**2024 NDIA MICHIGAN CHAPTER**
**GROUND VEHICLE SYSTEMS ENGINEERING**
**AND TECHNOLOGY SYMPOSIUM**
**MODELING, SIMULATION, PROTOTYPING & VALIDATION (MSPV) TECHNICAL SESSION**
**AUGUST 13-15, 2024 - NOVI, MICHIGAN**

# Validating Sensor Models for Off-Road Autonomous Mobility

**Andres Morales[1], Ryan Lamm[2], Sam Misko[1], Nic Bowen[1], Scott Bradley[3], Dr. Jayakumar Paramsothy[4]**

[1]University of Alabama at Birmingham, Birmingham, AL
[2]Southwest Research Institute, San Antonio, TX
[3]Keenesaw Research Center, Houghton, MI
[4]U.S. Army DEVCOM GVSC, Warren, MI

## ABSTRACT

*Within the scope of NATO AVT-341, a comprehensive framework to validate the fidelity of exteroceptive sensor models, used in the pipeline for simulating diverse military autonomy maneuvers, was developed. An experiment was designed, conducted, and metrics analyzed. This technical paper describes the sensor model validation framework and the results obtained from the conducted experiment.*

**Citation:** A. Morales, R. Lamm, S. Misko, N. Bowen, S. Bradley, J. Paramsothy. "Validating Sensor Models for Off-Road Autonomous Mobility," In *Proceedings of the Ground Vehicle Systems Engineering and Technology Symposium* (GVSETS), NDIA, Novi, MI, Aug. 13-15, 2024.

## 1. INTRODUCTION

Validated sensor models are essential for the effective use of virtual assessment tools in the planning of missions involving autonomous military ground systems. The error attributable to poor sensor models needs to be understood so that operational trust in virtual simulation tools can be established. While significant effort has been put into validating on-road sensor models for on-road autonomous maneuvers by industry, little work has been done off-road – where typical military operations are envisioned. NATO AVT-341 was designed and experimented with to prove a framework necessary for validation and to derive sensor-level metrics that map to operational/scenario metrics. Specifically, it identified how multi-dimensional the validation of sensor models can be, the dependence on the requirements of the target perception algorithms in the autonomy stack, as well as the importance of a good virtual environment model.

### *1.1. Sensor Model Validation Framework*

To measure the effect of sensor model performance on general mobility assessment methods and tools for autonomous military ground systems, NATO AVT-341 developed

a validation framework as shown in Figure 1. The simplified sensor model validation experiment was not intended to be an exhaustive test and was not intended to measure the accuracies of a virtual environment or vehicle model or the performance of the perception/autonomy algorithms within an Autonomy Stack. Instead, the simplified sensor model validation experiment was meant to allow for an initial comparison of physical sensor outputs collected over multiple tests (physical experiment), to the virtual sensor model outputs in a simulation environment.

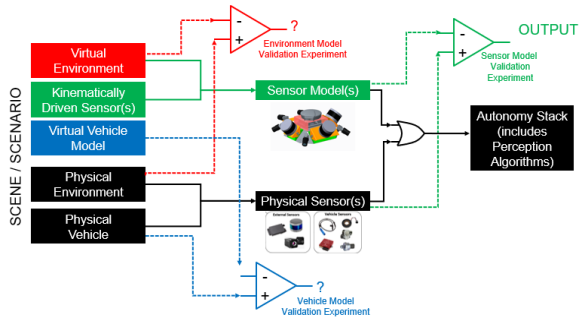


**Figure 1:** Validation Framework

### *1.2. Object-Level Sensor Validation*

In the context of simulating virtual exteroceptive sensors for military autonomous vehicles, a validation methodology that focuses on objects is crucial as it aligns closely with the real-world challenges and requirements faced by these vehicles. Motivation for this approach includes:

- Autonomous vehicles need to operate effectively in environments where they interact with objects (e.g., static objects, pedestrians, and vehicles), therefore, accurate object detection, identification, and tracking are paramount activities that must be closely replicated in the virtual environment, especially if given objects are mission critical.
- Virtual sensors must behave realistically, and validating their performance concerning objects is crucial to ensure the vehicle's ability to make informed decisions (e.g., braking and steering to avoid a collision).
- Objects are versatile and can appear in different forms and under various conditions, thus, a validation methodology that focuses on objects allows for its application in a wide range of scenarios.
- An object-level approach provides robustness against sensor pose estimation error. This approach maintains the ability to reliably identify and assess object characteristics within the scene, regardless of any deviations in sensor pose caused by measurement errors.

## 2. PHYSICAL EXPERIMENT

The physical experiment was set up in a flat, open test area of the Keweenaw Research Center (KRC) and consisted of two segments [Figure 2].

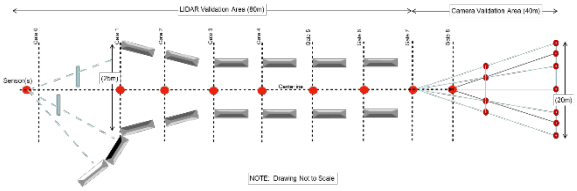


**Figure 2:** Simplified Sensor Model Experiment

The first segment consisted of a set of jersey barriers, arranged at specific gate locations, in a gauntlet configuration [Figure 3]. The jersey barriers alternated between white and orange along the centerline. The jersey barriers were 2.0m in length, 0.6m in width, and 1.1m in height. During the physical and virtual tests, the sensor was excited such that the yaw-pitch-roll (YPR) of the sensor reference changed in a manner sufficient to represent a non-smooth surface. Four-inch channels were used to induce the excitation [Figure 4]. To vary object reflectivity, two of the jersey barriers had panels installed painted opaque black and reflective white.

**Figure 3:** 1st Segment - Experiment Setup – KRC

**Figure 4:** YPR Excitation - KRC

The second segment consisted of 11 survey poles placed at specific locations such that as the vehicle approached the poles, the visibility of the poles in the far field would be occluded by the poles in the near field at specific locations [Figure 5]. The poles were 2.54 cm in diameter and 3.65 m in height.

**Figure 5:** 2nd Segment - Experiment Setup - KRC

To eliminate errors from different vehicle models, as the sensor was moved through the physical environment, the kinematics of the sensor were recorded so that the sensors could be kinematically moved similarly through the virtual environment without the need for a specific vehicle model. Eliminating the vehicle model from the sensor model validation experiment further reduced error in the analysis. The host vehicle was driven manually in a straight line through the segment.

The experiment was also replicated at Southwest Research Institute (SwRI) in San Antonio, TX United States as shown in Figure 6.

**Figure 6:** Physical Experiment Scenario – SwRI

## 3. VIRTUAL SIMULATION

### *3.1. Scene Modeling*

For capturing the KRC test area, multiple survey-grade high-resolution point clouds were stitched together. These point clouds were used to recreate the test area in Unreal Engine 4, and accurately place the barriers, cones, and poles in the virtual scene.

It was apparent during the environment modeling process that the tree line in the background was key in increasing the realism of the environment, so a two-dimensional billboard virtual representation was added.

Additionally, it was essential to closely replicate the environmental lighting from the real world within the virtual scene. This was achieved by comparing the Hue, Saturation, and Value (HSV) of pixels in the barriers between the empirical data and the outcomes of the simulation. Hue represents the color of the pixel, Saturation measures the intensity, and Value measures the brightness. The simulation's Cloud Cover percentage was adjusted to directly influence the saturation and value of the pixels. Consequently, a Cloud Cover setting of 45% was chosen to

align with the peak values observed in the real-world data [Figure 7]. The task of accurately modeling the sky's appearance and environmental lighting presented significant challenges, primarily due to the limited availability of tools suitable for achieving precise calibration. In general, simulated environments underscore the intricate complexity of natural conditions.

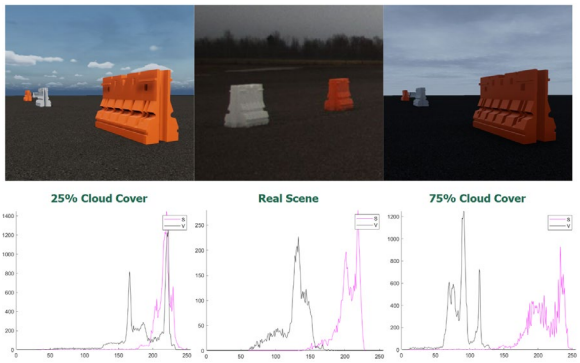


**Figure 7:** Cloud Cover %, Saturation, & Value

### *3.2. Sensors' Pose & Twist Estimation*

Odometry data recorded from the physical experiment was used to position the sensors in the virtual KRC scene. The position and orientation of the odometry were interpolated to match the sensors' timestamps. The virtual experiment was run by positioning the vehicle in a pose, pausing to allow the sensor data to be captured (camera image or LIDAR point cloud), and advancing the vehicle to the next recorded position. These steps are repeated until the vehicle reaches the final recorded position. For the LIDAR simulation, it was necessary to add a slight delay (10 HZ) between moving the vehicle and capturing the sensor to simulate motion distortion (included in the dSPACE AURELION LIDAR models). This approximates the kinematics of the sensor in the physical experiment.

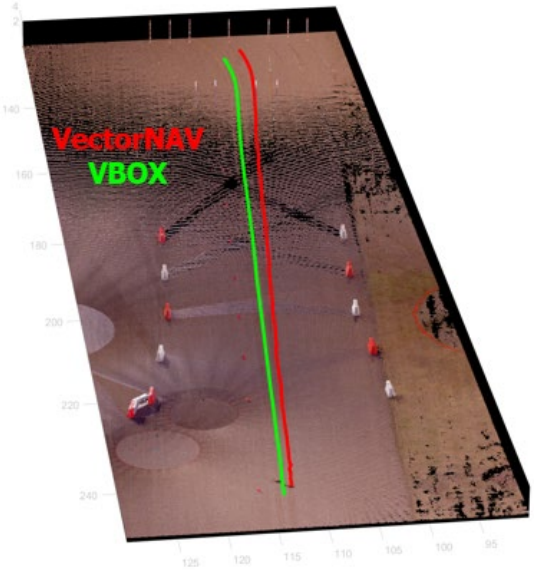


**Figure 8:** Difference in Localization Estimation

**Figure 9:** Real Camera (LEFT) and Virtual Camera (RIGHT)

The vehicle counted with two localization systems, a VectorNAV VN-300 DUAL GNSS/INS and a Racelogic VBOX. However, only the data from the VBOX demonstrated sufficient accuracy to carry out a validation [Figure 8].

## 4. CAMERA RESULTS & ANALYSIS

The experiment used a FLIR ORYX camera with a global shutter. The camera parameters and intrinsics were modeled in dSPACE AURELION. The simulation includes the distortion caused by the curvature of the lens.

Figure 9 illustrates both the actual and simulated camera outputs for a specific frame from the real-world test and simulation.

### *4.1. Analysis Software*

Scripts were developed within MATLAB for evaluation of the real and virtual camera raw images. Within MATLAB, the integrated ROS Toolbox and Image Toolbox are required to process the sensors' output and analyze the sensor's raw output. To reduce file size and computation time, the videos were downsampled to 15 FPS (from their native 60 FPS).

### *4.2. Camera: Object Detection*

Barriers, cones, and poles from the test scene were selected as relevant objects of interest. Therefore, the following metrics describe how these objects are perceived by the camera sensor [Figure 10].

The first step to computing metrics is to identify and isolate the objects of interest in the raw image. Several algorithms were evaluated before selecting a color-thresholding-based algorithm for cones and barriers, and a scale-invariant feature transforms (SIFT) algorithm for the poles.

**Detection of Orange Barriers & Cones**

These plastic objects present a highly visible color that is not common in other pixels in the scene. The exception is a few pieces of trash in the real scene that happen to trigger the algorithm (the exclusion of these items on the virtual scene would be considered an environmental modeling error). Thus, for every frame, the color-based-algorithm is applied with the following steps:

1. *Color-Thresholding*: The raw image is transformed into a logical array where pixels meeting the specified RGB values are marked as 'true'. This yields a binary image.
2. *Closing*: A morphological closing is performed with a 16-pixel radius disk. This operation fills holes in the binary image caused by small variability in the pixels' RGB values.
3. *Minimum Size Thresholding*: The image is analyzed for connected regions of pixels. If a region has less than 1,000 pixels, the region is discarded. This step prevents the identification of objects that are considerably far from the camera and adds robustness against detecting trash in the real test scene.

**Detection of White Barriers**

In contrast to the orange barriers and cones, the RGB values for pixels present in the white barriers are commonly present in other areas of the scene, particularly in the sky. Therefore, two additional steps were added to achieve a satisfactory segmentation:

4. *Maximum Size Thresholding*: Large areas of the sky share similar RGB values to the white barriers. Therefore, an additional step to discard regions that contain more than 20,000 pixels was included.
5. *Semi-Automated Filter for Centroid's Vertical Location*: Some small areas of the sky were still being identified by the segmentation algorithm. Therefore, a vertical filter was added to discard connected pixel regions with a centroid above a threshold (barriers are in the bottom portion of the frame).

Nevertheless, the vertical location of the horizon is changing as the vehicle is moving. Therefore, a semi-automated algorithm was implemented to change the threshold value based on the frame number.

**Detection of Poles**

The poles presented the most challenges for object detection due to:

- *Narrow Size*: The poles are only a few pixels in width.
- *Generic Colors*: The poles contain a white section that is often indistinguishable from the sky. Additionally, the reddish sections of the poles when compared to the high-visibility barriers/cones. Furthermore, the objects were modeled with the same orange color as the barriers while the real objects present a slightly purplish tone (environmental modeling inaccuracy).
- *Non-Uniform Size and Orientation*: The poles differ slightly in vertical height and orientation as they are easily influenced by the wind.
- *"Poking-into-Sky":* As the vehicle approaches the poles, portions of the poles cross the horizon, which is not clearly defined due to the presence of trees in the background. This behavior makes the poles harder to discriminate from the sky.

To identify the visual features associated with the poles, the Scale-Invariant Feature Transform (SIFT) algorithm was employed. The SIFT algorithm is a computer vision technique used for identifying and describing distinctive features in an image [1]. Some relevant key principles of SIFT features include:

- *Descriptor*: Each SIFT feature is described by a vector of values that capture information about its vicinity. These descriptors are used to match features between images.
- *Scale Invariance*: SIFT features are designed to be invariant to changes in scale. This means that the same feature can be detected at different scales within an image, making it robust to objects changing in size. This is important as the poles increase in size as the vehicle approaches them.
- *Rotation Invariance:* SIFT features are also invariant to image rotation. They can be detected and described consistently regardless of the orientation of the object within the image. This is advantageous as the pole's orientation fluctuates due to roll in the vehicle and wind deflection.

To implement this algorithm, the local SIFT features for a reference pole (both real and virtual poles) were calculated. Subsequently, for every frame, the local SIFT features were calculated and compared with the reference descriptors. Any SIFT features in the frame within a certain threshold from the reference feature descriptors were identified as matches and subsequently recognized as points on the pole.

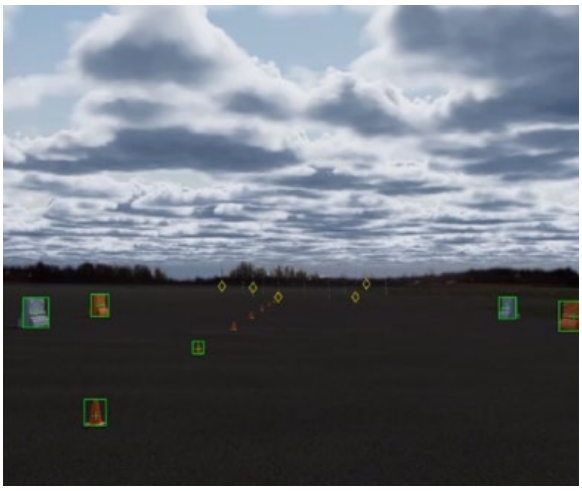

**Figure 10:** Object Detection of Barriers & Cones [Green Rectangles] and Poles [Yellow Diamonds].

### *4.3. Object Detection & Tracking*

A valuable tool for simplifying the complexities of any object detected by a camera sensor is a bounding box. A bounding

box is fully defined by four variables: horizontal location, vertical location, width, and height (when the orientation is constrained, which is common practice to avoid "slanted" boxes). In this section, the objects' locations and dimensions of their corresponding bounding boxes are compared.

## Bounding Boxes

The bounding boxes' dimensions are of particular significance as their dimensions are minimally influenced by the camera's orientation. Therefore, any variations can be more precisely attributed to disparities in virtual camera modeling.

Figure 11 indicates the bounding box dimensions (width & height) for a particular orange barrier and cone in the scene. In general, virtual bounding boxes have a reduced size when compared to real bounding boxes. This effect causes virtual objects to be initially detected at a later stage, as the camera needs to get closer for objects to reach the minimum size threshold. A visual assessment suggests that the virtual camera may have a slightly wider field of view, resulting in a reduction in the number of pixels occupied by objects of interest. Furthermore, the bounding box dimensions display a noticeably erratic behavior in the virtual scenario. These may be attributed to rendering artifacts appearing near the edges of the objects. However, a comprehensive understanding of this behavior requires further exploration and analysis.

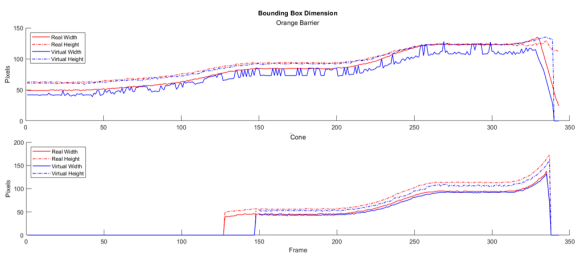


**Figure 11:** Bounding Box Dimension

## Object Tracking

Although the center of a bounding box may be used to define an object's location, the centroid (geometric center) of objects was selected as the variable of interest. As objects have irregular shapes, capturing the centroid provides a more accurate representation of the object's true visual center. Furthermore, the center of the bounding box relies solely on the most extreme (location-wise) pixels detected within an object, while every pixel contributes to the location of the centroid.

For illustration, three objects of interest were selected: a specific cone, a specific orange barrier, and a specific white barrier. Figure 12 compares the trajectories of the centroids across the frame as the vehicle advances through the scene. While they exhibit a good trend agreement, the trajectories present a significant offset which may be explained by an incorrect initial alignment between the real and virtual camera.


**Figure 12:** Trajectory of Various Objects

To mitigate this misalignment, calculating the centroid's velocity (pixels displaced per second) provides a more accurate metric directly linked to the actual sensor behavior and the virtual model. The horizontal and vertical velocity components for the objects of interest are illustrated in Figure 13.

The velocity analysis reveals significantly more unpredictable behavior in the virtual centroids' locations compared to their real counterparts (this is similar to the bounding box dimensions). This irregularity persists even during intervals when the vehicle (and

consequently, the camera) remains stationary. Additionally, a minor delay between the real and virtual sensor data is evident, possibly arising from timing synchronization issues between the odometry and the camera.

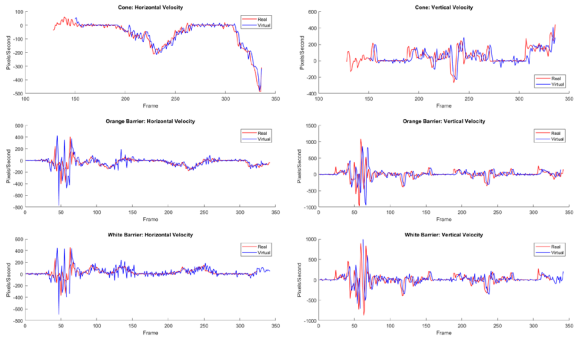


**Figure 13:** Centroids' Velocities

### *4.4. Hu Moments of Inertia*

The Hu Moments of Inertia (or Hu Invariants) are a set of mathematical moments used in image processing and computer vision to describe the shape of an object [2]. A characteristic of interest in the scope of the project is that these metrics are invariant to translation, rotation, and scale transformations, which turn these into robust metrics against errors in sensor pose estimation. Therefore, applying the Hu Moments of Inertia to the global grayscale raw image provides numerical data for the image structure, which can be used to identify differences between the real sensor and the virtual sensor. There are seven identified moment invariants ($I_1$-$I_7$), however, for illustration, the first Hu Moment of Inertia ($I_1$) may be calculated by:

$$I_1 = \eta_{20} + \eta_{02}$$

Where $\eta_{pq}$ is the translation and scale invariant given by:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\left(1+\frac{p+q}{2}\right)}}$$

Where $\mu_{ij}$ is the central moment of an image. For a digital image, the equation is:

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y)$$

Where $x$ and $y$ indicate the position of a pixel, $\bar{x}$ and $\bar{y}$ the location of the image centroid, and $f(x, y)$ the pixel value.

The Hu Moments of Inertia may be applied to binary images following segmentation, where $f(x, y) = \{0,1\}$. These metrics yield results highly oriented to the detection of objects of interest.

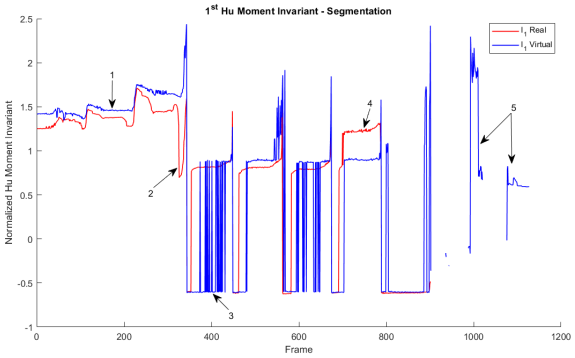


**Figure 14:** 1st Hu Moment of Inertia

Figure 14 compares the evolution of the 1st Hu Moment ($I_1$) as the vehicle advances through the scene, highlighting fundamental differences between real and virtual cameras. Some of the observed deviations include:

- The Hu Moment remains relatively steady as the vehicle crosses the YPR excitation zone, displaying its invariance to the camera's pose.
- Both the real and virtual cameras display similar trends as objects (barriers & cones) move across the frame. The periodic behavior between frames 340-800 is caused by the repeating pattern of equally spaced cones.
- Arrow 1 indicates a higher 1st Hu Moment for the virtual scene caused by a difference in the number of pixels identified as objects of interest.

- Arrow 2 indicates a large drop caused by a barrier leaving the frame in the real environment, while the barrier is still visible in the virtual frame (due to the wider view angle).
- Arrow 3 indicates that the virtual cones detection is flickering, while the real cones are steadily detected.
- Arrow 4 indicates a higher value compared to the previous periodic values. This is caused by a piece of trash present on the real scene that is incorrectly labeled as an object of interest.
- Arrows 5 indicate false positive detections in the virtual scene, which is caused by segments of the poles being incorrectly identified as cones/barriers.

### 4.5. Pole Detections

A simple and effective metric for assessing the sensors' perception of poles is the count of SIFT detections over time. Figure 15 reveals that poles are detected earlier in the virtual scene, but detection vastly improves in the real scene as the vehicle approaches the poles. In the virtual environment, detection proved challenging when the pole had the sky as its background. False detections were common in the virtual scene, while non-existent in the real scene. Further efforts may consider the accuracy and location of detections.

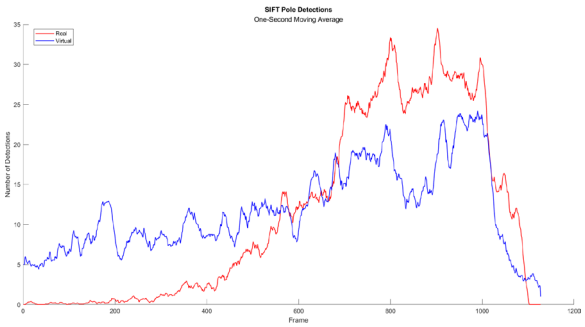


**Figure 15:** SIFT Pole Detections

### 4.6. General Global Metrics

In the following subsections, the raw image from the sensors is identified as an object of interest. Therefore, global metrics that are calculated from the whole image can be used to identify differences between the real and global cameras. As environmental lighting modeling remains a challenging area, the selected global metrics focus on the structure of the image. These metrics tolerate differences in lighting (e.g., brightness) related to environmental modeling errors not attributable to sensor modeling.

### 4.7. Global Hu Moments of Inertia

The Hu Moments of Inertia may be applied to the global grayscale raw image where $f(x, y) = [0,255]$. These metrics provide numerical data for the image structure, which can be used to identify differences between the real sensor and the virtual sensor.

While a comprehensive interpretation and analysis of the Hu Moments of Inertia are provided in the object-level analysis section, they are intentionally excluded from this section, which highlights the significance of Hu Moments as potent indicators for detecting areas of variance, even before the image segmentation process begins.

### 4.8. Non-Reference Quality Metrics

Several efforts have been dedicated to establishing an objective metric for image quality. A valuable quality metric correlates well with the subjective perception of quality by a human. If a reference image is not available (as is the case in this project), no-reference algorithms use statistical features of the image to evaluate image quality. All no-reference quality metrics usually outperform full-reference metrics in terms of agreement with a subjective human quality score. Some of these no-reference algorithms are:

- *BRISQUE*: Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE). A

BRISQUE model is trained on a database of images with known distortions, and BRISQUE is limited to evaluating the quality of images with the same type of distortion. BRISQUE is opinion-aware, which means subjective quality scores accompany the training images [3].

- *NIQE*: Natural Image Quality Evaluator (NIQE). Although a NIQE model is trained on a database of pristine images, NIQE can measure the quality of images with arbitrary distortion. NIQE is opinion-unaware and does not use subjective quality scores [4]. The trade-off is that the NIQE score of an image might not correlate as well as the BRISQUE score with human perception of quality.
- *PIQE*: Perception-based Image Quality Evaluator (PIQE). The PIQE algorithm is opinion-unaware and unsupervised, which means it does not require a trained model. PIQE can measure the quality of images with arbitrary distortion and in most cases performs similarly to NIQE. PIQE estimates block-wise distortion and measures the local variance of perceptibly distorted blocks to compute the quality score [5].

In the scope of this project, a difference in quality indicates a difference in statistical measurements performed in the camera's output. Therefore, no-reference quality metrics serve as an additional validation metric. As the virtual sensor model departs from realism, the quality metrics will correspondingly diverge.

## BRISQUE & NIQE

The BRIQUE and NIQE scores present a notable difference between the real and virtual images' scores. On both evaluators, a smaller score indicates a better perceptual quality. Interestingly, the algorithms present conflicting results. The BRISQUE scores assess the real images as having a higher quality than the virtual images, while the NIQE score considers the real images to be lower quality than the virtual cameras. As the scores are assigned by trained algorithms, the causes for this difference are unlikely to be explainable, without additional insight into the model training database and intricacies. Nevertheless, while BRISQUE aims to capture structural and texture-based artifacts and tends to rank images with these artifacts as lower in quality, NIQE primarily focuses on naturalness and may rate images with these artifacts as higher in quality. In both cases, the scores have an oscillatory behavior which could be caused by the movement of the vehicle [Figure 16].

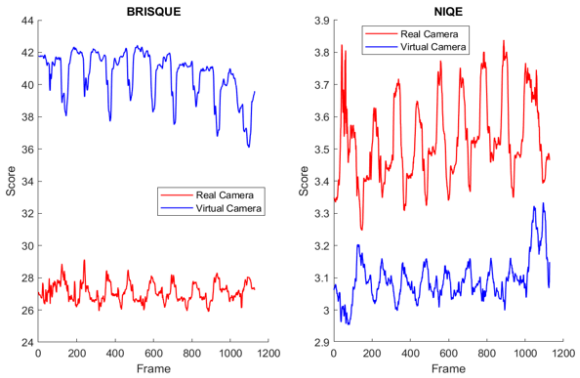


**Figure 16:** BRISQUE & NIQE Scores. A smaller score indicates better perceptual quality.

## PIQE

The PIQE scores are closely aligned with the BRISQUE scores, where the real images are considered higher quality than their virtual counterparts [Figure 17]. The relationship between the PIQE scores and quality range can be categorized as follows:

**Table 1**. PIQE Scores & Quality.

| Score Range | Quality |
|---|---|
| 0-20 | Excellent |
| 21-35 | Good |
| 36-50 | Fair |
| 51-80 | Poor |
| 81-100 | Bad |

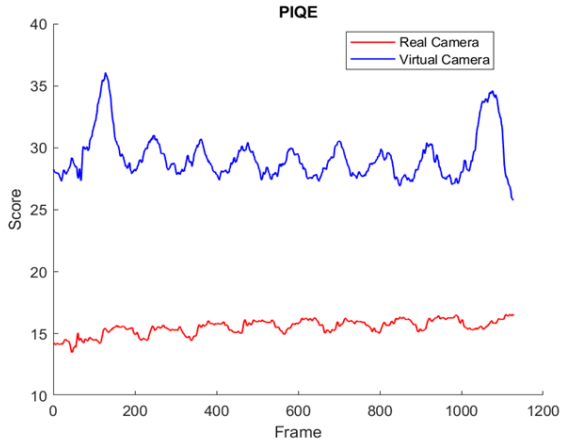


**Figure 17:** PIQE Score. A smaller score indicates better perceptual quality.

While the entirety of real images is classified as "Excellent", almost all virtual images are considered "Good" with a few instances of "Fair". Similarly, to the previous algorithm, the oscillating behavior is also present.

An advantage of the PIQE algorithm is that beyond providing an overall score, it also measures the local variance and may indicate blocks of the image that have high spatial activity, noticeable artifacts, and/or noise. Active Blocks represent areas with increased spatial variability, often caused by factors like artifacts and noise. Within the Active Blocks, the Artifact Blocks highlight areas containing artifacts resulting from compression or abrupt distortions. Finally, the Noise Blocks pinpoint Active Blocks containing significant Gaussian noise. Overlaying these blocks on the original image allows for a deeper understanding of the score difference between the real and virtual images [Figure 18].

Some observations of the block's behavior include:

- *Active Blocks:* The totality of real images presents high spatial activity, which reflects the real-life variability of environments, lighting, objects, and sensors. The virtual images lack increased spatial activity in distant regions to the camera, which may be due to the rendering detail of farther areas. Furthermore, areas within the objects and sky also present low spatial variability, which may be explained by the simplified modeling of their physical characteristics.

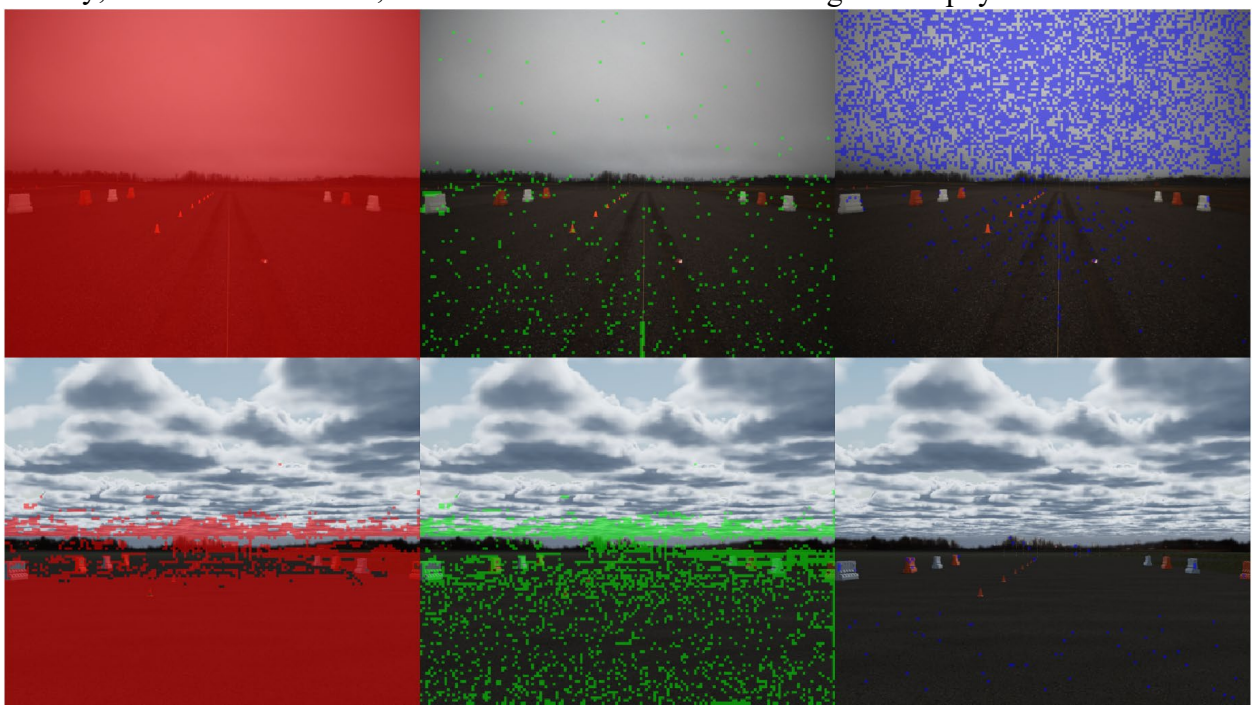

**Figure 18:** Quality for Real Camera (TOP ROW) and Virtual Camera (BOTTOM ROW). Active (RED), Artifacts (GREEN), and Noise (BLUE) Blocks Overlayed.

- *Artifact Blocks:* The virtual images display significantly more artifacts than their real counterparts. This effect clearly explains the reduced quality indicated by the PIQE algorithm. The artifacts tend to concentrate on the horizon transition and areas farther from the image that align with rendering issues, such as large bands of black pixels. Virtual objects also present more artifacts than physical objects.
    - *Noise Blocks:* The real images show considerably more noise than the virtual images, where the noise is mostly constrained to areas of the ground closer to the camera. The largest contrast is within the sky, with increased noise in the real sky and none in the virtual sky.

## 5. LIDAR RESULTS & ANALYSIS

The experiment used a Velodyne VLP-16 LIDAR sensor. The sensor parameters were modeled in dSPACE AURELION. The simulation includes the motion distortion caused by the rotation of the LIDAR emitter.

### *5.1. Analysis Software*

Scripts were developed within MATLAB for the evaluation of the real and virtual LIDAR point clouds. Within MATLAB, the integrated LIDAR Toolbox, Image Toolbox, ROS Toolbox, Navigation Toolbox, and Statistics Toolbox are required to process and analyze the sensors' output.

### *5.2. Establishing "Ground Truth"*

Multiple high-precision survey-grade scans were conducted and subsequently merged to generate a point cloud, which serves as an effective ground truth for comparative analysis with the real and virtual scans captured from the vehicle. The ground truth data was then cropped to a smaller region of interest, encompassing all relevant objects, including cones, barriers, and poles. To eliminate LIDAR returns originating from the ground, a Simple Morphological Filter (SMRF) algorithm was initially applied. However, this algorithm exhibited limitations in completely segregating ground and non-ground points, necessitating some additional manual labeling. To isolate specific objects of interest, a robust clustering algorithm was developed and implemented. Once the objects' point clouds were properly isolated, the point clouds were denoised and fitted with minimal bounding cuboids.

A cuboid is a three-dimensional geometric shape that has six faces, all of which are rectangles. A minimal bounding cuboid refers to the smallest cuboid that can fully enclose the object represented by the point cloud. In the context of this paper, the minimal bounding cuboids optimally fitted to the high-precision survey scans are designated as "Ground Truth Cuboids" [Figure 19].

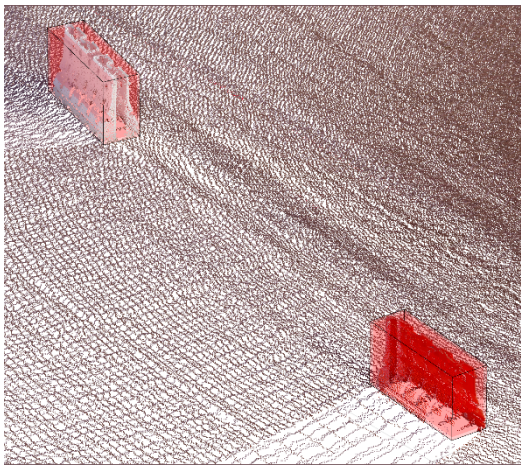

**Figure 19:** Ground Truth Cuboids. Obtained by fitting Minimal Bounding Cuboids to Objects in the Survey-Grade Scans.

### *5.3. Real Vehicle LIDAR: Ground Segmentation & Ego Vehicle*

A limitation of the virtual environment generation is that the LIDAR sensor model only received returns from the objects, as the terrain is not configured to deliver LIDAR returns. Therefore, to proceed with the comparison, the LIDAR returns from the

ground in the real point clouds had to be removed. This procedure involved fitting a plane into the raw point cloud and segmenting non-ground points based on their respective distances from the fitted plane and the angles formed with the plane's normal vector. This basic algorithm (which was the best performing of several explored) still fails to always distinguish correctly ground from non-ground points. In particular, the algorithm struggles with the small height of cones, as they are commonly incorrectly segmented as ground. Finally, returns generated by the ego vehicle were also removed based on their distance from the sensor (since the virtual counterpart did not include a vehicle that would generate returns).

### 5.4. 3D Hu Moments of Inertia

The Hu Moments of Inertia (also applied to the camera image) can be extended to describe general shapes in three-dimensional space. This metric may be applied to binary point clouds, accounting solely for point location, as well as weighted point clouds, considering both point location and intensity. However, it's worth noting that this comparison is not currently viable for the virtual point clouds due to their lack of ground returns. Meaningful comparisons in this regard can only be made once the virtual simulation is enhanced to incorporate ground returns accurately.

### 5.5. LIDAR: Object Detection

The non-ground point cloud obtained from the LIDAR sensor is then segmented with an algorithm tailored for organized point clouds. The algorithm discriminates and clusters points based on the Euclidean distance between points and the angle between the sensor and the neighboring points. Additionally, clusters were further filtered based on their point cloud count and height.

### 5.6. Vehicle LIDAR Scans vs. Ground Truth Cuboids

The non-ground point clouds were compared against the ground truth cuboids. Points that fell inside a cuboid were labeled as true positives while points that failed outside of the cuboids were labeled as false positives. The precision (percentage of true positive to total points) was assessed for each point cloud received [Figure 20]. The virtual LIDAR provides a much higher precision, which is mainly explained by:

- The virtual LIDAR omits ground point returns. Meanwhile, on the real LIDAR data, these ground points are often mislabeled as non-ground points increasing the number of false positives, and thus, decreasing the precision.
- The real LIDAR has a much higher degree of motion distortion compared to the virtual LIDAR. Therefore, after distortion, returns from objects fall outside of the ground truth cuboids.

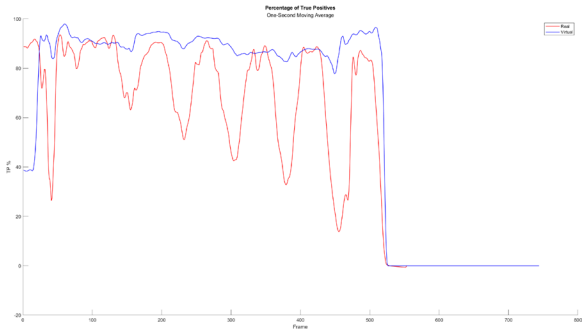


**Figure 20:** Percentage of True Positive in Vehicle LIDAR Scans

### 5.7. Vehicle LIDAR Cuboids

After the LIDAR point cloud has been segmented and clustered, a cuboid is fitted to an individual cluster using an L-shaped detection algorithm. The L-shaped detection is an algorithm optimized for real-time [6], where cuboids are fitted to LIDAR point clouds by iterating through rectangles with different orientations and minimizing the square error.

These bounding cuboids are defined by nine parameters: 3D location, 3D size, and 3D orientation. The parameters are recorded for each object in every point cloud.

**Cuboid Tracking**

Since the portions of an object that generate LIDAR returns depend on the sensor location, a direct comparison of the cuboids generated with the sensor data and the ground truth cuboids is not applicable. However, given that all LIDAR returns emanate from within the bounds of the ground truth bounding cuboid, it is reasonable to anticipate that the centroid of the cuboid generated from LIDAR data also lies within this ground truth geometry. While the virtual data remains closer to the ground truth centroid, the real data diverges in the direction of vehicle movement (motion distortion). Therefore, virtual simulation fails to accurately capture this real-world behavior.

### *5.8. Occupancy Maps*

Validation of generated occupancy maps plays a major role in the comprehensive assessment and fidelity verification of sensor simulation systems. These maps, which depict the spatial distribution of objects or obstacles within an environment, are a fundamental component of autonomous vehicle navigation. As these maps become the inputs for local and global planners, any deviations in the sensor simulation will cascade across the whole simulation system.

Therefore, in this proposed validation, probabilistic (Bayesian Filter) occupancy maps are generated with real and virtual data and compared. To accentuate fine-grained disparities, the occupancy maps used an exceptionally high resolution (10 cells per meter). As seen in Figure 21, it becomes apparent that motion distortion elongates the barriers in the occupancy maps. Furthermore, the absence of noise and distortion in the virtual data enables the occupancy maps to accurately capture the presence of poles, a feat not replicated in the real-world data.

Finally, since some ground points in the real LIDAR are commonly misclassified as objects, the real probabilistic occupancy map includes some areas without being incorrectly marked as occupied.

The F1 score is a metric commonly used in binary classification tasks to evaluate the performance of a classification model. The probabilistic occupancy maps can be binarized using an occupied threshold and compared. In this exercise, the F1 score was 97.23%.

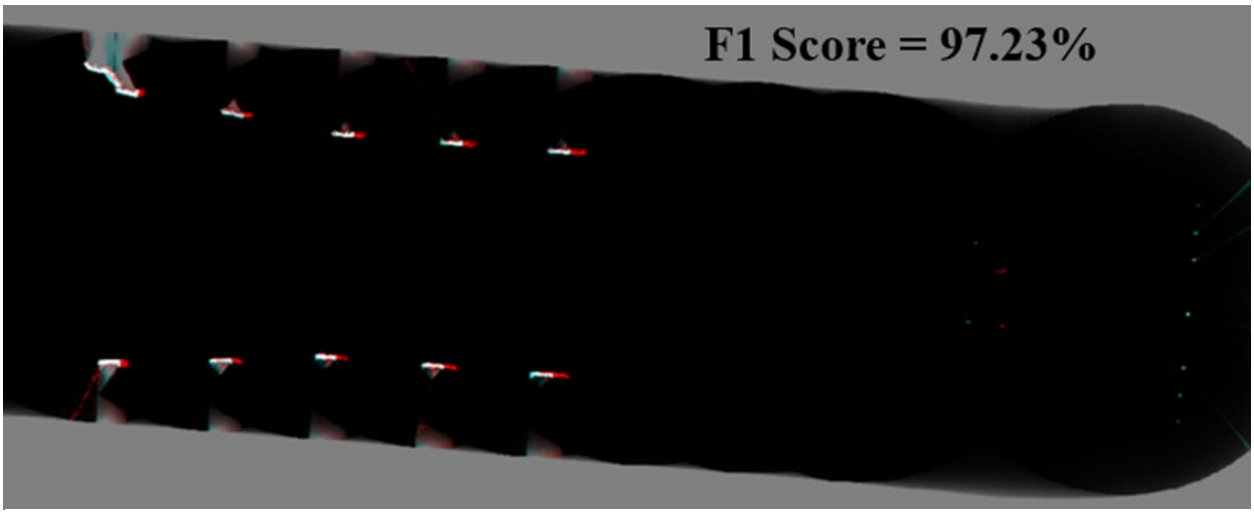


**Figure 21:** Comparison of Occupancy Maps. Black: Free in Both. White: Occupied in Both. Red: Higher Occupancy in Real LIDAR. Cyan: Higher Occupancy in Virtual LIDAR.

## 6. IMPACTS

The metrics above provide a validation framework for exteroceptive sensor models in the context of vehicle autonomy. The development of this toolchain was the main goal of the project; however, these metrics may also be used for:

- *Evaluation of Sensor Models:*
  The metrics provide a quantifiable difference between real-life sensors and their virtual counterparts. Therefore, superior virtual models should provide metrics in closer agreement with their real counterparts.
- *Improvement of Sensor Models:*
  The metrics exposed areas for improvement in the sensor modeling. For instance, if the virtual model consistently underestimates the size of an object compared to the real sensor, this discrepancy can lead to improvements in the virtual model.
- *Metrics for Sensor Selection:*
  The metrics may also be used to evaluate the perception of objects with a given sensor kit. For example, by comparing the performance of different sensor kits in a virtual environment, one can choose the kit that best identifies and tracks objects of interest.
- *Metrics for Perception Algorithms:*
  The object-level validation is a subsystem validation of the sensor model and perception algorithm. Therefore, different perception algorithms should yield different metrics. For instance, two algorithms may differ in their ability to identify objects from sensor data; these differences will be reflected in the metrics.
- *Understand the Gap between Simulation and Reality:*
  Sensor modeling will continue to improve, and metrics from real and virtual sensors will continue to come closer. In the meantime, understanding the differences will help understand in which scenarios the virtual sensor is a good representation of a real sensor. For example, if a virtual sensor struggles to accurately represent a real sensor in foggy conditions, this gap can guide the development of more realistic fog simulations.
- *Trade-Off for Computational Load:*
  In general, higher-fidelity simulations require more computational resources. Nevertheless, a higher fidelity may have diminishing returns if the metrics remain similar. For instance, if a simple sensor model and a complex one yield similar metrics, one might opt for the simpler model to save computational resources.

## 7. FUTURE WORK

- *Evaluation of Additional Sensor Models:*
  While this effort is concentrated on a singular sensor model, it is essential to extend this methodology to multiple models. Of particular interest is a comparison between physics-based models and ideal sensor models, for assessment of the complexity of sensors' models on the simulations' fidelity.
- *Evaluation in Different Weather Conditions:*
  Cameras and their virtual models are very sensitive to changes in weather and environmental lighting conditions. Therefore, assessment of these validation metrics on various conditions would provide a comprehensive evaluation of the camera's model which is independent of scene conditions.
- *State of the Art Segmentation Tools:*
  Several of the metrics developed in this study can be applied to assess the sensor component in isolation. However, the introduction of object-level validation signifies the evaluation of a subsystem comprising the camera sensor and object detection algorithms. While the

development of these algorithms was beyond the scope of this project, it is advisable to apply the same validation methodology in scenarios where the subsystem is composed of state-of-the-art segmentation tools.

- *Virtual Ground Returns:*
  The lack of virtual ground returns is a clear shortcoming in the virtual simulation that should be addressed to properly reflect the challenges posed by real-world LIDAR for autonomous vehicles.
- *Motion Distortion Calibration:*
  While motion distortion is a feature of the physics-based simulation program employed, this effect had more severe consequences in the real world. Further exploration is required to achieve the same behavior.
- *LIDAR Intensity Study:*
  None of the object detection algorithms explored used the intensity values in the point cloud, therefore, the difference between the real and virtual LIDAR point return intensity was not evaluated. Although we proposed a validation (Hu Moment) that would take these values into account, intensity-based algorithms are required to validate the LIDAR intensity simulation at an object level.

## 8. CONCLUSION

A comprehensive validation framework has been developed for exteroceptive sensor models used in simulating diverse military autonomy maneuvers, particularly for off-road operations. While physical exteroceptive sensor performance in on-road applications is well understood, modelling these sensors' performance in an off-road setting is more complex due to the unstructured nature of the environment they operate in. The validation framework allows for an initial comparison of physical sensor outputs collected over multiple tests to the virtual sensor model outputs in a simulation environment.

This validation methodology focuses on objects, recognizing the need for autonomous vehicles to operate effectively in environments where they interact with objects. The NATO AVT-341 effort concluded that an object-level analysis of sensor performance is more effective than a direct pixel-level analysis, as it is more robust to sensor pose estimation errors. Physical experiments conducted at the Keweenaw Research Center (KRC) and replicated at the Southwest Research Institute (SwRI) verified this validation process.

As part of these experiments, the following metrics proved to be most effective for highlighting differences between the real and virtual images:

- Statistics for Bounding Boxes
- Hu Moments of Inertia
- Non-Reference PIQE.

Likewise, the following metrics proved to be most viable for identifying disparities between the real and virtual point clouds:

- Statistics for Cuboids
- Hu Moments of Inertia
- Probabilistic Occupancy Maps

This comprehensive approach to sensor model validation is essential for understanding how an automated vehicle will perform in unstructured environments and is therefore crucial for predicting mission success.

## 9. REFERENCES

[1]Lowe, David G.. "Distinctive Image Features from Scale-Invariant Keypoints." Int. J. Comput. Vision 60 , no. 2 (2004): 91--110. [2] H. Jones, "The Cross of Coronado," Journal of Archaeology, vol 3, issue 5, pages 106-120, 1914.

[2]M. K. Hu, "Visual Pattern Recognition by Moment Invariants", IRE Trans. Info. Theory, vol. IT-8, pp.179–187, 1962.

[3] Mittal, A., A. K. Moorthy, and A. C. Bovik. "No-Reference Image Quality Assessment in the Spatial Domain." IEEE Transactions on Image Processing. Vol. 21, Number 12, December 2012, pp. 4695–4708.

[4] Mittal, A., R. Soundararajan, and A. C. Bovik. "Making a Completely Blind Image Quality Analyzer." IEEE Signal Processing Letters. Vol. 22, Number 3, March 2013, pp. 209–212.

[5] N. Venkatanath, D. Praneeth, Bh. M. Chandrasekhar, S. S. Channappayya, and S. S. Medasani. "Blind Image Quality Evaluation Using Perception Based Features", In Proceedings of the 21st National Conference on Communications (NCC). Piscataway, NJ: IEEE, 2015.

[6] Xiao Zhang, Wenda Xu, Chiyu Dong and John M. Dolan, "Efficient L-Shape Fitting for Vehicle Detection Using Laser Scanners", IEEE Intelligent Vehicles Symposium, June 2018.