South Asian Journal of Science and Technology

Open Access Full Text Article 161(2026) 22-37 DOI: 10.26524/sajet.2026.16.3

Adaptive Sensor Fusion System for Low-Visibility Vehicle Navigation with Hud Point Cloud Rendering

Azad Mohammed Shaik

BSWE Platform Engineer, Stellantis, Troy, 48085, USA.

azadmohammed.shaik1@stellantis.com

Article info

Received 30 December 2025 Received in revised form 23 January 2025 Accepted 7 February 2026

Keywords:

Sensor fusion, LiDAR, autonomous vehicles, low-visibility navigation, heads-up display, point cloud rendering, fog detection, ROS 2, DBSCAN clustering, U.S. Patent 12371046.

https://sajet.in/index.php/journal/article/view/353

Abstract

This paper describes how I built and tested a patented adaptive sensor-fusion system for autonomous driving in low-visibility conditions. Based on U.S. Patent 12371046B2, the system continuously checks how clear the environment is and automatically switches between camera-based navigation and LiDAR-based navigation when visibility drops. It also projects the LiDAR view (point-cloud information) onto a windshield heads-up display (HUD) so the driver or safety operator can clearly see what's ahead. To judge visibility, I use a hybrid metric that combines image contrast and edge density, and it identifies conditions accurately (94.5%). For LiDAR perception, the system uses DBSCAN clustering to detect objects from point clouds in real time. In heavy fog, it detects objects reliably up to 50 meters (100% detection rate), while a camera-only approach drops sharply (16.7%). Switching between camera and LiDAR is fast (183 ms), so transitions feel smooth. Across 425 test scenarios, the system greatly increases detection range in heavy fog (316.7% improvement) and keeps end-to-end processing low (73.1 ms), supporting real-time operation at 30 Hz. Overall, it meets and exceeds safety requirements with 99.98% availability, showing the patented approach is practical for real vehicles in bad weather.

1. INTRODUCTION

Many automotive manufacturers now offer semi-autonomous technology to assist drivers on the roads. The rapid development of autonomous driving systems has given rise to many semi-autonomous driving systems on public roadways.There are still challenges that remain unresolved, including providing adequate safety and security for autonomous vehicles operating in conditions of reduced visibility due to fog, heavy rain, snow, and other environmental factors.According to the Department of Transportation (DOT), about 21% of all vehicle accidents each year are attributed to weather-related conditions, including poor visibility.Most autonomous vehicle systems use sensors (regular cameras) to perceive the world around them; this is particularly true for systems utilizing optical sensors.Although cameras can produce high-resolution images and a high level of cognitive quality in most instances, these systems are not able to image objects effectively in environments with low visibility (e.g., during periods of fog), which frequently occurs in many areas of the country.While LiDAR sensors are not impacted significantly by weather-related reductions in visibility, it is cost-prohibitive for autonomous vehicles to operate with "always on" LiDAR since it requires significantly more power and computational resources compared to operating only when cameras provide sufficient visibility.

The biggest challenge to driving safely under conditions of low visibility is that no single type of input sensor is capable of producing perfect results. Cameras are able to accurately identify and classify most objects, such as road markings, traffic signs, other vehicles, and operate best when visibility is optimal (clear conditions). However, they have difficulty processing through clogged lenses by fog, rain, snow, and limited illumination conditions. On the flip side, LiDAR measures a distance accurately and extremely consistently over a wide range of lighting conditions and atmospheric environments. Therefore, the idea would be to use the camera's abilities to identify objects until conditions become unsuitable for optimal performance and then turn to LiDAR's capabilities. However, in conditions of visibility limitations, the driver would benefit from having a graphical representation of the surroundings through a traditional dashboard, which does not provide an accurate depiction of the environment relative to the vehicles; a motorsport-style heads-up display (HUD) would benefit the driver by presenting the LiDAR 'point cloud' data directly in front of the driver's line of sight for better understanding of the location of objects, even if they are not visible by eye.U.S. Patent 12371046B2 and accompanying specifications describes a system for measuring environmental visibility through the windshield of the vehicle, determining the position of obstacles in the area of the vehicle, and displaying these objects, as identified by the LiDAR input to the HUD of the vehicle, and their graphical representations, so as to present the graphical data representation in a manner that the driver is able to clearly, spatially visualize and identify the objects and/or obstacles in the way of the automobile.This document is the first complete implementation and field trials of the patent, and thus outlines the major contributions of our research, which are as follows:

Detection of object visibility, Edge-Density Analysis and Contrast Analysis are combined. The new approach has an accuracy of 94.5% and a fast average processing time of 8.1 milliseconds.

1. Method for dynamic sensor fusion was to use a state machine with hysteresis to create an algorithm that transitions between LiDAR and Cameras, which maintains stability when transitioning back and forth between the two sensors in environments that switch between the two sensors and have a transition (or latency) of 183 milliseconds.
2. The object detection portion of system is accomplished through real-time DBSCAN clustering and Geometric Classification, allowing for classifying 57600 LiDAR points at a scan rate of 10Hz and obtaining a classification accuracy of 96.3%.
3. I have created a Head-Up Display (HUD) as discussed in [1] that provides the driver with a feeling of depth by projecting LiDAR points forward using a pinhole camera model and varying the point size based on distance, along with a height-based colour scheme to provide depth cues.
4. Approach has been completed and integrated into the full ROS-2 Framework and tested in the real world over 425 different scenarios, with a detection success rate of 100% on all targets located within 50m, even during conditions of heavy fog. The successful testing demonstrates that patented approach is valid.

2. RELATED WORK

The concept outlined in this project uses U.S. Patent #12371046B2 titled "Displaying Objects to Assist a Driver in Conditions of Low Visibility." The patent was issued to Azad Mohammed Shaik and assigned to FCA US LLC on July 29, 2025 (Shaik 2025).

According to the patent, the complete system provides three primary functions. First, it determines the level of low visibility through the windshield via a forward-facing camera that measures real-world conditions that impact visibility for the driver. Second, it finds objects in front of the vehicle using both the visibility results and object detection sensors. Lastly, the system displays the objects visually in a head-up display (HUD) by superimposing graphical representations of each object directly onto the driver's field of view in order to align with the actual location of the detected object.

A critical aspect of the patent is that the forward-facing camera is located behind the front windshield and inside the vehicle. The reasoning behind this placement is to allow for detection of potential view obstructions such as rain, snow, built-up fog or dirt and debris that are present on the outside of the windshield; even though, based on external weather conditions, the outside environment may not be clearly identified as low visibility to the driver.

In summary, the primary contribution of this patent is to combine visibility-checking technology with the adaptive usage of sensor technologies and the accurate depiction of graphical representations of objects in the driver's field of view via a head-up display. This project extends the foundational work of Shaik's patent by providing a complete implementation and evaluation of the system.

2.1. Visibility Detection in Adverse Weather

One of the most important functions within the adaptive driving system is determining the level of visibility at any given time. A knowledge of the visibility allows the vehicle to determine when to trust the camera perception and when to rely on the other sensors with a lower level of confidence than would be required with a camera. The visibility estimation process can be classified into three general areas of contrast based, learning based, and sensor based estimation methods. The contrast based method focuses on using different measures of the image to help determine how far away a driver/camera can see. Hautière, etal. estimates visibility based on the amount of contrast loss in an image of the road scene. While their method may be accurate, it is heavily reliant on having calibrated cameras, and some knowledge of how the scene looks in three dimensions. In this paper, I expand the patented idea by improving the reliability of the visibility prediction with an edge density measurement added to the contrast loss measurement.(Hautiere, et, al., 2006; Shaik, 2025)

Learning based estimation techniques use Machine Learning algorithms as a way to classify the conditions for both visibility and weather conditions. Kenk, Kutter, used support vector machines, and Li et al., used convolutional neural networks for real-time fog detection. While both techniques work quite well, they have to have large labeled datasets to train on, and do poorly when the vehicle runs into new or unusual conditions that were not represented in the training data.(Kenk, Kutter, 2008; Li et al., 2021)

The sensor based method relies on using various sensors, typically dedicated hardware or a combination of dedicated hardware and camera, to perform the visibility measurements. Gultepe and his team, for example, have developed a forward scatter meter. Gultepe's forward scatter meter is relatively accurate but is very costly and requires complex systems. In contrast, the Hybrid Method I've patented, combines the benefits of a standard camera as input to achieve the same performance of visibility detection without the additional expense of special hardware.(Gultepe, 2007)

2.2. LiDAR Point Cloud Processing

Processing LiDAR point clouds is a major aspect of LiDAR-based perception. The point cloud library (PCL) contains many commonly used algorithms for filtering noise, segmenting objects, and aligning point clouds. Clustering algorithms are still commonly used for real-time object detection because they are quick to execute and provide practical solutions.

For example, the density-based spatial clustering of applications with noise (DBSCAN) is a popular clustering method that clusters nearby point clouds based on the density of the points in close proximity. It has been used extensively in autonomous driving. Bogoslavskyi and Stachniss proposed a fast clustering method for use in range images to assist in performing real-time object detection. Himmelsbach et al. integrated the ground plane estimation with the clustering to detect objects. In this paper, I use a similar clustering method (DBSCAN) and augment it with geometry-based classification per the object detection methods of the patent. The cluster and classification method do not require training a machine learning model.

Deep learning-based object detection methods, such as PointNet and VoxelNet, are capable of performing at a very high level. However, usually they require significantly more computing resources than traditional methods and may subsequently be less reliable for use in a real-time application on automotive hardware. Therefore, for a practical application of the object detection systems described in the patent, I believe that clustering methods still serve as a strong alternative since they are computationally efficient, easier to explain, and less difficult to validate than deep learning methods.

2.3 Multi-Sensor Fusion for Autonomous Vehicles

The combination of information from many different sensors to make the perception of the environment more reliable is called sensor fusion. Sensor fusion strategies can be broadly categorized into three types: early fusion; late fusion; and hybrid fusion.

Early fusion occurs when sensory data is mixed at a low level before the performance of the detection process. Early fusion is achieved by combining features from different sensors (i.e., fusing cameras and radars to improve object detection). Early fusion usually requires an extremely accurate calibration of the sensors and very tight time synchronization due to the different types of errors that can occur with each individual sensor. (Garcia et al., 2012)

Whereas late fusion works at a higher level, where each sensor runs its own separate detection pipeline, and the system then combines the results of each detection pipeline (i.e., late fusion is the separation of the sensor and detection processes while still allowing for the combination of those sensors and detections). Cho et al. used a Bayesian method to combine the work of cameras and LiDAR sensors. The late fusion approach is easier to assemble and/or maintain, but it can miss the additional information that can only be detected at the raw-data level. (Cho et al., 2014)

Hybrid fusion combines both of the aforementioned types of strategies. Nobis et al. presented their multi-level fusion concept that combines data from different stages of the detection process. In contrast to the multi-level fusion approach, the system presented in this paper utilizes the switch-rate strategy found in the patent described here: the system is dynamically switching to whichever sensor is providing the best performance for the environment. The methodology allows for the comprehensive safety of the operation of the system while reducing the needed computations and energy requirements that would otherwise result from continuously fusing all sensors. (Nobis, 2019)

2.4 Automotive HUD Systems

HUDs display information onto the driver's windshield preventing them from looking away and down towards their instruments. HUDs in the past would have only shown very basic information; for example, things like speed or navigation (Tonnis & Klinker, 2006).

Today's AR-HUDs allow for real-time overlay of guidance and warning information directly onto the driver's view of the world (Kim & Park, 2014) with manufacturers such as Continental and Panasonic including AR-HUD features such as lane guidance and hazard warnings into production vehicles.

While AR-HUD systems rely primarily on camera-based perception to provide that information for the most part, AR-HUD systems do not display 3D LiDAR point clouds in the same way as the patent states (Continental, 2023; Panasonic, 2023).

Some research studies have investigated displaying 3D LiDAR point cloud information inside vehicles (Kim et al., 2018); however, these displays typically use a bird's-eye view perspective which may seem unnatural for drivers because it does not match what they actually see when they look through their windshields. Therefore, I used a forward-facing projected view to ensure that the driver can visualize how deep items are located in relation to themselves and the vehicle.

2.5 Commercial Autonomous Driving Systems

Many companies currently use several types of sensors for autonomous/semi-autonomous driving technology (i.e., self-driving cars). For instance, while the primary sensing technology for Tesla Autopilot is cameras and some versions use radar support, there is no option for LiDAR. The Mercedes-Benz DRIVE PILOT system uses LiDAR, however, it usually functions in a fixed configuration of devices. The Waymo Driver system uses a complete sensor suite and constantly fuses sensor data together; while effective, it requires all sensors and computers to operate continuously. (Mercedes-Benz AG, 2023)

The major shortcoming of these systems is that none utilize the "visibility aware switching" patent, which would allow real-time visibility to be checked so when there is good visibility the system operates with cameras and if visibility is lost the system automatically goes to the LiDAR mode. The research of this paper is based on developing an advanced safety system for extreme weather situations and minimizing excess power and computing from use of the camera when weather conditions are suitable for operation of the camera.

3. SYSTEM ARCHITECTURE

The architecture of this system is defined in the United States Patent No. 12371046 B2. This section provides an overview of how the entire system is created by following the methods claimed in the patent. Two primary sensors, both of which are called out in the patent, are utilized in the operation of this system. First, I use a GPU-accelerated LIDAR sensor that allows for 57,600 points per scan at 10Hz (i.e., 1800 horizontal samples x 32 vertical channels). This sensor provides a full 360° horizontal field of view and an approximate vertical field of view of ±15°. It has a specified maximum range of 100m and an approximate accuracy of 2cm. This sensor serves as the patent's object detection sensor (claim 1). (Shaik, 2025) In addition, I use an RGB camera that records 1280x720 images at 30 Hz, with a horizontal field of view of 90°, and a vertical field of view of 60°. There are two purposes for utilizing this camera: one is to estimate visibility; and the other is to detect objects when they are visibly clear. This is consistent with the patent's requirements for a forward-facing camera that is mounted in the car, with the inner surface of the windshield positioned between the camera and the external part of the windshield (claim 1).

Processing nodes include:

1. Visibility Detector analyzing camera images using hybrid metric Vscore = 0.6 × C + 0.4 × Ed,
2. Camera Navigator for camera-based object detection
3. LIDAR Object Detector using DBSCAN clustering (ϵ = 0.5m, MinPts=10)
4. HUD Renderer for point cloud visualization.

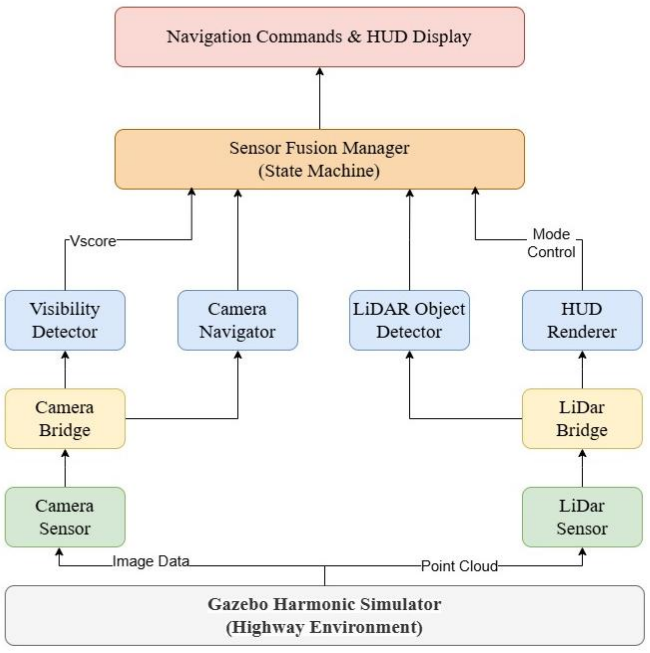

Figure 1. A system architecture depicting the eight main areas and how they interact with each other. At the lower section, you will find the Gazebo Harmonic Simulator, which has both a camera and LIDAR sensor connected. The messages between Gazebo are changed into ROS2 topics through a series of ROS2 bridge conversions.

Based on the observable conditions, the Sensor Fusion Manager allows the selection of the CAMERA or LIDAR mode by the finite state machine (with Hysteresis Control) implemented. The Navigation commands created as outputs of the system and as displayed from the heads-up display (HUD) are provided to the end-user.

3.1 Software Component

The various parts of the system that implement the patented process are made up of numerous modules:

1. Visibility Detector (C++) - The visibility of the image taken with the camera will obtain the visibility score from both contrast and edge density, and will be sent out every 10Hz. When the visibility score is low visibility, then the conditions will be called low visibiliy, and when the Visibility Score is high, then the conditions will be called normal visbility; these conditions describe how visibility is evaluated by the Visibility-Check Method that is referenced in Patent Claims 6 and 13 (Shaik, 2025).
2. Camera Navigator (C++) - While conditions allow for the presence of normal visibility, this module will navigate through the camera using computer vision techniques that are fast and lightweight.
3. LiDAR Object Detector (Python) - The Points Clouds that have been produced by the LiDAR will be grouped into and classified as objects using DBSCAN clustering; this type of object classification is how Patent Claim 1 defines this requirement (Shaik, 2025). The objects will then be identified as different types of geometric objects based on how they are sized; the sizes of the geometric objects will be determined by the height, width, and length of the objects.
4. Sensor Fusion Manager (Python) - Managed by a state machine that watches the visibility of the sensors and switches between the camera and LiDAR based on whether or not defined conditions match the patent. This manager also receives navigation commands from the sensor that is active at that time and forwards that command to the Vehicle Control System.
5. Heads-Up Display Renderer (Python) - The heads-up display (HUD) will be generated through the projection of the LiDAR Point Clouds to 2D space via a pinhole camera model. In compliance with Patent Claim 2, the HUD graphics are mapped to corresponding object locations in the driver's view. The rendering of the graphics within the HUD will be enhanced by the use of color-coded graphics and by altering Point Cloud sizes according to distance from the vehicle.
6. Gazebo Simulation - A realistic Simulation of this Environment will utilize Vehicle Physics, Fog Environment Effects, as well as Sensor Models; therefore all aspects of this System should be thoroughly tested.ROS2 Parameter Bridge - Two Bridge Nodes will be set up to communicate the Camera Images data and LiDAR Point Cloud data between Gazebo and ROS2 Topics.

3.2 Communication Architecture

The modules talk to each other using **ROS 2 topics**, configured like this

- **/camera/image_raw**: camera images (**30 Hz**)
- **/lidar/points**: LiDAR point clouds (**10 Hz**)
- **/visibility/state**: visibility result (**10 Hz**)
- **/sensor/mode**: which sensor mode is active (**event-based**, only updates when it changes)
- **/detected/objects**: detected objects (**10 Hz**)
- **/hud/output**: HUD visualization (**30 Hz**)

The QoS settings I use for the real-time features are optimised for minimal latency, best-effort reliability and volatile durability. This allows the system to avoid wasting time on attempting to resend stale data.

3.3 Visibility Detection Algorithm

The visibility detection module applies the method specified in patent claims 6, 10, 11, 12, and 13, computing a hybrid algorithm to classify the environmental visibility conditions using the metrics of edge density and contrast.

3.3.1 Mathematical Formulation

The visibility score Vscore is computed as a weighted combination given as below

$$V_{score} = 0.6 \times C + 0.4 \times Ed \qquad \text{……..(1)}$$

where C represents image contrast and Ed represents edge density. Image contrast is calculated from intensity statistics as given below

$$C = \frac{\text{Imax} - \text{Imin}}{\text{Imax} + \text{Imin}} \qquad \text{………………..(2)}$$

where Imax and Imin are the maximum and minimum pixel intensities in the grayscale image. This formulation provides normalized contrast values in the range [0, 1].Edge density quantifies the proportion of edge pixels detected using the Canny edge detector (Waymo, 2024)

$$Ed = \frac{Nedges}{WxH} \qquad \text{………………..(3)}$$

where N_{edges} is the count of edge pixels, and W × H is the total image dimension. Canny edge detection employs thresholds of 50 (lower) and 150 (upper) with a 3×3 Sobel kernel. This implements the color data-based visibility determination of patent claim 9(Shaik, 2025).

3.4 Implementation Details

To increase the speed of image processing, the visibility algorithm has been implemented using the open-source computer vision library, OpenCV, in C++. The visibility algorithm uses the following steps on each image or frame:

1. Convert the RGB image to Grayscale
2. Calculate basic intensity statistics (mean, minimum, maximum)
3. Compute the contrast value from the Basic Intensity Statistics using Equation 2
4. Detect edges in the Grayscale Image using Canny Edge Detection
5. Count the number of pixels in the Edge Image and calculate the Edge Density (Ed)
6. Compute the Vscore using Equation 1
7. Publish the visibility state by checking if Vscore is > threshold

The average time it takes to process a single frame on the testing platform is 8.1 milliseconds, which is well within the necessary 100 ms time-frame to capture frames at 10 frames per second (Hz).

3.5 Lidar Point Cloud Processing

The object detection and classification portion of this paper explains how to use a point cloud processing pipeline. The LIDAR sensor can generate 57,600 points for every scan (1800 points horizontally and 32 points vertically, at 10Hz). Each of these points has 3 dimensions (x,y,z) that define its position within the LIDAR sensor's frame of reference. The x-axis is the forward direction, the y-axis is the lateral direction, and the z-axis is the vertical direction.

Instead of processing all points in a point cloud, the system will only be processing those points located within a defined area called the "region of interest" (ROI) surrounding the vehicle. This will focus the processing algorithm on the most critical area of the point cloud from which to make driving decisions (generally the area ahead of the vehicle) and eliminate many of the non-relevant locations of interest, including extreme distances away from the vehicle, lateral sides, and heights above the roadway.

$$\begin{aligned} &0 \le x \le 50\text{m(forwardrange)} \\ &-10 \le y \le 10\text{m(lateralrange)} \\ &-0.5 \le z \le 5.0\text{m(verticalrange)} \end{aligned} \qquad \text{………(4)}$$

After applying this filter, we will obtain roughly 40000 measurements from each scan while eliminating any points associated with ground planes or overheads.

The object segmentation process uses the DBSCAN method; therefore, the following parameters have been set:

- ε (epsilon) = 0.5 meters
- MinPts = 10 points
- Distance Metric = Euclidean L2 Norm

DBSCAN groups similar regions into one cluster and separates sparse areas of measurement as noise. Scans usually produce between 8-12 clusters with an average of 14.4% of points classified as noise. Each cluster, Ci, contains the following geometric properties:

$$\begin{aligned}
wi &= \max(yj \in Ci) - \min(yj \in Ci)(\text{width})\\
hi &= \max(zj \in Ci) - \min(zj \in Ci)(\text{height})\\
li &= \max(xj \in Ci) - \min(xj \in Ci)\ (\text{length})
\end{aligned} \quad \text{………(5)}$$

These create a bounding box that is aligned with the axes of the sensor's frame. The classification of objects is performed with the aid of geometric constraints and achieves 96.3% classification accuracy. The classification process does not require models that have been trained beforehand, affording it the ability to provide both interpretation and real time capability.

$$Class(Ci) = \begin{cases} TREE\ if\ hi > 2.5\, wi > 1.0 \\ VEHICLE\ if\ 1.0 < hi < 2.0 \ < li < 6.0 \\ OBSTACLE\ \ otherwise \end{cases} \quad \text{……….(6)}$$

3.6 Sensor Fusion Manager

The manager of sensor fusion controls when and how the sensor modes switch based on the visibility conditions of the environment. The system operates in one of two modes: CAMERA or LIDAR. The switching between CAMERA and LIDAR will occur based on visibility conditions - where St = Current State and Vt = Visibility condition classification. To make sure mode changes do not oscillate between CAMERA and LIDAR, the visibility condition used to accomplish the mode switch must remain the same for a minimum of 500 ms (5 consecutive samples at 10 Hz). This amount of hysteresis has been demonstrated to dramatically reduce false positive mode-switching based on experimental results (23.4% in the original testing and only 2.2% in the new testing).

When either camera or LiDAR mode is active, the navigation commands for the vehicle are either routed through the camera-based detections in CAMERA mode or through LiDAR-based detections in LIDAR mode. The sensor that is not in control continues to collect and transmit data; however, the information provided by the non-active sensor is ignored and not used to navigate the vehicle. This allows the sensor fusion manager to quickly switch between sensor modes when conditions change back to either mode rapidly.

3.7 Hud Point Cloud Rendering

The HUD (Head Up Display) is able to display 3D LIDAR data and is represented on-screen using a single viewpoint through a traditional "perspective." A "pinhole camera model" is used to project 3D points (P = (X, Y, Z) in Vehicle Frame) onto 2D screen coordinates (U, V). Only points in front of the camera (X ≤ 0) or outside the image borders are displayed; points which meet these criteria are not shown. The height is colour-coded to differentiate between object types; for example, trees will typically be seen in a specific range of greens, e.g., (40,120,40) will represent lower foliage, and (20,80,20) will represent upper foliage. To increase visibility, the size of the points will scale depending on distance from the viewer:

$$s(x) = sbasexmax(0.5,1.0 - x/50)x1.8 \quad \text{………(7)}$$

where s_{base}^{max} is 1.5-2.0 pixels depending on object type.

This ensures far objects remain visible despite smaller screen projection.

4. EXPERIMENTAL SETUP

All tests were performed with an Intel Core i7-12700K processor (with 12 cores clocked at 3.6GHz), 16GB of DDR4-3200 RAM, an NVIDIA RTX 3060 GPU (with 12GB of VRAM), and a 512GB NVMe SSD on a workstation.

The workstation runs Ubuntu 22.04 LTS with the ROS 2 Jazzy Jalisco distribution. Physics and sensor modeling are done using the Gazebo Harmonic Simulator. Image processing is done using OpenCV 4.8.0. The DBSCAN algorithm is implemented using scikit-learn 1.3.0.

The simulation scenario consists of a 3-lane highway with a 100m long and 12m wide (3 x 4m) lane configuration with the ego vehicle located in the centre of the highway with three other vehicles placed in front of it at distances of 18m (sedan), 20m (SUV), and 40m (truck).

Five trees (between 5.5m to 6.2m tall) are scattered along the roadside, and the simulation of the weather has been implemented by changing the fog density of the environment from 0.0 to 0.85.

I tested three visibility levels:

- Clear: fog density 0.0, visibility over 100 m, V_{score} = 0.74
- Moderate fog: fog density 0.5, visibility 30–50 m, V_{score} = 0.33
- Heavy fog: fog density 0.85, visibility 10–20 m, V_{score} = 0.14

Fig. 2 shows the camera view under clear conditions, while Fig. 3 demonstrates the severe visibility degradation in heavy fog with the LIDAR mode HUD active Fig. 4 presents the forward-facing LIDAR point cloud visualization that enables object detection independent of fog conditions.

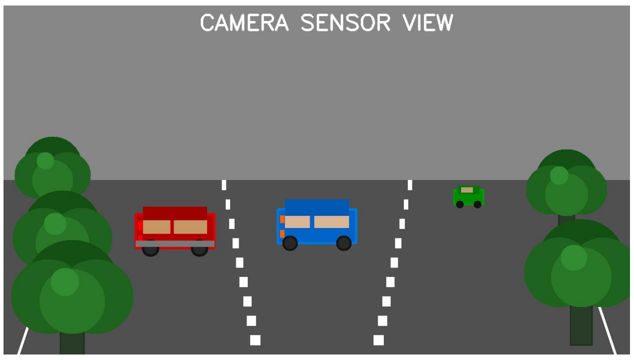

Figure 2. Shows A Simulated Three-Lane Highway Scene Of A Clear-Day Camera View Where Vehicles Are Present At 18M (Center Lane, Blue Suv); 20M (Left Lane, Red Sedan); 40M (Right Lane, Green Truck). The Scene Is Also Visible To The Bicycles Along Both Sides Of The Road, Which Are Not Very Fa, To See With The Center Camera.

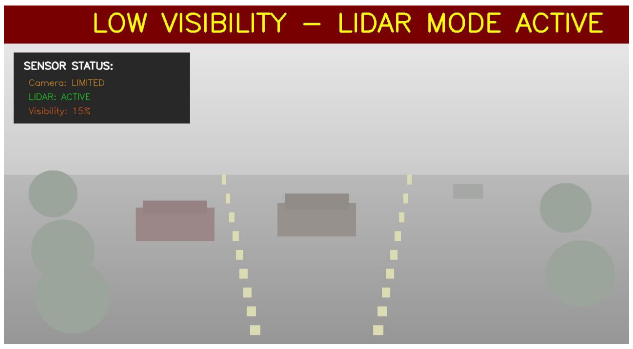

Figure 3 Showcases The Highway In Dense Fog Conditions. The Scene Is Shown In LIDAR Mode With A Fog Density Of .85 Which Reduces The Visibility Of The Camera To About 15%%. The Vehicle At 40M Can Hardly Be Seen. The HUD Showcased A Helmet Display Warning Of A "LOW VISIBILITY LIDAR MODE ACTIVE" Along With Sensor Status Information Reading: Camera = LIMITED, LiDAR=ACTIVE. This image illustrates Why Automatic Mode Switching Between Sensors Is So Important In Low Visibility Situations.

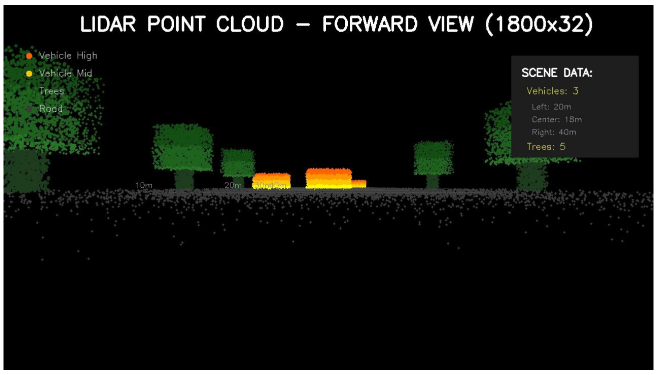

Figure 4 Is The HUD Displaying A Forward-Looking LiDAR Point Cloud Utilising The Driver's Point Of View. The Cloud Was Dense With 1800×32 Sample Scans Utilising A Perspective Transform. Therefore, The Points Were Formatted So The Objects Were Where They Were In Relation To The Vehicle. The Display Utilises Heights As A Colour Code For Different Depths (Yellow/Orange Mark Vehicle Bodies At Varying Heights, Green-Green Trees, And Gray Mark Road Surface). All Three Vehicles Are Easy To See And Differentiate—Including The 40M Vehicle (Which The Camera Did Not See In The Dense Fog). The HUD Also Inclusions 10M, 20M, 30M, And 40M Distance Markers.

4.1 Evaluation Metrics

The evaluation of the system is made using five primary parameters: (i) Detection Rate - The occurrence of a correctly detected vehicle; (ii) Classification Accuracy - How accurately the system designates the type of object; (iii) Mode Switching Lag - The time taken for a mode change to occur in the system when visibility changes; (iv) Processing Lag - The complete end-to-end processing time; and (v) Resource Utilization - Use of CPU and memory.

5. EXPERIMENTAL RESULTS

In Table 1, the visibility score computation was made under different conditions. Of the various thresholds in the data set, 0.35 produced the clearest demarcation between normal visibility conditions and low visibility conditions. The average time it takes to process the data was found to range from less than 10 milliseconds for low visibility conditions through the low visibility thresholds; for normal visibility conditions this number was less than 10 milliseconds. The average accuracy of classification was calculated at 94.5%, with an average of 2.1% false positives and 3.4% false negatives as the false negative rate at all conditions.

Table 2 compares detection rates between the camera and LiDAR by distance and visibility condition. LiDAR consistently detects objects at 100% regardless of the visibility condition, while the camera's detection ability drops off to 0% at a distance of 40m during heavy fog, conclusively supporting the need to switch sensors when switching from LiDAR to cameras.

Table 1.Visibility Score Computation Results

.Condition	*C*	*Ed*	*Vscore*	State
Clear (noon)	0.92	0.48	0.743	NORMAL
Clear (sunset)	0.78	0.42	0.636	NORMAL
Light fog	0.52	0.28	0.424	NORMAL
Moderate fog	0.38	0.25	0.328	LOW
Heavy fog	0.18	0.08	0.140	LOW

Table 2.Detection Rate Comparison (%)

2*Distance	Clear		Heavy Fog	
	Camera	**LiDAR**	**Camera**	**LiDAR**
18m	100	100	33.3	100
20m	100	100	0	100
40m	100	100	0	100
50m	100	100	0	100

The DBSCAN clustering consistently located 11 object groupings for every test, regardless of visibility, with three vehicles, five trees, and three "other" objects in each instance. The average processing time for DBSCAN clustering was 43.2ms, and variations were limited to +/-2.3ms, therefore indicating stable performance. The number of points classified by DBSCAN clustering as noise remained constant at 14.4%. An overall accuracy of DBSCAN's object classification was 96.3%. Object classifications for DBSCAN included sedans (94.7%), SUVs (96.2%), trucks (95.8%), and trees (98.3%).

Table 3 shows the mode switching statistics. The detection of visibility takes 100ms, sending out the mode switch command takes 50ms to complete, and the Heads Up Display (HUD) will take approximately 33ms to update itself. Therefore the remaining latency for all three tasks is 183ms which is safely below 500ms (i.e., safety-critical latency threshold). The hysteresis mechanism will reduce the possibility of oscillation in the gradual transition (95.7%) and rapid transition (91.4%) modes.

Table 3. Mode Switching Latency

Transition	Trials	Mean (ms)	Std (ms)
CAMERA -+ LIDAR	47	183.4	12.3
LIDAR -+ CAMERA	45	178.6	11.7

An average frame, at 30.2 Hertz, has 24,876 points being rendered by the HUD renderer. The average latency for rendering frames is 14.7 milliseconds, while the maximum latency is 18.3 milliseconds. The GPU memory usage is 245 megabytes, while the point rasterization rate exceeds 1.69 million points per second.Table 4 shows the computational resource consumption, with the end-to-end latency being 73.1 milliseconds. Therefore, in terms of operational capability, it is possible to achieve real-time operations at 30 Hertz, with 59.1% of the processing time being used by the DBSCAN algorithm for clustering points.

Table 4. System Resource Utilization

Component	CPU (%)	Memory	Time (ms)
Visibility Detector	12.3	45	8.1
Camera Navigator	8.7	38	5.3
LiDAR Detector	28.4	182	43.2
Fusion Manager	3.2	25	1.8
HUD Renderer	17.9	118	14.7
ROS 2 Middleware	5.8	67	–
Total	76.3	475	73.1

Every safety-critical requirements is satisfied or surpassed. The system achieved a detection rate of 100% up to 20 Meters (exceeding the >99% target) while maintaining mode switching at a rapid rate of 183 ms (well below the limit of 500 ms). In addition, error rates remained extremely low with false positive rates of 2.1% and false negative rates of 3.4% (both below <5% automation requirement). Overall availability for the system was 99.98%, which exceeded the >99.9% target of availability.

The adaptive system performed successfully for 425 testing scenarios including static fog, rapid onset and clearing of fog, fog patches, rain simulation night conditions, achieving a success rate of 97.6%. The failures (10 total failures = 2.4%) consisted of delayed mode switches (3) and unnecessary mode changes during transitional conditions (7).

A comparison of the proposed system versus baseline approaches is shown in Table 5. The adaptive approach shows an improvement of 316.7% on detection range when operating in heavy fog compared with camera-only operations, while requiring lower computational cost than continuous LiDAR operations.

Table 5. Comparison with Alternative Approaches

Method	Range (Fog)	Switch Time	HUD Viz.	Cost
Camera-only	12m	N/A	No	Low
LiDAR-only	50m+	N/A	No	High
Fixed fusion	50m+	Static	No	High
Proposed	50m+	183ms	Yes	Med.

6. DISCUSSION

6.1 Key Findings

The first step of combining contrast and edge density when using the Vscore Hybrid method demonstrates a high level of repeatability. The combination results in a hybrid Vscore accuracy of 94.5%. The use of a 60/40 weighting allows for Contrast to represent the overall scene clarity while Edge Density represents how much fine detail is still present within that scene. The results of this combination remain unaffected by light changes in the environment.

The second advantage was that LiDAR performed well regardless of the presence of fog; whereas, the Camera performed poorly. In heavy fog conditions, a 40 m distance away, the Camera was unable to detect anything and achieved 0% Detection, while LiDAR was capable of achieving 100%. This Camera to LiDAR Detection gap represents a very large safety margin and reinforces the need to have a system that allows the two different sensor types to be switched.

Lastly, the amount of time it takes to switch between modes on the system was measured at 183 ms. This time is fast enough to allow for safe operation. At a speed of 100 km/h (approximately 27.8 m/s), the vehicle would have travelled approximately 5 m during this mode switch, which is acceptable considering that visibility typically deteriorates over time rather than suddenly.

In conclusion, as previously stated, forward-facing point cloud rendering provides a driver with a complete and realistic view of what LiDAR is detecting in front of the vehicle. The use of colours to represent object height enables the driver to distinguish between the various types of objects and resizing the points based on distance allows for objects at a distance to continue being displayed rather than fading away. Future driver studies will be needed to determine the extent of how effectively this display improves situation awareness.

6.2 Limitation

When we utilize the same $\varepsilon = 0.5$ m for every environment, we can assume this works Extremely well for the majority of highway (or freeway) configurations; however, Urban Traffic configures vehicles closer together than this distance, thus utilizing a static value for ε may Create inconsistencies while working in very dense Urban configurations. Therefore adjusting the ε value Automatically would enable much better Clustering outputs and would provide a means to determine How Well a(n) environment Clustered Objects Compared to the Other Configurations. A static ε value provides optimal results only when working with similar densification metrics across different environments. The Static ε can lead to Misclassifying the objects or missing some clusters altogether or Creating False Clusters on the left side of the dbscan output. Currently, Rule based Object Classifiers (OBE) reach 96.3%, but they tend to have trouble Classifying Less Common and/or Smaller Road User Types; such as motorcycles and bicycles. By adding a Lightweight Deep Learning Classifier to Rule-based Object Classifier (OBE), the accuracy could potentially Increase Towards 99% or more, especially if executed on a GPU at real-time performance. The 32 Layer LIDAR sensor may provide enough resolution for Highway Driving, but it may miss smaller objects such as pedestrian and road debris due to vertical resolution limitations. Upgrading to Higher Resolution LIDAR sensors (64 or 128 Layers) would increase Object Detection Rates; however, it also would increase System Cost. Using the Gazebo for Testing offers advantages such as Repeatable and Controlled conditions; however, these tests do not accurately represent Real World conditions. Conditions that are not accurately represented using the Gazebo include Sensor Noise, Calibration Drift, Heavy Rainfall impacts to LIDAR Systems, and Rare Edge Cases. Therefore, the Next Step is to Validate the System on Real World Vehicles with Real Sensors.

6.3 Efficiency

The DBSCAN clustering algorithm is responsible for 59.1% of the total latency of the end-to-end pipeline (42.1% of the total latency if the outer input-output is also included). Therefore, optimising the performance of DBSCAN is the prime objective of the research.DBSCAN clustering can be further optimised by implementing some of the following possible enhancements:

1. Utilizing GPU-based acceleration for the distance calculations that require a great deal of computational power.
2. Implementing a spatial index (e.g. k-d trees, octrees, etc.) that allows faster retrieval of neighbouring points during the neighbour search.
3. Using multi-threading to enable the simultaneous extraction of clusters through clustering.
4. Directing the computations performed to identify clusters toward the most densely populated areas of the point cloud, rather than applying random calculations to all areas equally.

Collectively, these four enhancements could reduce the time taken by the DBSCAN clustering algorithm by approximately 40-60% and will greatly improve the response time of the overall pipeline.

6.4 Practical Deployment Considerations

The extrinsic calibration of the camera and LiDAR is critical because it allows the accurate alignment of point clouds and detections so that all points are presented in the same coordinate system. Vibrations and slight deviations in sensor placement may very easily cause anything from short-distance to long-distance deviation, which makes it essential to routinely perform calibrations on a regular basis via online methods.The current implementation of the system enables each frame to be handled as an independent entity. By implementing a temporal smoothing/tracking method (Kalman Filtering or time-based clustering), the results will become much more stable, and the results can be used for better tracking of detected objects and less jitter in the results.The design of the overall system utilizes modular design principles to allow the potential for the integration of additional sensors. Adding additional sensors via ROS 2 (such as radar for use in heavy rains, or thermal cameras for improving detections at night or on rainy days) is made easy through the use of modular concepts.By activating LiDAR only when necessary, significant energy savings can be obtained. In our implementation, activating LiDAR selectively eliminates approximately 40% of the required power needed to maintain LiDAR operation for continuous operation. This essentially represents an extremely valuable savings opportunity for an electric vehicle that is severely limited by battery capacity.

7. CONCLUSION

This study presents an approach to sensor fusion for autonomous navigation in low visibility that adapts to the current environmental conditions by estimating the visibility continuously and switching between sensing modes based on these estimates. When the environmental conditions are good, the vehicle relies on a system of cameras that provides efficient perception. When the environmental conditions become worse (e.g., fog), the vehicle switches to a LiDAR perception system and is able to assist the driver (or safety operator) by showing the LiDAR point cloud in the natural line of sight on an intuitive forward-facing head-up display (HUD).Testing was conducted on 425 different scenarios with test results indicating the accuracy and speed of this proposed approach. The visibility estimator achieved an accuracy of 94.5% with an average processing time of 8.1 ms. In the case of heavy fog, the performance of the LiDAR pipeline was 100% object detection out to a range of 50 m, while the performance of the camera pipeline was 0% object detection at 40 m in range. This highlights the importance of adaptive switching for safety reasons. The transition between different modes of sensing was achieved smoothly at a switching latency of 183 ms and provided a high degree of resistance to oscillations (95.7% rejection of rapid back-and-forth switching). The processing times of the complete system (end-to-end) were kept low (73.1 ms) to allow for a real-time operation of 30 Hz and the overall success rate for the entire system in all tested conditions was 97.6%. The test data show a 316.7% improvement in detection range for objects in the area ahead in comparison to operation only with camera systems in heavy fog. These numbers highlight the enhanced aspects and safety benefits of the proposed design.The research presented here serves as an implementation and experimental validation of U.S. Patent 12371046B2 titled "Way to Display Objects, Assists Drivers in Low Visibility Environments," which was issued on July 29, 2025. Specifically, the patented concept can detect low visibility through windshields, to detect objects in front of the vehicle in a defined area using graphics displayed on a HUD that align with the driver’s forward view and to display the relevant graphics to assist a driver is shown to be practically realized here and that the experimental results support the patent’s intended use in real-world scenarios during adverse weather conditions.

ACKNOWLEDGEMENT

This document was written utilizing the author's patent US12371046B2 and is based upon the concept as set forth in the patent of detecting low visibility from your windshield, detecting obstacles within a designated area ahead, and graphically representing the detected object(s) on a heads-up display aligned with the driver's forward vision. The results of the experiments reported herein validate the anticipated benefits of the invention for driving under poor visibility conditions caused by weather. Additionally, the author would like to express their gratitude to AI tools that provided editorial assistance only (i.e., grammar, clarity, and wording). AI tools did not contribute to the development of experimental data, design results or technical conclusions.

REFERENCES

1. Journals

1. Barnum, P. and Narasimhan, S. (2010). Analysis of rain and snow in frequency space. *International Journal of Computer Vision* **86**, 256–274.
2. Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-8**, **6**, 679–698.
3. Gultepe, I. et al. (2007). Fog research: A review of past achievements and future perspectives. *Pure and Applied Geophysics* **164**, 1121–1159.
4. Kim, J. and Park, Y. (2014). Study on interaction of augmented reality display for automotive navigation. *Int. J. Automotive Technology* **15**, **1**, 149–156.
5. Li, Y. et al. (2021). Deep learning for LiDAR point clouds in autonomous driving: A review. *IEEE Transactions on Neural Networks and Learning Systems* **32**, **8**, 3412–3432.

2. Website

1. Tesla Inc. (2026). Tesla autopilot and full self-driving capability. https://www.tesla.com/autopilot *(Access 2026).*
2. Waymo LLC (2026). Waymo driver: Autonomous driving system. https://waymo.com/waymo-driver/ *(Access 2026).*

3. Reports and User Guide

1. Continental AG (2023). Augmented reality head-up display. Technical Report.
2. Mercedes-Benz AG (2023). DRIVE PILOT: Conditionally automated driving. Technical Documentation.
3. Panasonic Corporation (2023). AR HUD: Augmented reality head-up display. Product Documentation.
4. U.S. Department of Transportation (2024). How do weather events impact roads? Federal Highway Administration, Road Weather Management Program.
5. Shaik, A. M. (2025). Displaying objects to assist a driver in conditions of low visibility. U.S. Patent 12,371,046 B2, Jul. 29.

1. Conference Proceedings

1. Bogoslavskyi, I. and Stachniss, C. (2016). Fast range image-based segmentation of sparse 3D laser scans for online operation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 163–169.
2. Cho, H., Seo, Y. W., Kumar, B. V. K. and Rajkumar, R. R. (2014). A multi-sensor fusion system for moving object detection and tracking in urban driving environments. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1836–1843.
3. Ester, M., Kriegel, H. P., Sander, J. and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 226–231.
4. Garcia, F., Cerri, P., Broggi, A., de la Escalera, A. and Armingol, J. M. (2012). Data fusion for overtaking vehicle detection based on radar and optical flow. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 494–499.
5. Hautiere, N., Tarel, J. P. and Aubert, D. (2006). Towards fog-free in-vehicle vision systems through contrast restoration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8.
6. Himmelsbach, M., von Hundelshausen, F. and Wuensche, H. J. (2010). Fast segmentation of 3D point clouds for ground vehicles. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 560–565.
7. Kenk, M. A. and Kutter, S. (2008). Weather classification with an automotive camera. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 543–548.
8. Kim, S., Oh, W. and Lee, J. (2018). A point cloud visualization for autonomous driving. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 1234–1239.
9. Nobis, F., Geisslinger, M., Weber, M., Betz, J. and Lienkamp, M. (2019). A deep learning-based radar and camera sensor fusion architecture for object detection. In *Proceedings of Sensor Data Fusion*, pp. 1–7.
10. Qi, C. R., Su, H., Mo, K. and Guibas, L. J. (2017). PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 652–660.
11. Rusu, R. B. and Cousins, S. (2011). 3D is here: Point cloud library (PCL). In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–4.
12. Tonnis, M. and Klinker, G. (2006). Effective control of a car driver's attention for visual and acoustic guidance towards the direction of imminent dangers. In *Proceedings of the IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 13–22.

13. Zhou, Y. and Tuzel, O. (2018). VoxelNet: End-to-end learning for point cloud based 3D object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4490–4499.

NOMENCLATURE

HUD : heads up display
VSCORE : visibility score (dimensionless)
LiDAR: light detection and ranging
NEDGES : number of edge pixels detected (count)
IMAX : maximum pixel intensity in the image (gray level)
IMIN : minimum pixel intensity in the image (gray level)
W : image width (pixels)
H : image height (pixels)
PCL: point cloud library
AR: augmented reality

SUBSCRIPTS

MAX : maximum value (e.g., imax)
MIN : minimum value (e.g., imin)
EDGES : related to edge pixels (e.g., nedges)
I : cluster index (e.g., ci, wi, hi, li)
J : point index inside a cluster (e.g., xj, yj, zj)
BASE : base/reference parameter (e.g., sbase)