Geometric Fidelity Requirements for Meshes in Automotive Lidar Simulation

Christopher Goodin 1,*, Marc N. Moore 1, Daniel W. Carruth 1, Zachary Aspin 2 and John Kaniarz 3

1 Center for Advanced Vehicular Systems, Mississippi State University, Box 5405, Starkville, MS 39762, USA; dwc2@cavs.msstate.edu (D.W.C.)
2 Mobility Systems Branch, US Army Engineer Research and Development Center, Vicksburg, MS 39180, USA; zachary.s.aspin@erdc.dren.mil
3 Ground Vehicle Systems Center, Warren, MI 48397, USA; john.g.kaniarz.civ@army.mil
* Correspondence: cgoodin@cavs.msstate.edu

Abstract: The perception of vegetation is a critical aspect of off-road autonomous navigation, and consequently a critical aspect of the simulation of autonomous ground vehicles (AGVs). Representing vegetation with triangular meshes requires detailed geometric modeling that captures the intricacies of small branches and leaves. In this work, we propose to answer the question, “What degree of geometric fidelity is required to realistically simulate lidar in AGV simulations?” To answer this question, in this work we present an analysis that determines the required geometric fidelity of digital scenes and assets used in the simulation of AGVs. Focusing on vegetation, we use a comparison of the real and simulated perceived distribution of leaf orientation angles in lidar point clouds to determine the number of triangles required to reliably reproduce realistic results. By comparing real lidar scans of vegetation to simulated lidar scans of vegetation with a variety of geometric fidelities, we find that digital tree models (meshes) need to have a minimum triangle density of >1600 triangles per cubic meter in order to accurately reproduce the geometric properties of lidar scans of real vegetation, with a recommended triangle density of 11,000 triangles per cubic meter for best performance. Furthermore, by comparing these experiments to past work investigating the same question for cameras, we develop a general “rule-of-thumb” for vegetation mesh fidelity in AGV sensor simulation.

Keywords: lidar; simulation; autonomy

1. Introduction

Synthetic digital terrains (virtual worlds) may be developed for a variety of purposes ranging from video games to computer animation to medical training. However, one application area for virtual worlds that has seen rapid growth in the last two decades is the physics-based simulation of autonomous ground vehicles (AGVs) [1,2]. The time and expense associated with creating virtual worlds for AGV simulation can be considerable [3], and modeling virtual worlds in the highest possible detail may only increase the cost. Therefore, when it comes to assessing the quality of virtual worlds for AGV simulation, the question often arises, “How good is good enough?” This question may pose some difficulty because the visual cues used by human observers to assess the realism of a virtual world may be different than the features identified by the sensors and software algorithms typically used on AGVs.

AGVs rely on a suite of sensors to perceive the navigation environment, classify terrain, and inform path planning and control algorithms. While the sensor suite may include a variety of sensors like cameras, GPS, and accelerometers, lidar is an important sensor modality for enabling autonomous navigation in intelligent vehicles [4]. Off-road navigation is particularly difficult due to the lack of information that on-road navigation
typically provides, including road networks and lane markings [5]. AGVs navigating in off-road environments are especially reliant on lidar to compensate for the lack of predictable geometric patterns like roads and lanes. Lidar is well known to generate a unique signature when interacting with vegetation due to the apparent “porous” vegetation surface, a feature that has been used to detect vegetation in point clouds for decades [6].

As the number and capability of autonomous vehicles have grown, so too has the simulation capability used to enable autonomy. In fact, a recent review of simulators for AGVs that include perception found that there are at least 11 different popular simulators for AGVs [7]. While many of the simulators in that work handle certain aspects of lidar simulation differently, they all share one feature in common—these simulators use geometric, triangulated models (meshes) of the digital scene as input into the sensor simulations. Given the importance of these meshes to the overall simulation quality, it is important to quantify the requirements for meshes being used in AGV simulation. However, this type of quantitative analysis of required digital mesh fidelity has not been conducted for lidar up until this work. For simulation developers, the answer to this question is critically important. Developing high-fidelity vegetation meshes requires time and money, and optimal digital asset creation will generate meshes with the necessary fidelity, but no more.

In our previous work, we used a systematic approach relying on computer vision to determine mesh requirements for vegetation meshes in camera simulation [8]. In this work, we assess the fidelity requirements for simulating lidar by comparing the distribution of perceived leaf orientation angles between real and simulated point clouds. In the following sections, we present a brief review of the related work (Section 2), a detailed explanation of our analysis method (Section 3), and a presentation of our results (Section 4).

2. Background and Related Work

While simulation has rapidly gained importance for the development and testing of autonomous vehicles, a systematic study of simulation fidelity requirements for AGVs has been noticeably absent. Recent high-fidelity simulators like the Virtual Autonomous Navigation Environment (VANE) have pushed the level of scene fidelity to new extremes [9], but the necessity of this fidelity level has not been evaluated quantitatively. Liu et al. [10] examined the role of vehicle dynamics model fidelity in rollover tests using model predictive control. The role of mesh fidelity in reproducing real results for image classification in camera simulation was also studied recently [8]. However, the question of mesh fidelity requirements for lidar simulation has not been studied prior to this work.

Lidar simulation for autonomous navigation can be achieved through a variety of means. Most approaches are either data-driven (empirical) or physics-based. An early example of data-driven modeling was reported by Browning et al. [11] and Deschaud et al. [12], who used field measurements to construct a voxelized representation of the world with statistical models for how lidar beams interacted with a voxel that were derived from real data. Later, Tallavajhula et al. [13] employed a similar approach using empirically derived terrain primitives rather than voxels. Other data-driven approaches use a mixture of real and simulated data to augment real data sets with targets of specific characteristics [14]. The empirical approach can also be used in a “physical simulation” method by using an anti-reflective screen to mimic targets in an laboratory environment [15].

Physics-based simulation typically relies on triangular meshes and ray-tracing computations, such as early simulations of aerial lidar using “perfect” ray-tracing with empirical errors applied in a final computation stage [16]. The Virtual Autonomous Navigation Environment (VANE) simulator [1,2,17] employs supercomputing to enable a high-fidelity ray-traced approach with realistic errors. Similarly, the MSU Autonomous Vehicle Simulator (MAVS) uses multi-threaded ray-tracing to realistically simulate the interaction of lidar with vegetation [18,19]. Popular simulators like CARLA [20] and Airsim [21] using modern game engines to employ ray-tracing simulations of lidar at lower fidelity, and some high-fidelity simulators also conform to popular output formats like Open Simulation Interface [22].
The relative advantages and disadvantages between the empirical and physics-based methods are clear; the empirical approach may be realistic, but only in the environments and sensors used to create the data, while the physics-based approach is more versatile but potentially less realistic. There has been some study of which features of lidar are important in physics-based simulation. Manivasagam et al. [23] developed a procedure for measuring the so-called “domain gap” of real and simulated lidar and determined that multiple returns and dropped points were important for capturing realistic point clouds in simulation. Similarly, ref. [24] evaluated the important aspects for simulating and Ibeo Lux sensor and found multi-echo effects were influential. However, none of these previous studies systematically evaluated the features in the digital terrain that were necessary to yield realistic results.

While the fidelity of the sensor model has been previously studied and discussed, the terrain and object representation are equally important in determining the final result. Therefore, in this work we take a systematic approach to evaluating the digital terrain fidelity and its influence on lidar simulation realism, specifically for simulating lidar interaction with vegetation. Vegetation is a good choice for evaluating synthetic scene fidelity because of the high variability in real vegetation, including a range of tree shapes and heights, leaf types (i.e., conifer versus deciduous), trunk and leaf colors, and branching complexity [25].

3. Method

In order to evaluate the “realism” of a synthetic digital scene versus the “fidelity” of that scene, it is first necessary to define what the words “realism” and “fidelity” mean. The context and application may influence the interpretation of these terms. For example, a virtual world may appear very realistic in the visible spectrum, but not very realistic in the near-infrared. It is therefore important to define realism in a given context and application. The realism metric that is used in this work is the measurement of leaf inclination angle by terrestrial lidar, as discussed below, while the “fidelity” metric is taken to be the geometric fidelity, as represented by the number of triangles in a vegetation mesh in the synthetic scene.

Recent research has shown that the leaf inclination angles in vegetation have a predictable distribution that is approximately Gaussian when measured by terrestrial lidar [26]. Therefore, in this work we acquired measurements of leaf inclination angles in several dozen experiments and then compared these measurements to the results from simulation. We used a high-resolution lidar point cloud to estimate the orientation distribution of clusters of vegetation, as measured by an approaching vehicle. We compared these distributions to those generated in simulated experiments for varying levels of mesh fidelity. More details on the data collection and analysis method are presented in the following subsections.

3.1. Physical Experiments

The physical experiments were conducted with a Polaris MRZR-D4, as shown in Figure 1 (top). The vehicle was equipped with a sensor suite that included two Ouster OS1-64 lidars—one mounted on the top of the vehicle and another mounted on the front bumper area. The top lidar was one meter above and 1.4 m behind the front lidar. The point cloud data were logged as ROS bag files for later analysis. More information about the vehicle, data acquisition system, and sensor suite can be found in Carruth et al. [27].

This work uses point cloud data from 44 discrete tests in which a short lane with varying types of vegetation was selected and driven through, as shown in Figure 1 (bottom). Tests were conducted at the Center for Advanced Vehicular Systems (CAVS) off-road vehicle proving ground, a 55-acre test range that features a variety of terrain and vegetation cover including grasses, shrubs, and old and new-growth forests. Test lanes were typically 10–20 m long and were approached with a speed of 2–3 m/s.
In order to conduct the analysis of the point cloud data, the point cloud from the top lidar was transformed into the front lidar frame and the two point clouds were combined into a single ROS topic. The merged point cloud was then registered to world coordinates using LIO-SAM [28], a tightly-coupled lidar and inertial odometry method, which is itself based on earlier work on lidar odometry and mapping [29]. Once the point clouds were registered to the world frame, they were combined into a single point cloud with all the other scans. Finally, the combined point cloud was filtered to retain only the points that returned from vegetation in the direct path of the vehicle during the test, ensuring that only vegetation (not ground terrain or other objects) was analyzed this work.

The final merged bag topic was converted to “.pcd” format [30] and analyzed using the Open3D software library (version 0.18.0) [31] in Python. The normal vector of each point was calculated using the estimate_normals method of Open3D, which uses a KDTree [32] with 30 nearest neighbors to estimate the normal at each point. The inverse cosine ($\cos^{-1}$) of the z-component of the each normal—a measure of the orientation angle of the associated surface—was added to a histogram and plotted, as shown in Figure 2. The mean and standard deviation of the histogram were computed for each of the 44 individual measurements performed in the field tests.

As shown in Figure 2, the orientation angles tended to be normally distributed around $\pi/2$ or slightly above, a result also recently measured by Jin et al. [26] for leaf angle measurements of terrestrial lidar. The minimum standard deviation ($\sigma$) in our field measurements was 0.644 radians, while the max was 0.857 radians. The mean inclination angle ($\mu$) measured in this approach was 1.65 radians and the average standard deviation of the inclination angle was 0.757 radians. The characteristics of these distributions were consistent within a tight grouping across all 44 field experiments, indicating that the ($\mu, \sigma$) values potentially provide a good discriminator for distinguishing realistic point clouds from unrealistic ones.

Figure 1. The MRZR test vehicle used in the physical experiments (top) and a close-up of a test lane (bottom).
Figure 2. Example orientation angle distributions from 4 of the 44 field measurements used in this work (a–d) and contrasting simulated oriented angle distributions from 4 of the 27 simulated measurements used in this work, matching the tree meshes shown in Figures 3 and 4 (e–h).
Figure 3. Trees 1–15 of the 27 different geometric models of trees used in this work.
3.2. Simulated Experiments

Simulations were conducted using the MAVS [33] to simulate the lidar sensor as it approached vegetation. While MAVS has been used in a variety of simulation studies in recent years, including system-level analysis of off-road performance [27, 34] and sensor-level studies of lidar interaction with rain [35], it is MAVS’ advanced simulation of lidar-vegetation interaction [19] that makes it a good choice for this research. In particular, it has been shown that MAVS realistically captures the effect of extended objects like vegetation on the lidar return by accounting for the divergence of the lidar beam and oversampling the beam using ray-tracing [18].

Figure 4. Trees 16–27 of the 27 different geometric models of trees used in this work.
Because MAVS has been used to support a variety of field-relevant simulations [36], it was a natural choice for reproducing the field experiments. The simulated experiments were conducted in a similar manner as the field tests; the vehicle was driven toward the vegetation at a low speed while the lidar sensor recorded data and the point clouds were registered and logged to .pcd files for later analysis. The two simulated lidar sensors in MAVS were placed in the same relative position and orientation as the sensors from the real experiments.

Many different tree models (27 total) were used in the simulated experiments, with varying levels of model fidelity as shown in Figures 3 and 4. The tree models were chosen to include the wide variety of tree representations often found in gaming or other simulation applications, ranging from two-dimensional trees to lower-fidelity cartoon trees to highly realistic and geometrically accurate meshes. For this work, the number of triangles in the tree mesh was considered the measure of tree model fidelity.

The orientation angle of the point clouds collected in the simulated experiments were analyzed in an identical way to the real ones using Open3D. The resulting distributions are shown in Figure 2e–h. From these figures it is clear that the shape of some of the simulated distributions is quite different from the real-world measurements like those shown in Figure 2. In particular, the two-dimensional tree (Figure 2e) has a much narrower distribution, as may be expected. The billboard (Figure 2f) and cartoon (Figure 2g) also display orientation distributions that are non-Gaussian. In contrast, Figure 2h shows the output from a simulated distribution that had a more realistic shape, indicating that it was possible for the simulator to produce realistic results for certain meshes.

4. Results

With the real and simulated experiments complete, the analysis was conducted by considering the two datasets separately. There were 44 point clouds from the physical experiments and 27 point clouds from the simulated ones. The normals were calculated for each point cloud using the method described in the previous section, and the mean and standard deviation of the leaf orientation angle (relative to vertical) were calculated for each point cloud. This process allowed any trends in the relationship between mean and standard deviation to be determined, as well as comparison of the overall magnitude between the real and simulated datasets. Figure 5 shows the values of the standard deviation plotted versus the mean for the real and simulated measurements. It is clear that the mean \( \mu \) and standard deviation \( \sigma \) pairs are clustered around a central value for the real measurements (black circles in Figure 5). The center of this cluster is \( \mu_{\text{meas}} = 1.65 \) and \( \sigma_{\text{meas}} = 0.747 \). Therefore, we define inclusion in this cluster as a “realism” metric. Points that fall within the normal range of the cluster of black circles in Figure 5 should be considered realistic, and those that do not fall within the cluster are outside the range of our experimental measurements. Defining the distance parameter \( \epsilon \) as a point from the center of the cluster as

\[
\epsilon(\mu, \sigma) = \sqrt{(\mu - \mu_{\text{meas}})^2 + (\sigma - \sigma_{\text{meas}})^2},
\]

we find that the maximum \( \epsilon \) for the measured data is \( \epsilon_{\text{max}} \text{meas} = 0.11 \). Therefore, we define the realism criteria for a point cloud to be

\[
\epsilon \leq 0.11
\]

and use this criteria to evaluate the different simulated data points (blue x’s in Figure 5) as a function of tree mesh fidelity. This criteria is illustrated by the dashed circle in Figure 5. Points that fall within the circle could be considered realistic, while those outside the circle do not meet the realism criteria.
Figure 5. Standard deviation versus average of the perceived leaf inclination angle for real and simulated datasets.

While Figure 5 shows that there are number of simulated points (blue x’s) that meet the realism criteria, further analysis is required to determine what properties are shared by those figures that meet the criteria, and if it is indeed related to geometric fidelity. To achieve this, Figure 6 shows the $\epsilon$ value for each of the simulated point cloud distributions versus the logarithm of the number of triangles in the mesh used to make the distribution. The general trend shows that the $\epsilon$ decreases as the number of triangles increases. The dotted black line shows the value of $\epsilon_{\text{max}} = 0.11$. The solid black line is a fit that yields the observed mathematical relationship between the offset $\epsilon$ and the number of triangles.

$$\epsilon = 0.269 - 0.0121 \ln(n_{\text{tri}})$$  

(3)

where $n_{\text{tri}}$ is the number of triangles. This equation fits the data, which are noisy, with a reasonable effectiveness, having an $R^2=0.635$. By substituting the value for $\epsilon_{\text{max}}$ into Equation (3), the minimum required number of triangles to ensure a tree mesh meets the realism threshold defined in Equation (2) (the ordinate of the intersection point between the solid and dotted black lines in Figure 6) can be determined.

Figure 6. $\epsilon$ (deviation from average real measurement) of the simulated tests versus the logarithm of the number of triangles in the mesh.
Using this approach, we find that the minimum number of triangles in a mesh required to ensure that lidar scans of the mesh will have leaf angles distributed like real data to within the realism threshold is \( n_{\text{tri,min}} = 491 \, \text{K} \). This number (491 K triangles) is the intersection of the solid black line with the dashed horizontal line in Figure 6. Stated another way, the black line fits the measured deviation from “real” of each simulated data-point, and only values that fall below the maximum deviation of the actual real data points (the dashed line) are considered realistic by this standard. However, as noted, the data in Figure 6 are noisy, and there were some meshes with considerably fewer triangles that still fell beneath the threshold. In fact, the minimum number of triangles for a mesh that still met the requirement was 72 K triangles. In other words, the mesh that was the farthest to the left in Figure 6 and still beneath the dotted line threshold had 72 K triangles. The average number of triangles in the meshes that meet the minimum requirement (excluding the mesh with 3.5 million triangles as an outlier) is \( \mu_n = 262,435 \) and the standard deviation is \( \sigma_n = 114,732 \). So the value of 72 K triangles is 1.65 standard deviations from the mean, implying a confidence of about 90%. Therefore, we adopt two different thresholds: the minimum recommended number of triangles is \( n_{\text{rec}} > 491 \, \text{K} \), while the minimum required number of triangles is \( n_{\text{req}} > 72 \, \text{K} \).

All the tree meshes in the simulations were scaled to a volume of about 45 m\(^3\) in our analysis. Therefore, in order to achieve a scale-invariant recommendation, we also consider the triangle number density as the realism metric. When taking the volume into account, we find that \( n_{\text{rec}} > 11 \, \text{K} \) and \( n_{\text{req}} > 1600 \).

5. Discussion

There are many scenarios which may be explored in relation to simulation fidelity requirements. These include varying sensor fidelity, vehicle speed and properties, terrain properties, and sensor type. Our previous work studied fidelity requirements for camera simulation [8]. This work found that for cameras, tree meshes should have \( \frac{n_{\text{cam}}}{m^3} > 1000 \). The method of the camera analysis was similar to the one presented here—a set of varying-fidelity tree meshes was used to generate images, which were then classified by a machine learning algorithm trained on real data. In the camera analysis, we used three different rendering modes and two different simulators (MAVS and UE4) and found that the rendering engine needed to solve the full global illumination problem in order to produce realistic results. In this paper, we find that the \( \frac{n_{\text{rec}}}{m^3} > 1600 \) for lidar, comparable to the value of \( \frac{n_{\text{cam}}}{m^3} > 1000 \) for cameras, but the recommended value of \( \frac{n_{\text{req}}}{m^3} > 11 \) K is much higher. The most obvious explanation for this difference is the sampling rate of the two sensors.

In the camera experiments, the final images typically had about 100 K pixels “on-target”; that is, about 100,000 pixels that intercepted light reflected from the tree. Because the same trees were used in the previous camera experiment and this one, in terms of pixel density, the cameras resulted in about \( \frac{2000 \, \text{pixels}}{m^3} \) on the tree. In contrast, the lidar scans had as many as 750 K returns from the trees in these experiments because multiple scans were merged. This gives a sample density of \( \frac{17,000 \, \text{points}}{m^3} \) on the tree. Comparing the sample density for the lidar and camera to the required triangle density, we see that for both sensors, the triangle density required to accurately reproduce real-world results is about 0.5–0.65 times the sample density. Therefore, we propose the following generalized “rule-of-thumb” for vegetation mesh geometry.

\[
\frac{n_{\text{tri}}}{m^3} \approx \left( \frac{5}{8} \right) \frac{\text{num sensor samples}}{m^3}
\]

This result can be used to estimate fidelity requirements for a variety of sensor simulation applications for both lidar and cameras.
6. Conclusions and Future Work

In this work, we determined the fidelity requirements for simulating terrestrial lidar in off-road navigation, with a focus on simulating lidar interacting with vegetation. By comparing the perceived distribution of angle orientations in real and simulated data sets, we determined that a minimum triangle density of \( n_{\text{req}}^\text{tri} > 1600 \) is required for accurately simulating lidar, but a density of \( n_{\text{rec}}^\text{tri} > 11 \) K is recommended for best results. By comparing this to our previous work on mesh fidelity requirements for camera simulation, we developed the general rule of thumb that \( n_{\text{tri}}^\text{approx} \approx \frac{5}{8} \text{num sensor samples} \) for either sensor modality. These findings should be used to inform scene development for simulating autonomous ground vehicles in future work, allowing simulation developers to create realistic digital scenes without over-investing time and money in digital asset creation. Future work in this area will focus on how vehicle speed may influence sensor and scene fidelity requirements. Additional scenarios that may be considered include the consideration of different object types (other than lidar) different and more complex scenes, and evaluating simulation fidelity in the context of sensor fusion.

Author Contributions: Conceptualization, C.G., Z.A. and J.K.; methodology, C.G.; software, C.G.; validation, M.N.M.; formal analysis, C.G.; investigation, C.G.; resources, C.G. and D.W.C.; data curation, C.G. and M.N.M.; writing—original draft preparation, C.G.; writing—review and editing, C.G., Z.A. and J.K.; visualization, C.G.; supervision, D.W.C.; project administration, C.G. and D.W.C.; funding acquisition, C.G. and D.W.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was performed using funding from the US Department of Defense (DOD) High-Performance Computing Modernization Program (HPCMP) under contract W912HZ-22-C-0004. DISTRIBUTION A. Approved for public release; distribution unlimited.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study may be available on request from the corresponding author due to funding from the US Department of Defense.

Conflicts of Interest: The authors declare no conflicts of interest.

References


35. Goodin, C.; Carruth, D.; Doude, M.; Hudson, C. Predicting the Influence of Rain on LIDAR in ADAS. Electronics 2019, 8, 89. [CrossRef]

36. Meadows, W. Multi-LiDAR Placement, Calibration, and Co-Registration for Off-Road Autonomous Vehicle Operation; Mississippi State University: Starkville, MS, USA, 2019.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.