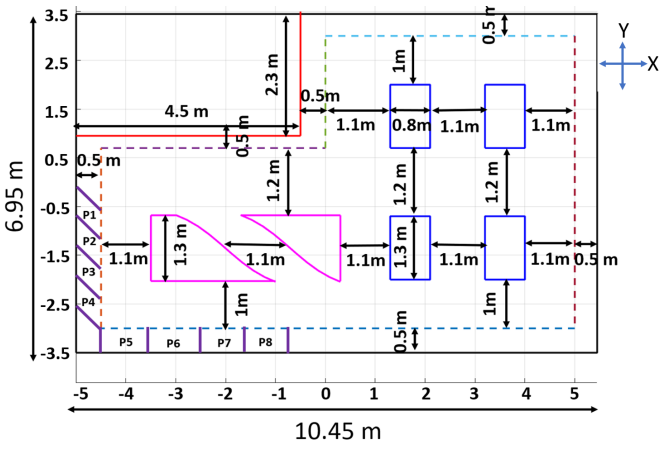


Fig. 3. lane track map

IV. Methodology

A. Color Filtering

Colour segmentation in OpenCV often uses the HSV colour space instead of RGB, as HSV provides better colour separation. The HSV space represents colours in a cylindrical format: **Hue** (0-179),0° is Red, 120° is Green, and 240° is Blue, **Saturation** (0-255) measures intensity from grey (0) to full colour(100%), and **Value** (0-255) represents the brightness of colour. To segment a colour, you set threshold values and create a mask to isolate the desired colour.figure 5 shows simple colour filter applied over RGB image.

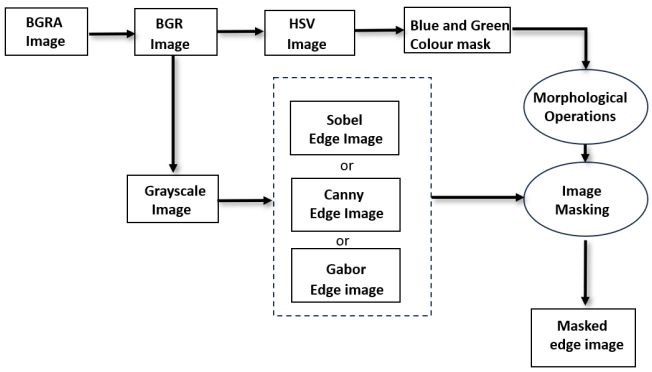


Fig. 4. image processing flowchart

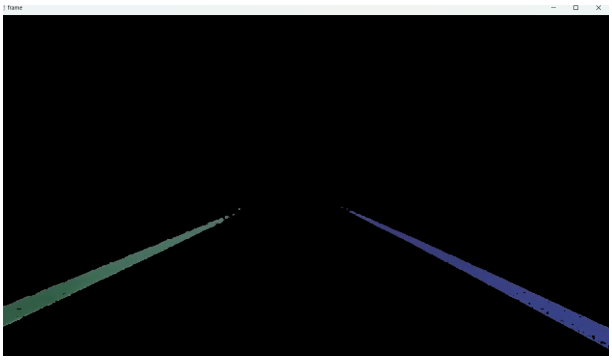


Fig. 5. Masked RGB

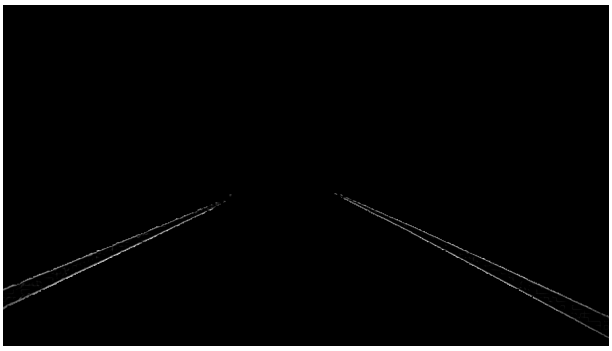


Fig. 6. Masked Sobel without morphological operations

B. Morphological and Affine Transformations

Morphological operations modify shapes and structures in binary or grayscale images. Key operations include:

- **Erosion**: Shrinks foreground objects by removing boundary pixels.
- **Dilation**: Expands foreground objects by adding pixels.
- **Opening**: Removes small objects (erosion followed by dilation).
- **Closing**: Fills small holes in objects (dilation followed by erosion).

opening helps in reducing noise in our colour mask, to detect both lane edges properly from the edge map, as shown in flowchart we apply ,fig 7 shows dialtion applied to masked RGB morphological closing on our colour mask before superimposing mask on edge map of image.

Affine transformation is a type of geometric transformation that preserves collinearity and the ratios of distances between points along a line. There are several types of affine transformations, including:

In our context of image processing, we can apply affine transformations to manipulate colour masks. For example, consider a blue mask in an edge map. By shifting this mask slightly to the right, we can isolate the innermost lane points of the right lane. Similarly, we can perform the same transformation on the left lane mask or isolate the outermost lane points from the edge map of the image.

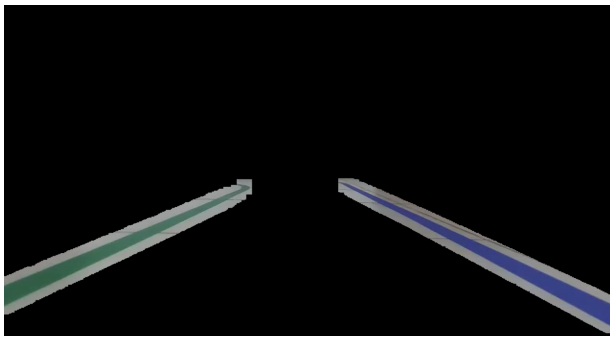


Fig. 7. Masked RGB with morphological operations

C. Edge detection

An edge detection algorithm refers to a technique used for image analysis to identify the locations of significant edges in an image while filtering out false edges caused by noise. It acts as a high-pass filter to measure the rate of change at each pixel and then thresholding the filter output to determine the pixels that represent edges.

1) ***Sobel Edge Detection****:* The Sobel filter uses two kernels/filters: one for detecting horizontal changes in direction and the other for detecting vertical changes in direction

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \text{ and } G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}$$

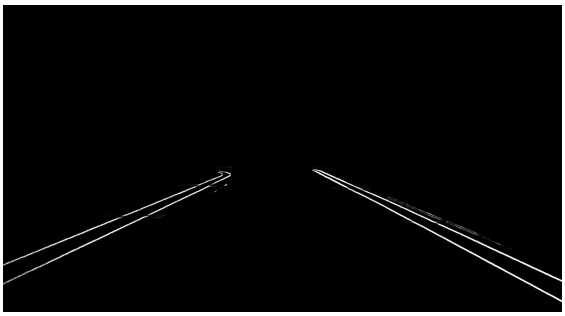


Fig. 8. sobel edge detection results

It works as a first-order derivate and calculates the difference in pixel intensities in an edge region,the kernel calculates the difference between right and left pixel values around that edge. Also, the centre values of both the first and third columns are 2 and -2, respectively. This gives more weightage to the pixel values around the edge region. This increases the edge intensity, and it becomes enhanced comparatively to the original image. [2].

2) ***Canny Edge Detector****:* The Canny edge detector combines gradient and Laplacian techniques, summarized in these steps:

1) Smooth the image with a Gaussian filter.
2) Compute gradients using the Sobel operator.
3) Determine gradient magnitude and orientation.
4) Apply a 1D Laplacian along the gradient direction.

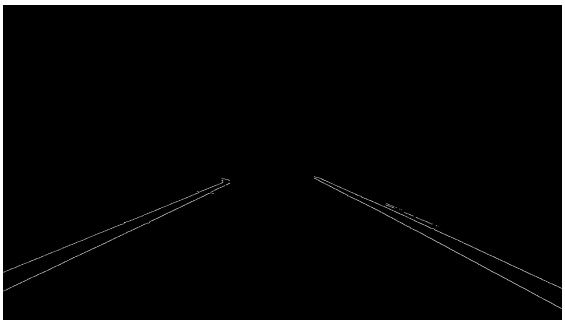


Fig. 9. Canny edge detection applied to a lane frame

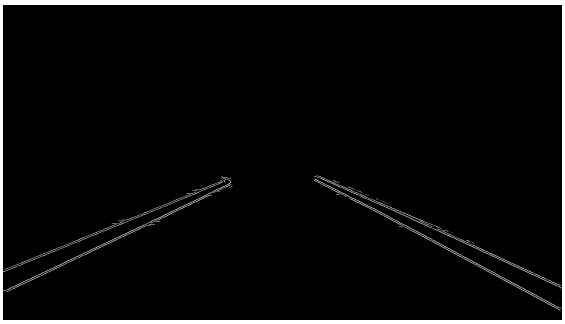


Fig. 10. Gabor filter for various values of λ and σ

Edges are detected at zero-crossings of the Laplacian. [3] Hysteresis thresholding is used to classify edges:

- Edges with values $> T_{\text{upper}}$ are sure edges.
- Edges with values $< T_{\text{upper}}$ but connected to sure edges are considered edges.
- Edges with values $< T_{\text{lower}}$ are discarded.

The Canny detector in OpenCV is applied with cv2.Canny(), where hysteresis thresholds are set with parameters T_{lower} and T_{upper}.Fig9 shows masked canny detection output image.

*3) **Gabor Filter**:* The Gabor filter is a linear filter used for edge detection and texture analysis. It combines a Gaussian function with a sinusoidal wave to form a bandpass filter that highlights specific frequencies and orientations in an image. When applying this to an image, we expose several favourable properties:

- It emphasizes edge pixels.
- Pixels immediately adjacent become suppressed.
- It is directional
- Pixels farther away have less influence (Local area)
- Since the filter exhibits both a sine wave and bell curve nature, parameters common to both affect our filter:

$$g(X) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \exp\left(i\left(2\pi\frac{x'}{\lambda} + \psi\right)\right) \quad (1)$$
$$X = (x, y, \lambda, \theta, \psi, \sigma, \gamma)$$

Key parameters:

- **Sigma (σ)**: Controls the width of the Gaussian envelope, affecting the filter's sensitivity.
- **Theta (θ)**: Specifies the orientation of the sinusoidal wave.
- **Lambda (λ)**: Determines the wavelength of the sine wave, influencing the stripe width.
- **Gamma (γ)**: Adjusts the ellipticity of the Gaussian envelope.
- **Phi (ψ)**: Phase offset of the sine wave.

V. Results and Conclusion

To assess the performance of various image filters, we conduct vehicle trials around the track at varying speeds for each lap, recording both the filtered output image and the filter's frame rate. Table 1 provides a summary of the frame interval time(Td) for the Sobel and Canny filters across different speeds, enabling a comparative evaluation of filter efficiency and responsiveness under dynamic conditions. As we increased the angular velocity from 80rpm to 300rpm, the output of the Sobel filter was observed to be less noisy. Also, the average time period between successive iterations for Sobel is almost similar to Canny for every angular velocity,and both are sufficient for real-time applications. However, the Gabor filter, while delivering significantly cleaner and sharper edge detection, caused a noticeable drop in the frame rate, making it less suitable for autonomous operations and real-time processing.

Speed (RPM)	Sobel Filter (Td in ms)	Canny Filter (Td in ms)
80	150–170,	120–150
140	150–180,	130–150,
170	150–180,	130–150,
240	150–170,	130–150,
300	150–160,	130–140,

TABLE I
Comparison of Td (ms) frame interval between Sobel and Canny Filters at Different Motor Speeds

Considering both speed and accuracy, the Sobel filter appears to be the most appropriate choice for our experimental setup. It strikes a balance between performance and stability, providing reliable lane detection at a frame rate that is high enough to meet real-time requirements. Canny edge detection, while usable, tends to produce less stable results compared to Sobel in this context. Therefore, Sobel is preferred for its consistent and faster output.**YouTube link of the video: https://youtu.be/ug9vr7d9Fro**

References

[1] S. Kumar, R. Pasumarthy, and N. P. Bhatt, "Deft - a scaled electric vehicle platform for testing autonomous driving algorithms," *TechRxiv*, 2024, www.doi.org/10.36227/techrxiv.24580615.v3.

[2] M. H. Hasan and D. Kr Das, "Analysis of edge detection for road lanes through hardware implementation," in *2021 Fourth International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, 2021, pp. 1–6.

[3] M. Philip, M. A. Kurian, R. Joseph, R. Sruthi, and S. Thomas, "A computer vision approach for lane detection and tracking," in *2021 2nd International Conference on Advances in Computing, Communication, Embedded and Secure Systems (ACCESS)*, 2021, pp. 188–192.