Enhanced Lane Detection System with Gaussian and Bilateral Filtering

RATNAYAKE Himath
Computer Science
Griffith University
Gold Coast, Australia
himath.ratnayake@griffithuni.edu.au

***Abstract*— Lane detection is a mechanism that is currently used in higher end cars as a technique that assists drivers in identifying lane markings. There are a variety of practical applications in which lane detection proves useful. This ranges from lane departure warning systems, which aim to assist drivers in keeping center within their lane, or to warn drivers when they veer off the lane in a dangerous way. Cars that employ this system have shown to visibly improve driver safety by reducing driver error resulting from distractions or drowsiness. More recently, lane detection systems have earned a place as an integral subsystem of self-driving cars. However, there are still several challenges posed by current lane detection system. Common driving situations in which issues can still occur includes night-time driving when visibility of lane markers is unclear, as well as when other road markings such as brake marks or shadows hinder the vision of the input camera. Moreover, one of the biggest challenges effective lane detection systems have is the need to overcome noise appearing in the input image or video frames. For this, Gaussian and Bilateral filtering image pre-processing methods were compared to see how they fared under different weather conditions, ranging from clear days to more challenging scenarios such as low-light or cloudy days.**

Keywords—lane detection, Canny edge detection, Hough line transform, Gaussian filter, Bilateral filter

I. Introduction

Lane detection research has been a topic of great discussion in recent years, and is still a technology which has much room for improvement. Currently, there are a number of key factors that hinder lane detection methods from reaching their full potential. One such factor is the wide variety of environments that lane detection systems must be robust to handling. This includes but is not limited to sunny, cloudy, and clear weather conditions as well as more challenging environments that pose significant noise such as rain or night-time driving. The other major factor that lane detection systems must be wary of is the computation time necessary to identify lanes in real time. For this, efficiency and speed in code computations is also paramount.

II. Previous Work

Although lane detection systems have been a field of study that has only recently garnered great interest, there have been developments as recently as the last year where different models have been implemented. Currently, lane detection systems fall under two major umbrellas – they are either feature based or model based. In model-based systems, mathematical methods such as neural networks and deep learning are used to recognise lane structure.[1][15] It has been found that although such systems are already very robust against noise, they remain difficult to carry out in real time lane detection due to the computational power necessary, and the limited processing power that is available within most consumer-grade cars to produce a result in a timely manner.

Therefore, a feature-based method has been used, which has been proven to take less computational time per frame [1]. Within feature-based systems, image pre-processing is already standard, but the extent of pre-processing differs greatly between different systems. For example, initial detection systems had very little pre-processing outside of converting an image to grayscale and were susceptible to failing due to factors such as foreground noise or lack of detectable edges [3]. In more recent implementations, image filtering techniques such as the Gaussian filter and Bilateral filter were explored in edge detection [4]. Within this report, the Gaussian and Bilateral filter's efficacy will be evaluated on a number of metrics, including accurate detection rate, robustness under different scenarios, and computation time needed to produce a result. Other pre-processing techniques will also be utilised. This includes reducing the image to a select region of interest to remove spurious edges.

III. TECHNICAL APPROACH

1. Grayscale Conversion

To tackle these issues, several image pre-processing tasks will be carried out beforehand. First, an input image is read in and converted to grayscale. By doing this, each pixel value is assigned an intensity value between 0 and 255 without the information of colour [5]. This reduces the image complexity and decreases computation time while still being able to highlight key image features such as shape, contrast, perspective and edges. This is because without the red, green, and blue channels of colour, the new grayscale image only carries the intensity of the pixel in shades from black to white. This conversion is defined by [6]:

$$gray(x,y) = 0.2989 * f(x,y,R) + 0.5870 * f(x,y,G) + 0.1140 * f(x,y,B) \quad (2)$$

Definition	:	
gray(x,y)	:	output image as grayscale image
f(x,y,R)	:	first color channel, red channel pixel value in specific (x,y) coordinates
f(x,y,G)	:	second color channel, green channel pixel value in specific (x,y) coordinates
f(x,y,B)	:	third color channel, blue channel pixel value in specific (x,y) coordinates

After this, the image is passed through either a Gaussian or Bilateral filter. This step is necessary to smooth out the noise in the image.

2a. Image Smoothing: Gaussian Filter

The Gaussian filter works by convolving the grayscale image with a Gaussian function. The transformation for each pixel is given by the equation:

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$

In this equation, σ denotes the standard deviation. This influences how much the center pixel's neighbours affect the center pixel's computation result [7].

However, the key limitation is that Gaussian filters will also smooth the edges of an image, increasing the possibility of missing weak edges [8]. This means that less of the weak edges are likely to be detected. In the context of lane detection, this means that in low light environments isolated edges are likely to appear in images where parts of the lane markings are low visibility and too weak to be detected.

2b. Image Smoothing: Bilateral Filter

The other filter that is explored is a Bilateral filter. In Bilateral filtering the similarity between the central pixel where the filter is applied and a pixel in its neighbourhood used for blurring is computed. It is essentially defined as a weighted average of nearby pixels, that takes differences in neighbourhood values into account so that edges can be better preserved when smoothing[8].

Essentially, for a pixel's value to influence another pixel, it needs to both occupy a location nearby and have a similar value. In doing so, Bilateral filtering has the effect that if the two pixels compared are very similar, the smoothing will happen as normal. However, if the pixel values are very different (which is a property of an edge), the filtering will not be done, thus skipping over the smoothing for the given pixel [9].

Bilateral filtering is defined as:

$$I^{\text{filtered}}(x) = \frac{1}{W_p} \sum_{x_i \in \Omega} I(x_i) f_r(\|I(x_i) - I(x)\|) g_s(\|x_i - x\|),$$

$$W_p = \sum_{x_i \in \Omega} f_r(\|I(x_i) - I(x)\|) g_s(\|x_i - x\|)$$

Where:

I^{filtered} is the filtered image;

I is the original input image to be filtered;

x are the coordinates of the current pixel to be filtered;

Ω is the window centered in x, so $x_i \in \Omega$ is another pixel;

f_r is the range kernel for smoothing differences in intensities (this function can be a Gaussian function);

g_s is the spatial (or domain) kernel for smoothing differences in coordinates (this function can be a Gaussian function).

The edge preserving nature gives the greatest advantage in conditions like environments where there is low visibility. Here, the amount of weak edges is generally greater. As can be seen by figure 1, the Bilateral filter was able to detect a larger portion of the left lane line, as opposed to the Gaussian one which was weaker when it came to the finer edge details.

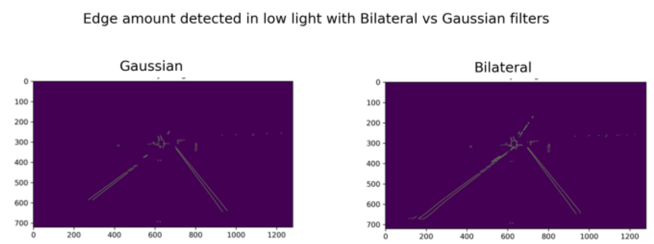

Figure 1: Edge detection in low visibility with Gaussian vs. Bilateral filtering as the image pre-processing technique

The caveat to this though is that the Bilateral filtering procedure is more computationally expensive than the Gaussian filtering method [10]. This computational difference is explored further in the experiments section.

3. *Cropping Region of Interest*

This new grayscale image, smoothed by one of the above filters was cropped such that only a specific region of interest surrounding the immediate lane markers in front is present. This region of interest was predefined as a set of image coordinates that the lane markers were most likely to fall between. In this way, unwanted foreground features that may be picked up can be reduced, resulting in a more accurate alignment of the lane markings.

The shape of the region of interest is that of a triangle with vertices at:

- (0, height of image)
- (width/2, height/2)
- (width, height)

Each image is unique in that the ideal region of interest can vary, but with the above edge points it was possible to get a region of interest that not only removes a majority of noise, but also was able to capture most of the relevant areas of road no matter the image size read in. An example of this is shown in figure 2 below, where it was able to focus on the lane markers while ignoring other foreground regions like the road signs, other lanes, and trees.

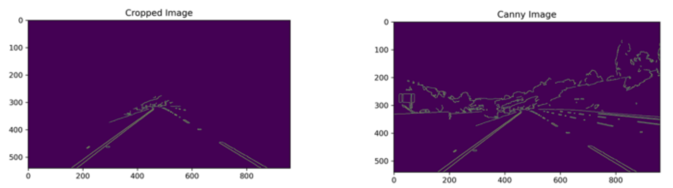

Figure 2: How cropping to a region of interest eliminates a significant amount of foreground noise

Common foreground obstructions encountered when testing this lane detection program that the region of interest selection was able to filter out included:

- Cars in the distance
- Incoming car headlights
- Traffic cones
- Traffic lights
- Pedestrians
- Trees and bushes
- Road signs
- Lane markings adjacent to the one the car was currently on

4. *Canny Edge Detection*

An edge is defined as sharp variation in intensity within an image. The Canny edge detector was the edge detection technique of choice in this investigation and was implemented using OpenCV's Canny() function, which does the following steps. After the image smoothing occurs, a Sobel kernel is used to filter the image both the horizontal and vertical direction. In doing so, the first derivation in the horizontal direction and the vertical direction can be computed [11]. Together, this can be used to find the edge gradient by:

$$\mathbf{G} = \sqrt{\mathbf{G}_x{}^2 + \mathbf{G}_y{}^2}$$

The direction of the edge can also be found by:

$$\theta = \tan^{-1}\left(\frac{G_y}{G_x}\right)$$

Then, non-maximum suppression is used to select the pixels with the highest intensity values in an edge, and setting the other neighbouring pixels values to zero. This will allow the final image to have finer edges. Hysteresis thresholding is then use to differentiate weak and strong edges. Weak edges connected to strong edges are considered as real edges, whereas weak edges not connected to any other strong edges are removed. In doing so, the final image has much clearer defined edges.

5. *Hough Line Transformation*

Hough line transform is then used to extract the lines in an image, using opencv's HoughLinesP() function. A single line can be expressed with the equation:

$$y = mx + c$$

Here, m is the slope and c is the gradient. However, there are limitations to this equation due to its inability to form vertical lines, resulting in incorrect values for the gradient m. Therefore, lines can instead be mapped to Hough space as seen in figure 3 [12].

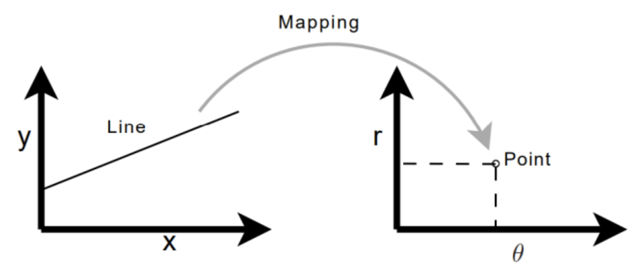

Figure 3: Conversion of a given line to a point in Hough Space

Here, θ becomes the angle the line has with the x axis and r is the distance from the line to the origin.

$$r = x \cdot \cos\theta + y \cdot \sin\theta \Leftrightarrow$$

An edge map from Canny is given, and edges are located placed as straight lines. The key idea is that a point is mapped to all lines that can pass through that point. Even though an edge map can have many points, the goal with Hough transform is that each edge point is transformed to a line in Hough space and the places where most Hough space lines intersect is determined to be a "true" line within the edge map.

6. *Selecting Line Candidates*

With the line candidates found, a special procedure is utilised to find the best two-line candidates for either side of the lane image. First, for each of the Hough lines found, the gradient of the line is calculated [13].

This is done by a simple:

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

Once the gradient for a line is calculated, lines are filtered by their angle of incline. If it is found to be 0 (a flat line) or its absolute value is less than 0.5, the line is discarded. By doing this, lines with a "weak" gradient are selectively removed, as the lane lines in almost all normal cases have an incline stronger than this, closer to a vertical incline rather than a flat one. After this, the lines are further divided into two groups. The "left" group corresponds to the left lane line and this is where lines with a negative gradient (< 0) are placed. Conversely, the "right" group is where candidate lines with a positive (>0) gradient are placed.

This method is then extended even further by calculating the degree of incline for each line by:

$$\theta = \tan^{-1}(m)$$

By doing this, it is possible to filter the lines being chosen even further by a specific angle value of choosing. For this investigation a number of angle values were tested. It was found that filtering such that the angle of the left line was greater than -65 degrees and the angle of the right line was less than 45 degrees yielded the best results. However, it is also important to note that in more extreme perspectives this selection can fail.

Once the ideal filtered candidate lines were found, the final "ideal" line must be found which best represents the candidate of lines found in the left and right subgroups.
This was done by fitting a linear function using the poly1d function in the SciPy library for both the set of left and right lines found earlier.

This results in a final fitted line, consisting of four coordinates: left side end point and start point, and the right line end point and start points. This was then overlaid on the original image.

IV. Experiments

1. *Performance Summary*

For each of the four weather conditions explored (clear, cloudy, rainy, and night-time), 20 images were analysed, sourced from online. For night-time and rainy images, frame samples were also extracted from videos online through a separate python program. To show the reliability of the lane detection program under each condition, a method was needed to quantify the accuracy of detection. For this, each time an image was finished with finding a lane, it was then visually scrutinized to see if the lane lines from the program were overlaid on the lane markings properly. A score was then given based on the extent of detection, to quantify the accuracy of how close the detected image was to the "ground truth", or actual placement of lane lines.
Each image could get a maximum score point of 1 (perfect detection) and a minimum of 0. An image would get a score less than 1 if it failed in one of the following areas:

- Failed to detect one of the lane lines (-0.5 points per line)
- If an overlaid line was visibly misaligned (-0.25 points per line)

With this scoring system in mind, the following results were found and tabulated into table 1:

Weather	Gaussian	Bilateral
Clear	18.5/20	19.5/20
Cloudy	18.5/20	19/20
Rainy	13.5/20	16/20
Night	11/20	16.75/20
Total	**62.5/80**	**69.25/80**

Table 1: Performance Comparisons with Gaussian and Bilateral Filtering

The Bilateral filter was able to perform slightly better than the Gaussian filter when it came to edge detection in clear and cloudy conditions. This similar performance makes sense, as the edges that comprise lane lines in clear and cloudy conditions are generally stronger than those in rainy or night conditions, so the better edge preserving nature of the Bilateral filter matters less. However, this edge preserving nature's effectiveness was shown to matter much more for images in rainy and night conditions. For example, in images where the Gaussian filter wasn't able to detect any lines, the Bilateral still managed to pick up at least one line in most cases, earning it a higher score. This is a strength of using this metric, as it allows for better differentiation between image detection features. However, due to the lower sample size of n=20 for each weather condition, it may be worth testing more images for each condition. However, the data was found to be reliable and backed up by the theory explained within the technical approach.
Specific key example outputs for each of these conditions are discussed in the next section of the report, including the images where the lane detection algorithm was not as effective at detection and why this was the case.

2. Performance in Clear and Cloudy Conditions

First, the algorithm's efficacy was tested in simple common road settings with both clear and cloudy conditions. In a majority of clear and cloudy conditions, the program was able to effectively find the lane markers on both sides of the image up to a reasonable distance, as can be seen by figure 4.

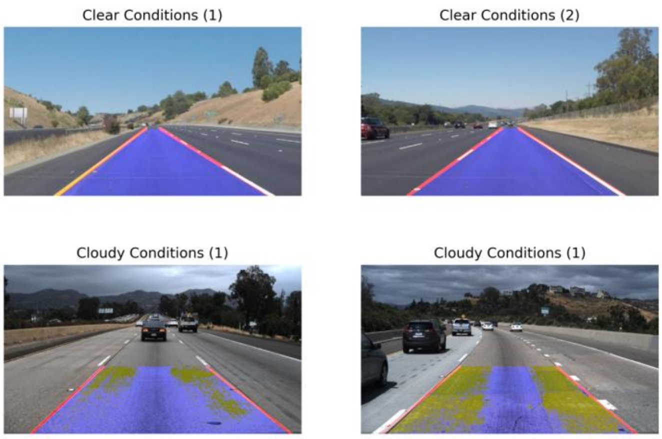

Figure 4: Lane detection in conditions where there is minimal noise

3. Performance Comparison with Degree Filtering

As discussed before, filtering the possible input lines by degree lead to a noticeable improvement in lane detection quality, especially on roads such as in figure 5 below where the surface itself contained a large amount of noise. As can be seen below, when lines were selectively chosen with the technique mentioned previously, the final portion of road detected was noticeably closer to the actual road shape. This is particularly noticeable in the Bilateral filter images with and without the degree filtering.

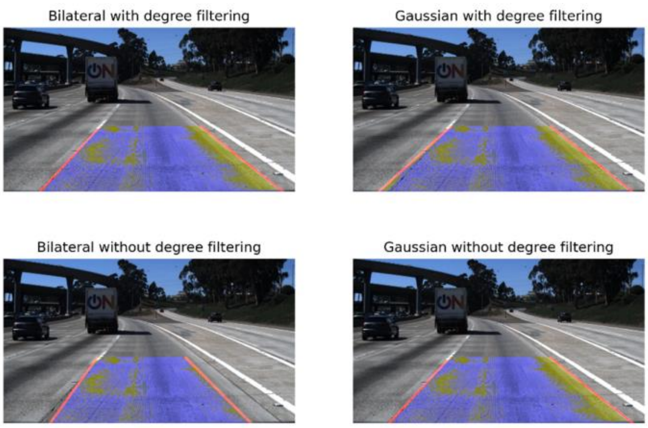

Figure 5: Comparisons of lane detection accuracy with and without filtering the candidate lines by angle

With degree filtering, the left line of the road more closely follows the shape of the left lane marker. Moreover, the right side of the lane is closer to the actual lane line, giving a more accurate representation of the road shape. In this particular instance, Gaussian filtering both with and without degree filtering was able to get a similar result to Bilateral with degree filtering. This can be attributed to the fact that it smoothed over the extra edge lines in the road that Bilateral picked up on; because of Gaussian's uniform smoothing the inner edge of the road (that was picked up as the right-side line of "Bilateral without degree filtering") wasn't detected as the road edge. However, with effective degree filtering, Bilateral was able to close this gap in detection deficiency.

4. Performance with unclear lane markings

The algorithm was then tested in a more difficult environment, in which the road edge markings were not as obvious but the weather itself was still clear. Here, as can be seen in figure 6, the Bilateral filter was very effective at finding the lane markings, both with and without degree filtering. This can be attributed to the fact that were few lines found that fell outside of the designated allowed degrees for each side, so there simply wasn't many spurious lines to filter.

However, the Gaussian filtering failed both with, and without degree filtering. This is attributed to the fact that in filtering, it uniformly smoothed over the right side of the image, which had lane markings which were not as apparent as those on the left side. Because of this, the lines found clustered around the left side, resulting in the incorrect recognition of the lane below.

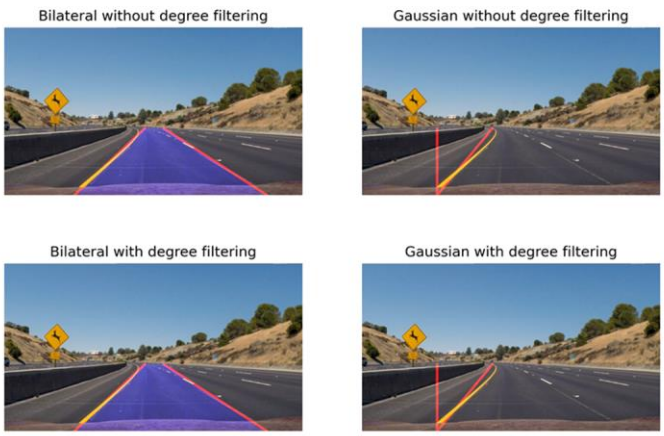

Figure 6: Comparisons of lane detection accuracy with Gaussian and Bilateral filtering when road markings are not as clear

5. *Performance in Rain*

Then, tests were conducted for a more challenging but still very common road condition: rain. With rain, a number of challenges must be overcome.

In table 2, a few of these key challenges are detailed.

Challenge	Explanation
Droplets on windscreen	In many cars, the placement of the camera is behind the windscreen. When rain falls on the windscreen, this then results in a greatly increased noise from the water droplets. This also comes with other consequences such as the presence of other sources of noise like windscreen wipers
Increased noise on the road	When rain hits the road, the amount of light reflection within the road itself also increases greatly. This lowers the contrast (intensity of pixel difference) of the lane markers with the road, resulting in edges becoming harder to find.
Overall lower visibility and darker	Usually, with rainy conditions the surroundings become darker. Although it is not quite as dark as driving in night conditions, there is still room for the overexposure of light to occur, further adding to noise within the image.

Table 2: Example challenges faced by lane detection in low light conditions

With these conditions in the mind, the Bilateral and Gaussian filters were tested in a number of images with rainy conditions. In figure 7, the image Rainy Conditions (1) exhibits some of the challenge points talked about in table 2. For example, there are clear droplets on the windscreen and the light reflection on the road is very prominent.

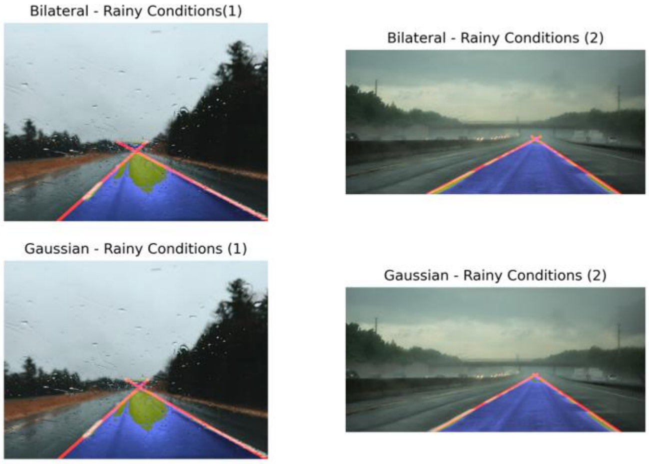

Figure 7: Performance of Gaussian and Bilateral filtering lane detection in rainy weather conditions

Nevertheless, both the Bilateral and Gaussian filters were still able to capture the lane markers effectively. In Rainy conditions (2), the surroundings are darker and there is more noise on the road in the form of tire prints from cars and also light reflection, which could potentially be detected as false positive edges. However, both Bilateral and Gaussian filtering operations were able to find the lane markings effectively, in part due to the assistance of degree filtering.

However, the algorithm still has limitations and showed some struggle in heavy rain conditions; particularly when the amount of rain droplets within the windscreen was very heavy. For example in figure 8, although both Gaussian and Bilateral filtering methods produced a line output, it didn't follow the actual lane markings, which is the ideal intended goal. In this instance where windscreen noise was heavy, the Gaussian filtering operation actually yielded more accurate results than the Bilateral. This is because in this scenario, the uniform filtering of the algorithm actually works in its favour.

As can be seen by figure 8 below, the Bilateral filter was able to find noticeably more "edges" than the Gaussian filter due to the edge preserving nature of the filtering; even smaller raindrops were detected when Canny was performed. However, a majority of these edges were insignificant noise. With Gaussian, only the raindrops with more prevalent edges were detected, resulting in less noise when the time to undertake Canny came, ultimately resulting in a slightly better, but still not ideal result.

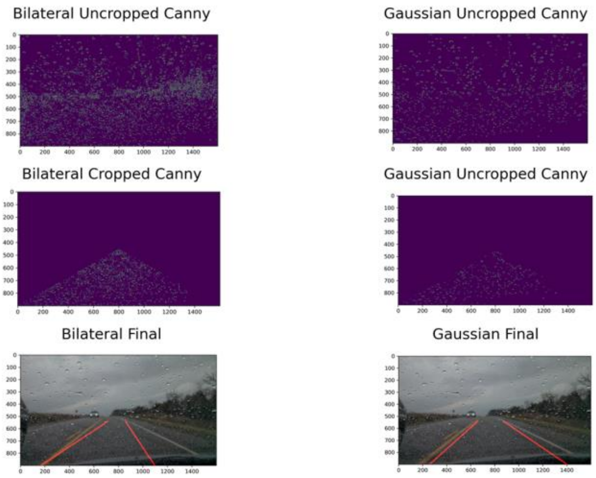

Figure 8: Extent of edge detection with images filtered by Gaussian vs. Bilateral filtering. Gaussian picked up less rain drops, leading to a slightly better result

6. Performance at Night

Finally, the algorithm was tested under night conditions. In lane detection tasks, this poses the most amount of challenge due to the lower visibility and contrast between the lane markers and the road itself. Foreground features such as car headlights are also prone to hindering the efficacy of the lane detection if careful region of interest selection is not undertaken. In all night cases tested, the Gaussian filtering struggled to detect the lane markings effectively as seen by Figure 9 below.

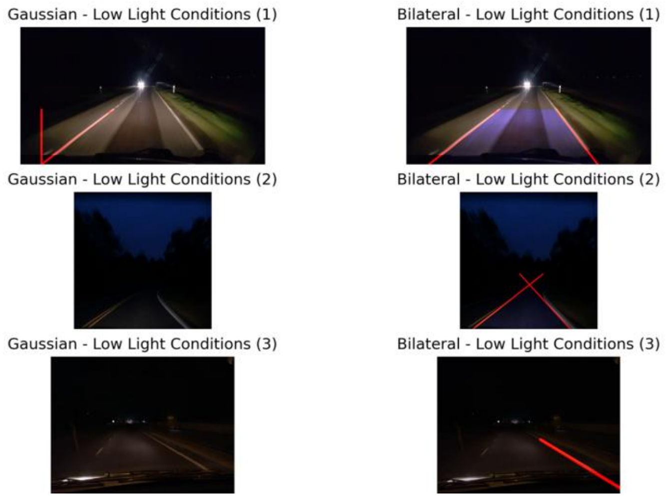

Figure 9: Detection of lane lines in low light conditions with Gaussian vs. Bilateral filtering

Low Light Conditions (1) was the only image in which the Gaussian was able to detect the lane marker, but this was again only on the left side. The right-side marker was surrounded by more noise with a variation in colour going from gray, to the white lane marker, to a dark gray colour making the edge detection much harder when uniform smoothing was done.
However, the Bilateral filter proved to be very effective due to its edge preserving nature, being able to effectively find the shape of the road in Low Light Conditions (1). It also ignored other noise in the image such as the incoming car headlights. Low Light Conditions (2) presented a much darker and more shadowy image, where one of the lane markers was not completely straight. Nevertheless, with Bilateral filtering, the program was able to find both lane markers unlike the Gaussian filtering, but it also overshot the length of said lane markers. Finally, in the Low Light Conditions (3), only the left lane marker could be found with Bilateral filtering. This is most likely because the left edge marker was not a solid line, but rather a disjointed one and the contrast between this marker and the road was particularly low.

7. Computation Time Comparisons

Finally, for finding the average computation time, a set of 40 image frames with varying weather conditions and noise levels were passed through under 4 varying parameters. This included 10 clear images, 10 cloudy images, 10 rainy images and 10 low-light images. The parameters explored were computation time with Gaussian filtering, Bilateral filtering, and each of those filters with and without the line filtering technique explored previously. The time taken for each image to be processed was stored in a list as the frame passed through, and the average time in seconds for the given parameter was then calculated from the list values. Each time an image was processed, it was also shown on screen and whenever an incorrect detection was found, it was manually noted. For this test, an incorrect image was defined by the stricter constraint that all lines in the image had to be overlaid correctly on the lane lines to be classed as a correctly detected image. These results are summarised in table 3 below:

	Gaussian (No DS)	**Gaussian (DS)**	**Bilateral (No DS)**	**Bilateral (DS)**
Avg. Time Per Img	0.031	0.031	0.039	0.041
Acc. (%)	78%	88%	85%	93%

Table 3: Processing time comparisons with Gaussian vs. Bilateral filtering

It was found that the usage of selection had no noticeable difference in computation time for Gaussian filtering, with the average time taken being the exact same both with and without selection. For Bilateral filtering, the time with degree selection was found to be slightly slower but this could also be attributed to random fluctuations with computation speed. It was found that Bilateral filtering with degree selection was the most accurate, achieving a 93% accuracy while the Gaussian without degree selection was least accurate at 78%, being heavily weighed down by its lower detection rate at night. It was found that selectively choosing the candidate lines by gradient incline and filtering out flatter lines (procedure discussed in the technical approach) led to a noticeable increase in accuracy and negligible increase in overall processing time. The computation time for Bilateral was on average 9 milliseconds slower than that of Gaussian, which was expected as discussed in the technical approach.

V. CONCLUSIONS

In this investigation, the effectiveness of lane detection using Canny edge detection and Hough Transform under a variety of common road conditions such as clear, cloudy, rainy and night-time was explored. The use of a variety of image processing techniques and their effectiveness was discussed, particularly regarding the pros and cons of using a Gaussian filter to smooth an image as opposed to a Bilateral filter. It was found that the Bilateral filter was more effective at finding lane markings in low visibility conditions, as opposed to the Gaussian one which struggled in night-time conditions. This can be attributed to the Bilateral filter's edge preserving nature. However, this edge preserving nature was also found to hinder the Bilateral filter when false positive edges like raindrops on a windscreen were present. The computation time between Gaussian and Bilateral filtering was also compared and it was found that Bilateral filtering on average took slightly longer, as was expected. In further studies, this gap in computation time should be explored in real-life situations to determine whether this slowness would render it less practical for lane detection in a real-time scenario. Ultimately, it was found that Bilateral filtering was more effective in a wider variety of conditions and should be further explored as a candidate for lane detection system pre-processing strategies.

REFERENCES

[1] Tsung-Ying Sun, Shang-Jeng Tsai, and V. Chan, "Hsi color model based lane-marking detection," 2006 IEEE Intelligent Transportation Systems Conference, 2006.

[2] S. Sultana and B. Ahmed, "Robust Nighttime Road Lane Line Detection using Bilateral Filter and SAGC under Challenging Conditions," 2021 IEEE 13th International Conference on Computer Research and Development (ICCRD), 2021.

[3] L. A. F. Fernandes and M. M. Oliveira, "Real-time line detection through an improved Hough transform voting scheme," Pattern Recognition, vol. 41, no. 1, pp. 299–314, 2008.

[4] Y. Li, L. Chen, H. Huang, X. Li, W. Xu, L. Zheng, and J. Huang, "Nighttime lane markings recognition based ON CANNY detection and Hough transform," 2016 IEEE International Conference on Real-time Computing and Robotics (RCAR), 2016.

[5] Dynamsoft, "Image processing 101 chapter 1.3: Color space conversion," Dynamsoft Document Imaging Blog, 16-Sep-2019. [Online]. Available: https://www.dynamsoft.com/blog/insights/image-processing/image-processing-101-color-space-conversion/. [Accessed: 30-Sep-2021].

[6] H. Ham, "Image enhancement using spatial domain filtering," School of Computer Science, 2019. [Online]. Available: https://socs.binus.ac.id/2017/03/20/image-enhancement-using-spatial-domain-filtering/. [Accessed: 30-Sep-2021].

[7] R. Fisher, "Gaussian smoothing," Spatial Filters - Gaussian Smoothing, 2003. [Online]. Available: https://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm. [Accessed: 30-Sep-2021].

[8] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271), 1998.

[9] "Bilateral filtering for gray and color images," Bilateral Filtering, 2006. [Online]. Available: https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MANDUCHI1/Bilateral_Filtering.html. [Accessed: 30-Sep-2021].

[10] S. Paris, MIT CSail, 2007.

[11] J. Canny, "A computational approach to edge detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-8, no. 6, pp. 679–698, 1986.

[12] N. Ferdinand, "A Deep Dive into Lane Detection with Hough Transform," Medium, 2020. [Online]. Available: https://towardsdatascience.com/a-deep-dive-into-lane-detection-with-Hough-transform-8f90fdd1322f. [Accessed: 2021].

[13] S. Lee, "Lines detection WITH Hough transform," Medium, 02-May-2020. [Online]. Available: https://towardsdatascience.com/lines-detection-with-Hough-transform-84020b3b1549. [Accessed: 30-Sep-2021].

[14] D. Liu, Y. Wang, T. Chen, and E. T. Matson, "Accurate Lane Detection for Self-Driving Cars: An Approach Based on Color Filter Adjustment and K-Means Clustering Filter," International Journal of Semantic Computing, vol. 14, no. 01, pp. 153–168, 2020.

[15] M. Aly, "Real time detection of lane markers in urban streets," 2008 IEEE Intelligent Vehicles Symposium, 2008.