ERDC/GSL TR-20-32

Geotechnical and Structures Laboratory

US Army Corps
of Engineers®
Engineer Research and
Development Center

ERDC
ENGINEER RESEARCH & DEVELOPMENT CENTER

Military Engineering

Sensor and Environment Physics in the Virtual Autonomous Navigation Environment (VANE)

Justin T. Carrillo, Christopher T. Goodin, and Juan Fernandez

August 2020

Military Engineering

ERDC/GSL TR-20-32
August 2020

Sensor and Environment Physics in the Virtual Autonomous Navigation Environment (VANE)

Justin T. Carrillo, Christopher T. Goodin, and Juan Fernandez

Geotechnical and Structures Laboratory
U.S. Army Engineer Research and Development Center
3909 Halls Ferry Road
Vicksburg, MS 39180-6199

Final report

Prepared for Engineer Research and Development Center
3909 Halls Ferry Road
Vicksburg, MS 39180-6199

Under Simulation Tools for Autonomous Ground Resupply
Project Number 471937 Task #2 Sensor and Environment Physics

Abstract

This report documents the physics models that are implemented in the Virtual Autonomous Navigation Environment (VANE), a sensor simulator that uses physics-based ray tracing to simulate common robotic sensors such as cameras, LiDAR, GPS, and automotive RADAR. The report will provide information about the underlying assumptions and implementation details regarding the physics models used in VANE simulations. These include surface reflectance and texture models, atmospheric models, weather effects, and sensor properties.

The purpose of this report is to provide information for VANE users, developers, and analysts who would like to use the VANE for sensor simulations.

DISCLAIMER: The contents of this report are not to be used for advertising, publication, or promotional purposes. Citation of trade names does not constitute an official endorsement or approval of the use of such commercial products. All product names and trademarks cited are the property of their respective owners. The findings of this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

DESTROY THIS REPORT WHEN NO LONGER NEEDED. DO NOT RETURN IT TO THE ORIGINATOR.

Contents

Figures and Tables

Figures

Tables

Preface

This study was conducted for the Engineer Research and Development Center (ERDC) under the program title “Simulation Tools for Autonomous Ground Resupply.” The technical monitor was Mr. David P. McInnis.

The work was performed by the Mobility Systems Branch (MSB) of the Engineering Systems and Materials Division (ESMD), U.S. Army Engineer Research and Development Center, Geotechnical and Structures Laboratory (ERDC-GSL). At the time of publication, Dr. Phillip J. Durst was Chief, MSB; Mr. Justin Strickler was Chief, ESMD; and Mr. Nicholas Boone was the Technical Director for Military Engineering. The Deputy Director of ERDC-GSL was Mr. Charles W. Ertle III, and the Director was Mr. Bartley P. Durst.

COL Teresa A. Schlosser was the Commander of ERDC, and Dr. David W. Pittman was the Director.

1 Introduction

The purpose of this report is to provide a detailed description of the physics models implemented in the Virtual Autonomous Navigation Environment (VANE) software for simulating sensors and the environmental impact on sensors, including any simplifying assumptions or implementation details. The intended audience for this report includes VANE users that want to understand the limitations imposed by the physics models, VANE developers who are adding new capability to the code, and code testers who need to document VANE capabilities for the verification and validation process.

1.1 Background and approach

The purpose of the VANE is to realistically simulate sensors in complex outdoor environments. It is targeted to sensors commonly used on medium and large, wheeled and tracked robotic systems. These sensors include LiDAR, GPS, color and near-infrared cameras, and automotive radar. A typical VANE analysis might simulate a 3-10 min autonomous mission featuring specific, sensor-dependent autonomous functionalities such as obstacle detection and avoidance (ODOA), terrain classification, or map generation (Goodin et al. 2017). A simulation like this may generate thousands of images and millions of LiDAR points. Therefore, while VANE is a physics-based ray-tracing simulator, great effort has been given to allowing the code to run in a reasonable amount of time, i.e., reasonable as defined in this case as giving the user the ability to complete a simulation in one day. Therefore, simplifying assumptions and approximations are made in many of the simulation models. Additionally, empirical and reduced order models are used for certain aspects of the simulation when appropriate.

The VANE is not a tool for analyzing new sensor concepts with novel operating principles. If the new sensor violates the assumptions implicit to the existing VANE models, the user cannot expect the result to be correct. However, VANE can be used to compare sensor models of similar operating principles. For example, if a new LiDAR sensor with improved range or spatial resolution becomes available, it can quickly be added to VANE, because it already accounts for these properties.

1.2 Objective

There are several free, commercially available, and Department of Defense (DoD) specific robotic simulation tools currently available. Recently, within academia and the DoD, two tools have emerged as favorites in industry. The first is Gazebo (Koenig and Howard 2004), a tool that was described by Michal and Etzkorn (2011) as a Robotic Development Environment (RDE). The second is the Autonomous Navigation Virtual Environment Laboratory (ANVEL) (Durst et al. 2012), an unmanned ground vehicle (UGV) simulator that uses video-game technology to provide an interactive simulation environment.

The most obvious difference between VANE and the two tools mentioned above is the level of detail and fidelity in the description of the environment and the impact of the environment on sensor performance. Gazebo is primarily for developers that want to debug their systems, not assess the impact of the environment on that systems performance. At an intermediate level, the ANVEL can be used to evaluate the performance of a UGV in a mission in a relevant environment, but the level of detail in the environmental physics is limited to what can be simulated in real-time on typical laptop computers. In contrast, VANE uses geometrically, highly attributed terrains to perform fully spectral simulations of the sensor-terrain and vehicle-terrain interactions.

VANE scenes and environments can be attributed with full spectral reflectance properties and refractive indices, which allows for a wide variety of sensors to be simulated realistically. For example, characteristics such as the normalized differential vegetation index (NDVI), a ratio of brightness in the red and near-infrared bands that is used by many autonomy algorithms to distinguish vegetation from other non-navigable obstacles, cannot be simulated with tools such as ANVEL and Gazebo that have no spectral characteristics in the material properties. However, this can easily be simulated with the VANE, allowing autonomy developers to fine-tune perception algorithms in simulation in a way that is not possible without the VANE.

Additionally, the physics of LiDAR sensor interaction with obscurants like dust and rain is realistically simulated in the VANE, in contrast to Gazebo and ANVEL. These features make VANE a valuable tool that complements real-time simulators.

2 Camera Simulation

The VANE can be used to simulate digital cameras in the optical and near-infrared wavelengths that use charge-coupled device (CCD) or complementary metal-oxide semiconductor (CMOS) sensors at the focal plane array. A VANE camera is defined by the focal length, size of the focal plane array, and number of pixels of the camera. In the simulation, the irradiance on each pixel is calculated by tracing one or more rays from the focal point, through a pixel, and into the scene. If an intersection is found, the lighting and reflectance models in the VANE are used to calculate the reflected light in the direction of the pixel. If not, the sky model is used to calculate the amount of ambient energy on the pixel; this process is performed for each pixel. The VANE simulation is parallelized by having each pixel calculation performed by a different thread or processor.

Non-ideal lensing systems can also be simulated in the VANE as well as electrical properties such as the gain and gamma compression of the sensor.

2.1 Parameterizing the lens system

Real cameras exhibit some degree of distortion. For systems in the optical and near-infrared wavelengths, the properties of the lensing system can be modeled using radial and tangential distortion coefficients. In order to model a camera in the VANE, these parameters must first be measured for the system to be simulated. The MATLAB camera calibration toolbox developed by Bouget (2015) was used, which was in turn based on an early paper by Brown (1966). The toolbox measures the focal length, principal point, skew, and five radial and tangential distortion parameters. The VANE camera model parameters are listed in Table 2-1.

Table 2-1. Camera optical parameters.

Parameter	Description
k1	1st order radial distortion coefficient
k2	2nd order radial distortion coefficient
k3	3rd order radial distortion coefficient
p1	Horizontal tangential distortion coefficient
p2	Vertical tangential distortion coefficient
cc1	Horizontal principal point [pixels]
cc2	Vertical principal point [pixels]
fc1	Horizontal focal length [pixels]
fc2	Vertical focal length [pixels]
α_c	Skew coefficient
f	Focal length of the ideal system [m]
N1	Number of horizontal pixels
N2	Number of vertical pixels
R1	Horizontal dimension of the pixel array [m]
R2	Vertical dimension of the pixel array [m]

For each pixel (u,v), a ray is traced from the origin at the focal point, **fp**, to the location on the plane. In the 2-D coordinate system of the pixel plane, this point is

$$x^p = \begin{bmatrix} x_1^p \\ x_2^p \end{bmatrix} \tag{2.1}$$

where

$$x_1^p = \left[u - \frac{N_1}{2}\right]\frac{R_1}{N_1} \tag{2.2}$$

$$x_2^p = \left[v - \frac{N_2}{2}\right]\frac{R_2}{N_2} \tag{2.3}$$

The re-centered coordinate, x^d is given by□

$$x^d = \begin{bmatrix} (x_1^p - cc_1)/fc_1 \\ (x_2^p - cc_2)/fc_2 \end{bmatrix} - \alpha_c \begin{bmatrix} (x_2^p - cc_2)/fc_2 \\ 0 \end{bmatrix} \tag{2.4}$$

The position of the distorted pixel, **x**, is then calculated using the iterative method presented in Heikkila and Silven (1997).

In the 3-D camera coordinate system where the **z** is perpendicular to the pixel plane, the ray direction, **d**, is then given by

$$\boldsymbol{d} = [x_1, x_2, f]/\sqrt{x_1^2 + x_2^2 + f^2} \tag{2.5}$$

and the updated ray origin, **o**',

$$\boldsymbol{o}' = [fp_1 + cc_1, fp_2 + cc_2, 0] \tag{2.6}$$

An ideal pinhole camera can be created by setting k_1=k_2=k_3=p_2=p_3=cc_1=cc_2=0 and fc_1=fc_2=f. Otherwise, the parameters for a particular camera can be measured with the MATLAB camera calibration toolbox (Bouget 2015).

2.2 Bidirectional reflectance distribution function

Reflectance is modeled in the VANE using the bidirectional reflectance distribution function (BRDF) of the reflecting surface. The VANE has two models for BRDF, i.e., a modified Phong model (Goodin 2013) and an implementation of the He-Torrance-Sillion-Greenberg (HTSG) model (He et al. 1991) (Peters and Goodin 2015).

The modified-Phong model is a simple 3-parameter model for diffuse and specular reflectance that is wavelength-independent and is relatively fast, computationally. This is the main BRDF model used in the VANE. Conversely, the HTSG model is a complex, computationally expensive, wavelength-dependent model that calculates the polarization state of the reflected light. The HTSG model is seldom used in VANE because the input parameters are difficult to determine. This model is described in detail in Peters and Goodin (2015) for interested readers.

The modified-Phong model is described in detail in Goodin (2013) and is briefly summarized in this document to show that it satisfies the requirements defined in Lewis (1993) for physically plausible shaders, namely that they are energy-conserving and reciprocal. For a surface with a normal **N** being illuminated by irradiance **E** from normalized incident direction $\mathbf{v}_i$, the reflected irradiance in the normalized direction $\mathbf{v}_r$ is given by the BRDF function.

$$f_r(\boldsymbol{N},\boldsymbol{v}_i,\boldsymbol{v}_r) = \frac{k_d}{\pi} + k_s\frac{n_s+2}{2\pi}\max(\boldsymbol{v_s}\cdot\boldsymbol{v}_r,0)^{n_s} \tag{2.7}$$

where $\boldsymbol{v}_s$ is the specular reflection direction given by□

$$\boldsymbol{v}_s = 2(\boldsymbol{v}_i\cdot\boldsymbol{N})\boldsymbol{N} - \boldsymbol{v}_i \tag{2.8}$$

In these equations, k_d is the diffuse reflectance coefficient, k_s is the specular reflectance coefficient, and n_s is the specular reflectance exponent. For the BRDF to remain physically plausible, the following conditions on the model parameters must be satisfied. Figure 2-1 shows an example of three different BRDF parameterizations within the VANE.

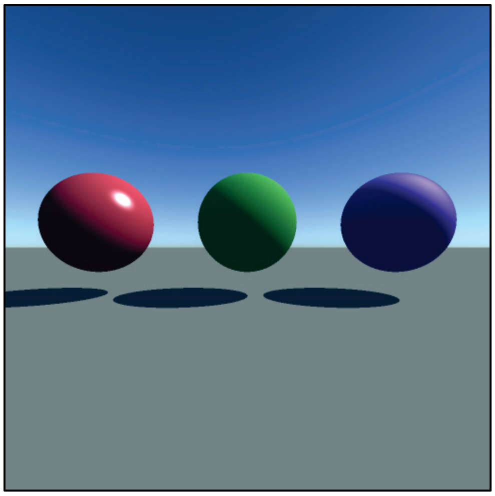
$$0 \le \mathrm{k_d} \le 1 \tag{2.9}$$

$$0 \le \mathrm{k_s} \le 1 \tag{2.10}$$

$$0 \le \mathrm{k_s} + k_d \le 1 \tag{2.11}$$

$$\mathrm{n_s} \ge 1 \tag{2.12}$$

Figure 2-1. Example of three different BRDF parameterizations in VANE.



2.3 Lighting model

For a given wavelength, λ, in the visual (VIS) / near-infrared (NIR) spectrum where emission can be neglected, the time-independent rendering equation is (Wallace et al. 1987)

$$L_r(x, v_r) = \int_\Omega f_r(N, v_i, v_r) L_i(x, v_i) \cos\theta_i \, d\omega_i \qquad (2.13)$$

where **x** is a point in space, **N** is the surface normal at **x**, and $\mathbf{v}_i$ and $\mathbf{v}_r$ are the incident and reflected light directions, respectively. The BRDF, $f_r(\mathbf{N}, \mathbf{v}_i, \mathbf{v}_r)$, which can take several forms, is already defined in Equation 2.7. L_r is the intensity of the reflected light and L_i is the intensity of incident light. In scenes with multiple light sources, such as streetlights or lamps, the L_i can be quite complex. However, in VANE L_i is simplified by assuming rendering will take place in outdoor scenes where the light source term can be split into two parts - a direct term from the sunlight and an ambient turn from the skylight.

$$L_i(x, v_i) = L_{\mathrm{sun}} \delta(\omega_{\mathrm{sun}}) + L_{\mathrm{sky}} \qquad (2.14)$$

where δ (ω_{sun}) is the Dirac-delta function in the direction of the sun and L_{sky} is assumed that it is uniform in all directions. It should be noted that this assumption is of course not strictly correct, but nevertheless provides visually realistic results and greatly simplifies the rendering equation to

$$L_r(x, v_r) = L_{\mathrm{sun}} f_r(N, v_i, v_r) + L_{\mathrm{sky}} \int_\Omega f_r(N, v_i, v_r) \cos\theta_i \, d\omega_i \qquad (2.15)$$

The rightmost term in this equation is known as the hemispherical reflectance distribution function (HDRF). Following Equation 2.7 if specular reflection of the diffuse sky is neglected, then the integral becomes

$$\int_\Omega \frac{k_d}{\pi} \cos\theta_i \, d\omega_i \qquad (2.16)$$

With this result, the final equation for the light radiance from a surface in VANE is

$$L_r(x, v_r) = L_{\mathrm{sun}} f_r(N, v_i, v_r) + k_d L_{sky} \qquad (2.17)$$

The radiance actually reaching a pixel is modified by in-scattering and absorption processes that take place between the camera and the surface.

$$L_{\mathrm{pixel}} = \left(\frac{I}{I_0}\right) L_r(x, v_r) + L_{\mathrm{inscattered}} \qquad (2.18)$$

where I/I_0 is the absorption factor, varying between 0 and 1, that quantifies the fraction of light that is absorbed by the transmitting medium, which in this case is the air. Dust and rain can also have scattering effects and will be discussed in Chapters 6 and 8.

The following sections will outline the models used in the VANE to calculate $L_{inscattered}$, I/I_0, L_{sun}, and L_{sky}. The underlying models used to solve Equation 2.18 are different for the visible/near-infrared bands (320-720 nm) and the longer wavelength infrared bands (>720 nm). This is due to the different physical processes that dominate the spectrum at those wavelengths. Namely, in the visible and NIR wavelengths, the spectrum is dominated by surface reflections, whereas thermal emission becomes increasingly important for longer wavelengths. In the following sections, the models for the VIS/NIR and IR will be discussed.

2.3.1 Visible and near-infrared

The direct and diffuse lighting for the visible and near-infrared region is taken from the analytical model developed by Hosek and Wilkie (2012). The Hosek-Wilkie model is an analytical fit to a lookup-table that was generated by performing physics-based scattering simulations of the atmosphere. The input of the model is the turbidity, T, of the atmosphere, a single parameter quantifying the aerosol content in the atmosphere. Quoting from Hosek and Wilkie (2012, 2013)

- T = 2 yields a very clear, Arctic-like sky,
- T = 3 a clear sky in a temperate climate,
- T = 6 a sky on a warm, moist day,
- T = 10 a slightly hazy day,
- T >= 50 represent dense fog.

A C implementation of the model is available from Hosek and Wilkie (2013), which is implemented directly in the VANE for optical wavelengths. The model's application programming interface (API) provides functions to get the solar radiance, which is the definition of L_{sun} from Equation 2.17 The diffuse color, L_{sky}, is calculated by averaging the sky-color over the hemisphere for the current sky condition (given the turbidity, time of day, etc.) with the Hosek-Wilkie sky model. The sky dome is sampled in 1-degree intervals over the zenith and azimuth directions.

While the Hosek-Wilkie model gives realistic results for the spectral intensities of the sky-dome, additional modeling is needed for rays that intersect an object to account for the influence of in-scattered atmospheric radiation between the camera and the object. In order to calculate this in VANE, the same turbidity parameter is used as in the Hosek-Wilkie model.

For light reflected from an object at distance, d, from the camera, the intensity of received light at the camera will be reduced according to the transmission factor

$$I = I_0 e^{-\beta d} \tag{2.19}$$

where β is the volume scattering coefficient with units of inverse meters (1/m). According to Platt et al. (2007), scattering in the lower atmosphere can be separated into two main processes, i.e., Rayleigh scattering from air molecules and Mie scattering from larger suspended aerosols. Equation 2.19 can then be rewritten as

$$I = I_0 e^{-(\beta_m + \beta_r)d} \tag{2.20}$$

where β_m and β_r are the Mie and Rayleigh scattering coefficients, respectively. The Mie coefficient can be estimated in the optical wavelengths by Platt et al. (2007) as

$$\beta_m = \beta_{m0} \lambda_{\text{micro}}^{-\alpha} \tag{2.21}$$

where λ is the wavelength of the light in μm and α is Angstrom's turbidity constant. Although α can theoretically vary from 0.5-2.5, it is typical to let $\alpha = 1.3$ (Preetham et al. 1999). The value of β_{mo} is related to the turbidity T by Preetham et al. (1999) as

$$\beta_{m0} = 0.04608T - 0.04856 \tag{2.22}$$

The Rayleigh scattering coefficient for optical wavelengths can be estimated by Preetham et al. (1999) as

$$\beta_r = \frac{8\pi(n_{\text{air}}^2 - 1)^2}{3N\lambda^4} \frac{6+3p_n}{6-7p_n} \tag{2.23}$$

where n_{air}=1.0003 is the refractive index of air in the visible wavelengths, $N = 2.54 * 10^{25}$ is the number of air molecules per cubic meter, $p_n = 0.035$

is the depolarization factor for air, and λ is the wavelength in meters. Taken together, Equations 2.14-2.17 are used to calculate the I/I_0 factor from Equation 2.12.

2.3.2 Mid-wave to long-wave infrared

Direct and diffuse irradiance for the infrared spectrum is calculated using the Bird simple spectral model, which is available through the National Renewable Energy Laboratory (NREL) (Rymes 2004). The inputs and outputs of the model are listed in Table 2-2.

Table 2-2. Sky model parameters for infrared wavelengths.

Inputs		
Input	**Description**	**Default**
Latitude, longitude	Decimal degrees North and East	(33.65, -84.43)
Altitude	Meters above sea level	200.0
Time zone	Decimal East	-5.0
Year, month, day	Date	1 Jan 2013
Hour, minute, second	Time	12:15:25
Temperature, Pressure	^{o}C, millibars	27.0, 1006.0
Ground reflectance	Albedo of the ground	0.5
α	Power on the angstrom turbidity coefficient	0.5
τ_{500}	Aerosol optical depth at 0.5 µm	0.2
Water Vapor	Precipitable water vapor, cm	1.36
Outputs		
Output	**Description**	**Equations**
Direct	Direct irradiance in W/m2/µm	L_{sun} in Eq. 4.11
Diffuse	Diffuse irradiance in W/m2/µm	L_{sky} in Eq 4.11

The output of the NREL model is used in Equation 2.11 to calculate the L_{sun} and L_{sky} values. Equations 2.13-2.17 are still used to calculate the in-scattered radiation ($L_{inscattered}$ in Equation 2.12), although the factor of $\frac{1}{\lambda^4}$ in these equations means there is relatively little in-scattered radiation in the infrared wavelengths.

3 GPS Simulation

The GPS simulation in VANE is outlined in detail in Durst and Goodin (2012). Much of that information is repeated in this work for convenience to the reader, as well as details on the implementation of the model in the VANE.

3.1 Ray tracing

Coordinates of the satellite are broadcast in each satellite's position signal in an earth-centered, earth-fixed (ECEF) system that uses the World Geodetic System 1984 (WGS84) referenced system. To perform the ray tracing in VANE, the satellite coordinates in ECEF (p_{ECEF}) are converted to a local East-North-Up (ENU) coordinate system (p_{ENU}) with origin at O with the following equations

$$\delta = p_{ECEF} - O \tag{3.1}$$

$$\lambda = \mathrm{atan2}(O_y, O_x) \tag{3.2}$$

$$\varphi = \mathrm{atan2}(O_z, \sqrt{O_x^2 + O_y^2}) \tag{3.3}$$

$$\begin{gathered} p_{\mathrm{ENU}}^{x} = -\delta_x \sin\lambda + \delta_y \cos\lambda \\ p_{\mathrm{ENU}}^{y} = -\delta_x \sin\varphi \cos\lambda - \delta_y \sin\varphi \sin\lambda + \delta_z \cos\varphi \\ p_{\mathrm{ENU}}^{z} = \delta_x \cos\varphi \cos\lambda + \delta_y \cos\varphi \sin\lambda + \delta_z \sin\varphi \end{gathered} \tag{3.4}$$

For each satellite, the viewing vector, **R**, is given by

$$\boldsymbol{R} = p_{\mathrm{ENU}} - O \tag{3.5}$$

and the range between satellite and receiver is given by

$$R = |\boldsymbol{R}| \tag{3.6}$$

The viewing angle, ε, is then given by

$$\epsilon = \mathrm{atan}(R_z / \sqrt{R_x^2 + R_y^2} \tag{3.7}$$

In the first phase of the ray-tracing segment of the simulation, a ray is traced from the receiver to the satellite. If any object in the scene is between the satellite and the receiver, then a direct signal is not received in the simulation. In the second phase of the ray-tracing, 100 rays are traced in even angular spacing in the hemisphere above the receiver. For each ray that intersects an object in the scene (that is closer than 45 m), a secondary ray is traced from the object to the satellite. If there is no object between the satellite and the secondary ray, then a multipath signal is received with a multipath error, dR_{multi}, added that is equal to the distance between the receiver and the intersected object.

3.2 Atmospheric error models

The ionosphere range error is modeled with the Montenbruck mapping function (Montenbruck and Garcia-Fernandez 2005)

$$dR_{\text{iono}} = \frac{2.037}{\sqrt{\sin^2\epsilon + 0.076} + \sin\epsilon} \tag{3.8}$$

where ε is the viewing angle between the receiver and the satellite calculated in Equation 3.7. The upper troposphere errors depend on the total vertical electron count, z_{200}, and the latitude, θ, in decimal degrees. The error is modeled using a Neill mapping function (Niell 2000) with the following equations:

$$dR_{\text{trop,wet}} = z_{\text{eq}} \frac{1 + \frac{a}{1 + \frac{b}{1 + c}}}{\sin\epsilon + \frac{a}{\sin\epsilon + \frac{b}{\sin\epsilon + c}}} \tag{3.9}$$

$$a = a_{lat} + d_{lat}(z_{200} - z_{\text{ref}}) \tag{3.10}$$

$$b = b_0^m \tag{3.11}$$

$$c = c_0^m + c_1^m(\theta - l_0^c) \tag{3.12}$$

$$a_{lat} = a_0 + a_1 \cos 2(\theta - l_0^a) \tag{3.13}$$

$$d_{lat} = d_0 + d_1 \cos 2(\theta - l_0^d) \tag{3.14}$$

$$z_{\text{ref}} = z_0 + z_1 \cos 2(\theta - l_0^z) \tag{3.15}$$

The constants used in these calculations are listed in Table 3-1.

Table 3-1. Constant numeric values.

Constant	Value
z_{eq}	0.1
a_0	0.00124
a_1	4.0E-5
l_0^a	2.0
d_0	7.4E-8
d_1	-1.6E-8
l_0^d	0.0
z_0	11836.0
z_1	619
l_0^z	3.0
l_0^c	0.0
b_0^m	0.002905
c_0^m	0.0634
c_1^m	0.0014

The lower troposphere errors are modeled using the equations from Kim et al. (2009)

$$dR_{\text{trop,dry}} = \frac{2.2779P}{1-0.00266\cos 2\theta-0.0028h} \tag{3.16}$$

where P is the air pressure at the receiver in kPa, θ is the altitude in decimal degrees, and h is the altitude above sea level in meters. Ephemeris errors are modeled using the work of Hwang and Hwang (2002)

$$dR_{\text{eph}} = \sqrt{(1 + 0.5\sin\theta)\left(10^{\frac{2R_{\text{ref}}}{R}}\right) + c^2 \times 10^{-18} + 0.0918} \tag{3.17}$$

where θ is the latitude in decimal degrees, R is the range in meters, $R_{ref} = 2.655E^7$, and c is the speed of light in m/s. Finally, all the errors, including multipath, are summed to give the total pseudorange error.

$$R' = R + dR_{\text{multi}} + dR_{\text{iono}} + dR_{\text{trop,wet}} + dR_{\text{trop,dry}} + dR_{\text{eph}} \tag{3.18}$$

3.3 Trilateration

To start the trilateration algorithm, an initial guess for the receiver position (p_r) is made. In the VANE algorithm, the position is always initialized to zero

$$p_r = [0,0,0] \tag{3.19}$$

For each satellite (1-n) that is in view, the satellite position (p_s) (which is broadcast by the satellite) is put into a matrix.

$$G = \begin{bmatrix} d_x^1 & d_y^1 & d_z^1 & 1 \\ d_x^2 & d_y^2 & d_z^2 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ d_x^n & d_y^n & d_z^n & 1 \end{bmatrix} \tag{3.20}$$

where $d^i = (p_r - p_s^i)/|(p_r - p_s^i)|$ is the normalized vector pointing from the estimated receiver position to the satellite position. The estimated range error is given by

$$\delta\rho = \begin{bmatrix} \delta\rho^1 \\ \delta\rho^2 \\ \vdots \\ \delta\rho^n \end{bmatrix} \tag{3.21}$$

where $\delta\rho^i = R' - (p_r - p_s^i)$ is the difference in the pseudorange calculated from the timing signal and the range calculated from the current estimated receiver position and broadcast satellite position. The receiver position correction vector is then calculated by

$$\delta x = \delta x + ((G^T G)^{-1} G^T)\delta\rho \tag{3.22}$$

The estimated position is then updated by

$$p_r = p_r + \delta x \tag{3.23}$$

And the process is repeated until $|\delta x| < 0.1$.

4 LiDAR Simulation

There are several important sources of error in LiDAR measurements. In time-of-flight (TOF) LiDAR systems (which are the predominant type used on outdoor robots), there are intrinsic errors introduced by the accuracy and precision of the timing system and electronics. For well-calibrated commercial systems, these errors are typically on the order of a few centimeters and can be modeled as Gaussian noise. However, errors introduced by environmental conditions, including target reflectance and geometry, weather, and beam scattering, can be significantly higher. In VANE, LiDAR scans are simulated as conical pulses in order to estimate the effects of these errors. While dust and rain influences on LiDAR are discussed in the following chapters, this chapter focuses on the basics of the LiDAR simulation with respect to geometry and reflectance of the environment.

Real LiDAR systems can be quite complex and their performance is influenced by factors such as the power of the laser, the shape and duration of the beam pulse, the receiver electronics, and the relative geometry between the laser and receiver. In addition, most commercial LiDAR perform signal processing on the return signal to filter out noise and accurately extract distance from a return that is typically broadened in time. For commercial LiDAR systems these design properties are usually difficult or impossible to discern, especially because certain aspects, like the signal processing, may be private, proprietary information. However, as mentioned in Chapter 1, the primary purpose of the VANE is not sensor design and analysis; the purpose of the VANE is to predict environmental factors impact on sensor signals and ultimately robotic perception. Therefore, the VANE model uses a few simple parameters (Table 4-1) to model the LiDAR sensor. The scan pattern and beam properties can usually be found in product specification sheets and the signal processing properties can be estimated from bench scale laboratory tests or from simple rules of thumb.

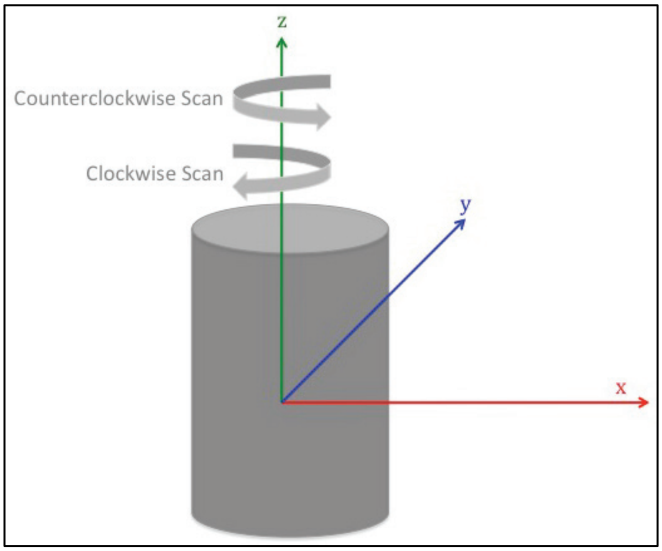
Table 4-1. Parameters in the VANE LiDAR model.

Parameter	Units	Description
Scan pattern (Figure 4-1)		
Sampling		Number of samples in the horizontal and vertical directions
Horizontal Angle Range	degrees	Minimum and maximum horizontal angles
Vertical Angle Range	degrees	Minimum and maximum Vertical angles
Sweep Direction		Clockwise or counter-clockwise
Beam properties (Figure 4-2)		
Max Range	meters	Maximum range of the system for 90% diffuse reflector
Minimum Radius	meters	Minimum size of the beam spot
Divergence	radians	Beam divergence
Laser Wavelength	nm	
Signal Processing (Figure 4-3)		
Distance Cutoff	meters	Maximum range between multipath signals
Noise Cutoff	fractional	Minimum amount of fraction signal to trigger sensor electronics
Optical Depth Threshold		See Ch. 10

4.1 Scan pattern

The VANE LiDAR coordinate system is shown in Figure 4-1. The horizontal angle range specifies the minimum and maximum angle of the scan pattern relative to the x-axis in the x-y plane, while the vertical angle range specifies the minimum and maximum angle of the scan pattern relative to the x-axis in the x-z plane. In each plane, the sampling is the number of scans in the plane between the minimum and maximum angles for each revolution of the sensor.

Figure 4-1. LiDAR coordinate system.



Most commercial outdoor LiDAR systems rotate or sweep through the x-y plane and the direction of this sweep in VANE is defined as either clockwise or counterclockwise, as depicted in Figure 4-1. As an example, the Velodyne VLP-16 (Velodyne 2017) has

Horizontal Angle Range = [-180,180]

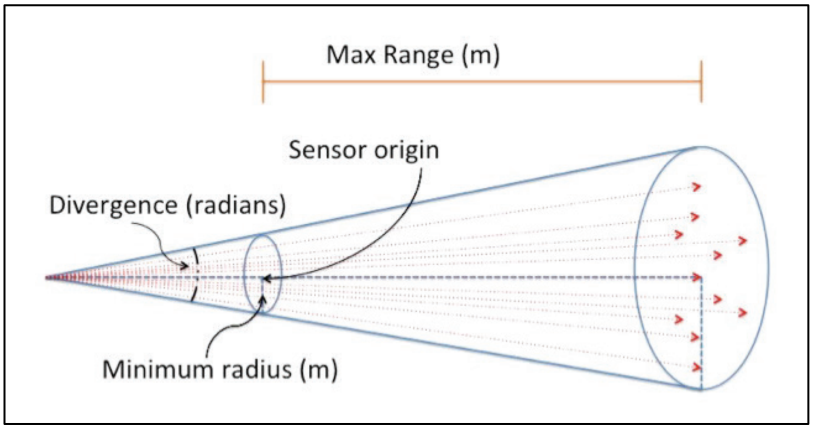
Vertical Angle Range = [-15.0, 15]

Sampling = [1875, 16]

4.2 Beam properties

In real LiDAR systems, much of the error is generated by the divergence of the LiDAR beam and the resultant broadening of the reflected signal. In VANE, this effect is captured by simulating each pulse as a diverging bundle of rays. A maximum of 25 rays per pulse are traced, so for a high-resolution sensor such as the VLP-16 listed above, this results is about 7.5 million rays per second of simulated time.

Figure 4-2. Ray sampling of LiDAR pulse.



The VANE assumes the beam is conical, implying a circular beam-spot. This assumption is not strictly correct for many systems. Future implementations will include several beam spot shapes such as circular, rectangular, and elliptical beams. For example, the VLP-16 has an elliptically shaped beam spot, with the longer dimension in the horizontal direction. However, even for systems with elliptical beam spots, the aspect ratio is typically close to one (VLP-16 horizontal/vertical = 1.33), so the effect of the assumption is negligible. Continuing with the VLP-16 example, in VANE, the beam properties are:

Minimum Radius = 0.0111 meters

Divergence = 0.003 radians

Max Range = 100 meters

Wavelength = 903 nm

4.3 Signal processing

LiDAR return pulses are broadened by both the divergence of the LiDAR beam and by the spatial extent of the target geometry. In order to calculate a distance from the return, some signal processing is required. A schematic of a typical LiDAR return is shown in Figure 4-3.

Figure 4-3. Example of LiDAR signal processing algorithm used in VANE.

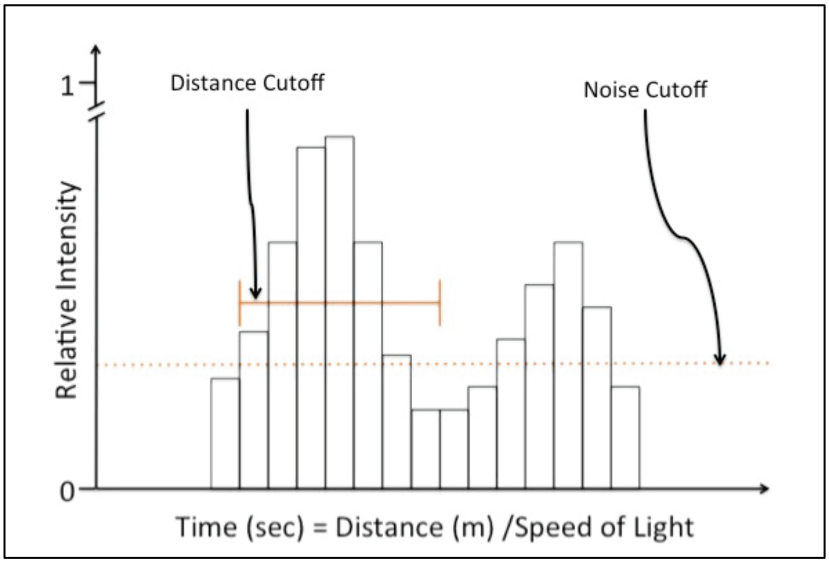


Figure 4-3 illustrates a typical case where there is a primary and secondary reflection. In the VANE signal that is processing the Noise Cutoff parameters it specifies the minimum returned signal strength that will open the sensor gate. Any signal received during the time window, specified by the Distance Cutoff parameter, will be stored and a weighted average (weighted with signal intensity) is used to calculate the return distance. While most modern commercial LiDAR sensors have several advances on this signal processing method, this is the basic technique used by all TOF LiDAR sensors.

The Distance Cutoff threshold can be measured in laboratory tests by setting up primary and secondary reflectors, then moving the secondary reflector back until the mixed pixel effect stops showing up (or in the case of multiple-return systems, the secondary return begins to show up). This is the Distance Cutoff parameter in the VANE model. In cases where the Distance Cutoff cannot be measured, typical values are around 1-3 m.

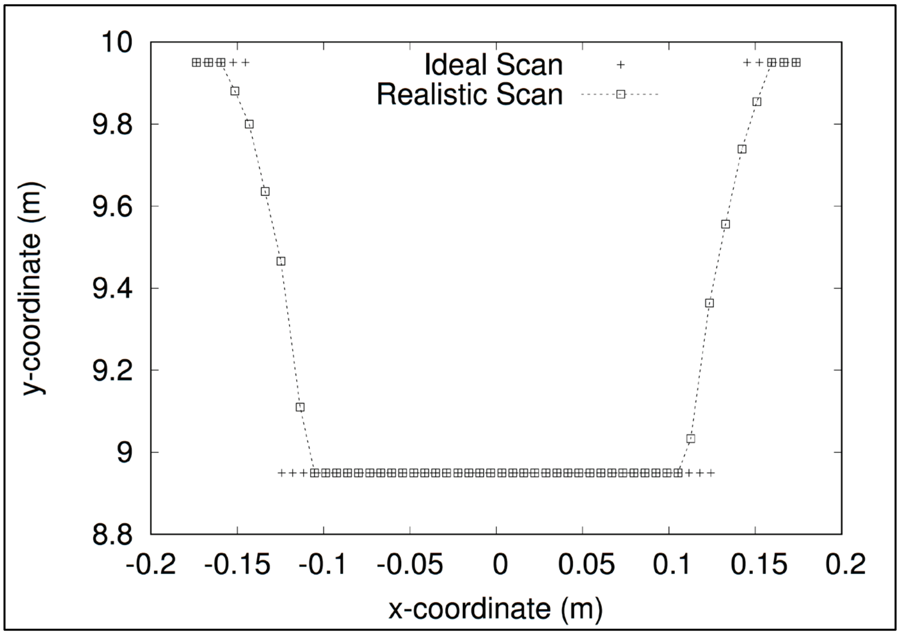
The Noise Cutoff parameters are somewhat more difficult to measure, but through laboratory tests and simulated experimentation, a good rule of thumb is

$$\text{Noise Cutoff} = 0.9/\pi/(\text{Max Range})^2$$

4.4 LiDAR simulation example results

Figure 4-4 shows the result of the beam divergence in the VANE LiDAR simulation. The figure shows the results from a small target in front of a larger background target. The + symbol is for an idealized scan with no beam divergence, while the □ symbol is for a more realistic simulation with beam divergence. While the ideal scan follows the exact position of the targets, there are points in between the two targets for the divergent beam. These are commonly referred to as "mixed pixels." While this example is a fairly extreme case, this is nevertheless a common phenomenon in real LiDAR that the VANE accurately reproduces.

Figure 4-4. Example VANE LiDAR scans with and without beam divergence.



5 Automotive Radar Simulation

Many modern vehicles use automotive radar for driver-assist functions like collision detection and avoidance (Stanislas and Peynot 2015). Automotive radar systems have a variety of different operating principles and rely very heavily on detailed signal processing techniques that are typically proprietary. Therefore, automotive radar in the VANE is simulated using a semi-empirical approach that captures the salient features of most automotive radar sensors without attempting to recreate the full-waveform generation, propagation, reflection, and processing that would be required for a fully physics-based radar simulation.

In VANE, automotive radars are characterized by four parameters, listed in Table 5-1. These parameters are typically found in the specification sheets for automotive LiDAR sensors.

Table 5-1 Parameters used in the VANE automotive radar simulation.

Parameter	Units
Max Range	meters
Horizontal Aperture	degrees
Vertical Aperture	degrees
Lateral Resolution	meters

The coordinate system for the radar is identical to that of the LiDAR, shown in Figure 4-1. The horizontal aperture and vertical aperture are similar to the horizontal angle range and vertical angle range parameters for the LiDAR, except in the radar it is assumed that the radar signal is centered around the x-axis in both the horizontal and vertical direction. The lateral resolution is the spatial scale at which the radar can resolve targets.

A VANE radar will return a list of targets with a location and return intensity. This return is similar to most automotive radar on the market today, although many commercial radars perform processing and return a smaller subset of this information, depending on the application. Additionally, some commercial systems also return relative velocity between the sensor and target, but this feature is not currently implemented in the VANE.

In VANE, for each radar scan, a set of rays evenly samples the beam pattern of the radar at the range and spatial resolution specified by the sensor and a set of discrete returns is generated, somewhat similar to a LiDAR point cloud, but it is a simulated radar signal. The points are placed in a grid and connected component labeling is used to determine which points belong to the same target. For each target identified, the location of the target is taken as the centroid of the cluster of points.

An example of output of the radar model, compared to a LiDAR scan, is shown in Figure 5-1 for the simple scene depicted in Figure 5-2. In the figure, the radar returns are the large X's, while the LiDAR returns are the smaller +'s.

Figure 5-1 Radar returns (X) compared to lidar returns (x) for a simple scene with cuboid obstacles.

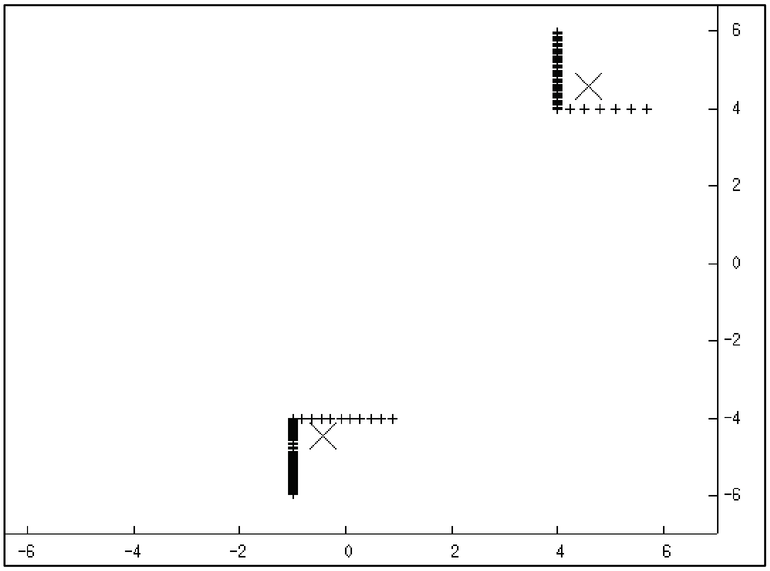
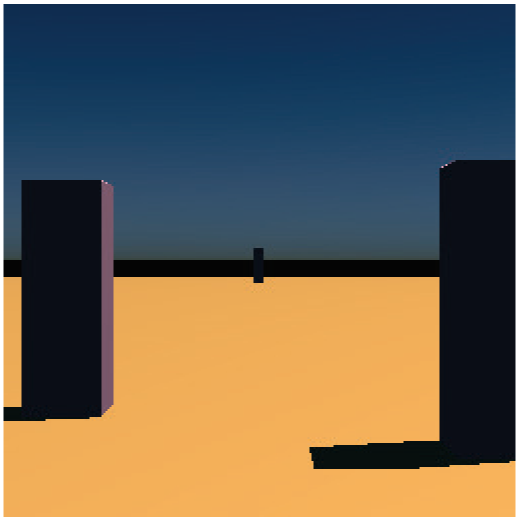


Figure 5-2. Simple scene used for radar comparison.



6 Modeling the Influence of Rain on Automotive LiDAR Sensors

This chapter summarizes the implementation of the rain model in VANE. The goal of this model is to realistically simulate the influence of rain on modern automotive LiDAR systems while being simple enough to parameterize using only data that is provided in LiDAR specification sheets. In order to achieve this goal, several simplifying assumptions to the LiDAR equation are made, as described in the next section.

Rain primarily impacts modern automotive LiDAR by reducing the range and accuracy of the system. Rasshofer et al. (2011) showed the max detection range of automotive LiDAR is reduced as a function of rainfall rate, and Filgueira et al. (2017) measured the range error introduced by rain.

Regarding false positives in the LiDAR point cloud introduced by rain, Fersch et al. (2016) and Wang et al. (2013) found there is a negligible chance of false positives caused by backscatter from raindrops, even for heavy rain. For ranges less than 5 m, Rasshofer et al. (2011) found a false alarm rate of 2.5 $s^{-1}m^{-1}$ for LiDAR in heavy snowfall. However, it is noted that one would expect much higher backscatter from heavy snow than from rain in the optical wavelengths. Additionally, ranges of less than 5 m are often filtered out of the LiDAR point cloud for automotive LiDAR applications. Therefore, false positives caused by backscatter from rain will be neglected in the VANE model.

6.1 Rain - LiDAR interaction model

The summary on LiDAR theory in this section is taken from Rasshofer et al. (2011).□The received signal power to a LiDAR system is given by the LiDAR equation.

$$P(L) = C_A \int_{t'=0}^{2L/c} P_T(t') H\left(L - \frac{ct'}{2}\right) dt \qquad (6.1)$$

where C_A is a property of the LiDAR system that does not depend on range or time. In the VANE model, returns are calculated based on the relative, rather than absolute, intensity of the reflected signal. Therefore, $C_A = 1$ in the VANE model. $P_T(t)$ is the power of the LiDAR pulse as a function of time. For the VANE model, it is assumed that the simplest case where $P(t) = E_p\ \delta(t)$ where E_p is the peak laser energy. Again, because VANE uses relative intensities for the LiDAR calculation, $E_p = 1$ is set. Therefore, Equation 6.1 reduces to

$$P(L) = H(L) \tag{6.2}$$

Here, H(L) is the spatial impulse response function and can be split into two parts; the spatial impulse response of the optical channel (H_C) and the spatial impulse response of the target (H_T) such that $H = H_C H_T$.

The response function of the optical channel is given by

$$H_C(L) = \frac{T^2(L)}{L^2}\xi(L) \tag{6.3}$$

where T(L) is the "total one-way transmission loss introduced by losses in the transmission media." In this case, the lossy medium is rain.

$$T(L) = e^{-\int_0^L \alpha(L')dL\prime} \tag{6.4}$$

The crossover function, ξ(L), is related to the configuration of the laser and receiver geometry. This equation is simplified by making two assumptions. First, while the value of ξ can theoretically vary between 0 and 1, it is assumed ξ = 1, which represents optimal performance of the system. It is noted that this only introduces small errors at ranges nearest to the detector. Second, for a homogenous transmission medium (such as rain of uniform intensity), the extinction coefficient is constant is assumed such that α(L) = α. With these assumptions, Equation 6.3 becomes

$$H_C(L) = \frac{e^{-2\alpha L}}{L^2} \tag{6.5}$$

The target response function for a hard target at range L_o is given by

$$H_T(L) = \rho\delta(L - L_0) \tag{6.6}$$

where ρ is the reflectance of the target as quantified by the BRDF and L_o is the range of the target. For targets smaller than the beam diameter at L_o, H_T will be reduced by the fractional area of the target to the beam-spot. However, targets this small are typically not of interest in automotive applications and are not considered in this analysis. With these simplifications, the LiDAR equation for a hard target at range L_o being detected through a scattering media (rain) with extinction coefficient α is given by

$$P(L_0) = \rho \frac{e^{-2\alpha L_0}}{L_0^2} \tag{6.7}$$

6.2 Estimating the scattering coefficient

In order to implement Equation 6.7, the dependence of the extinction coefficient, α, will need to be quantified on the rainfall rate, Lewandoski et al. (2009) found empirically that the extinction coefficient is related to the rainfall rate by a power function.

$$\alpha(R) = 0.063R^{0.37} \tag{6.8}$$

However, in the VANE implementation, this particular parameterization did not reproduce the range-reduction measured in Rasshofer et al. (2011). In order to match the published values for the range reduction, we instead let the coefficient and exponent of Equation 6.8 be parameterized such that

$$\alpha(R) = aR^b \tag{6.9}$$

By varying a and b in the VANE simulations, the optimum values were able to be found to match the range reductions published in Rasshofer et al. (2011), with the result that a = 0.003 and b = 1.0.

$$\alpha_{\text{VANE}}(R) = 0.003R \tag{6.10}$$

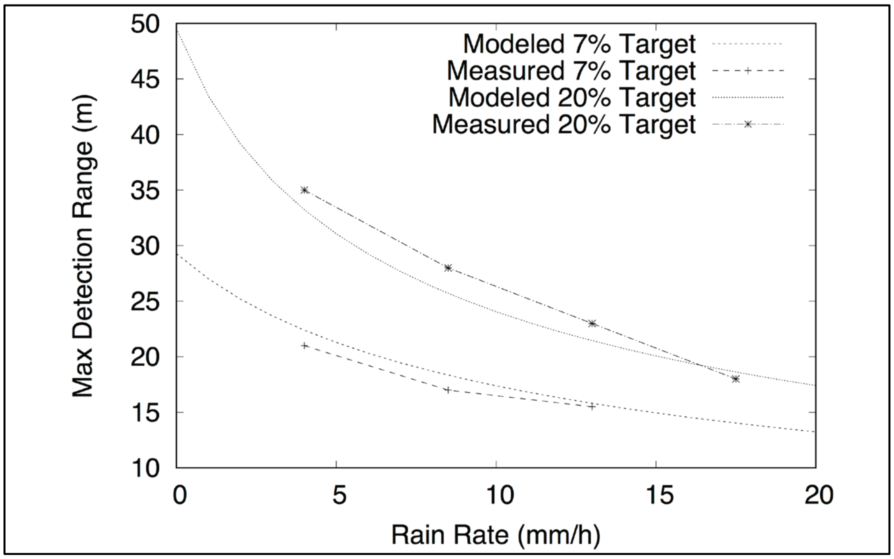
For a given rainfall rate, Equations 6.7 and 6.10 can be used to calculate the maximum range of the sensor for that rainfall rate. In order to do so, some parameterization of the sensor power was needed. Since most LiDAR product data sheets list the maximum range of the sensor, a model based on this input has been deployed. For a sensor with a maximum range $L_{\max}$, calculate the detection threshold, δ, for the sensor by

$$\delta = \frac{0.9}{\pi L_{\max}^2} \tag{6.11}$$

Here, it is assumed that the maximum range is for a 90% reflective Lambertian target. The result of Equation 6.11 can be compared to this threshold to determine the maximum range for a given rainfall rate.

Figure 6-1 shows the results of the VANE model compared to measured values listed in Figure 9 of Rasshofer et al. (2011) for Sensor A. Although no parameters are listed for the sensor, it was found that the model matched the data for a maximum range of 105 m. It is noted that this is a typical maximum range for automotive LiDAR sensors, giving confidence in the validity of the model.

Figure 6-1 Output of the VANE model for a LiDAR with a maximum range of 105 m, compared to measurements given in Rasshofer et al. (2011).



6.3 Implementation in VANE

To implement the rain model in VANE, Equations 6.7 and 6.10 were used to modify the reflected intensities calculated by the model with no rain. If the reflected intensity falls below the detection threshold, δ of the LiDAR sensor, the return is removed from the LiDAR point cloud, otherwise the returned intensity is modified. Finally, noting that Filgueira et al. (2017) and Starik and Werman (2003) showed that light rain typically introduces range errors of less than 1% for most targets, some small random noise was added to the ranges that are retained. In detail, these steps are

1. For each LiDAR pulse, i, calculate the range ($L[i]$), target reflectance ($\rho[i]$), and reflected intensity ($I[i]$) using the VANE ray-tracer.

2. Calculate the extinction coefficient given the rainfall rate using Equation 6.9.
3. To add some randomness to the simulation, calculate an effective extinction:
   a. $\alpha_{\mathrm{eff}} = \mathrm{rand}(0.5{,}1.0), \alpha$
4. Calculate the reflected intensity using Equation 6.7:
   a. $P(L[i]) = \frac{\rho[i]}{L^2[i]} e^{-2\alpha_{\mathrm{eff}} L[i]}$
5. If $P(L[i]) < \delta$, remove the point from the point cloud, otherwise, update $I[i] = P(L[i])$
6. If point is retained, update the range by introducing a random error with a 1% maximum:
   a. $L[i] = rand(0.99\, L[i], 1.01L[i])$

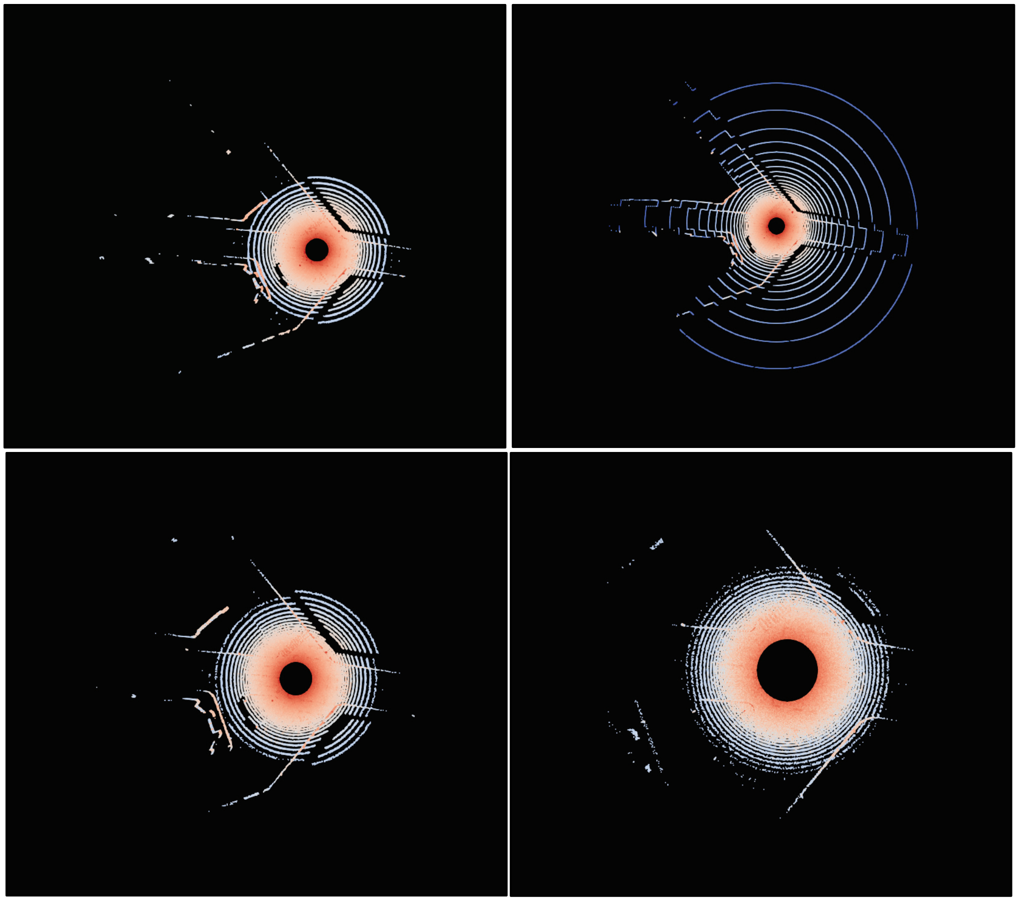
6.4 Simulation results

In order to demonstrate the result of the model in a VANE simulation with a complex scene, simulations were run of LiDAR and camera sensors in clear and rainy conditions. Three rainfall conditions were used: light (1 mm/h), medium (6.4 mm/h), and heavy (25.4 mm/h). The LiDAR had a maximum range of 120 m. The reduction in range as a function of rainfall rate is shown in Table 6-1. Note that in a complex environment like this, the maximum range is influenced not only by rainfall rates, but also by the orientation and reflectance properties of the surfaces in the environment. Therefore, the curve should not be expected to strictly follow Equation 6.7. Nevertheless, a reduction is seen in range as rainfall is increased, as expected.

Table 6-1 Maximum simulated range for a 120 m LiDAR in rainy environment.

Rainfall rate (mm/h)	Max range (m)
0.0	118.8
1.0	116.0
6.4	73.9
25.4	40.4

Figure 6-2. Example LiDAR scans for different rain rates.



7 Simulating Rain in Images

When simulating the impact of rain on images, it is not sufficient to treat the rain as homogeneous transmission medium as was done in the LiDAR simulation. This is because individual raindrops in the near field can clearly be identified in images as streaks across the frame. However, as Starik and Werman (2003) verified experimentally, for typical commercial cameras operating at frame rates of 30-60 Hz, individual raindrops cannot be tracked from frame to frame, *i.e.* each frame is independent. Therefore, rain streaks in VANE are generated using independent random generators for each frame, easing the necessity of tracking the dynamics of individual raindrops, following the outline presented in Starik and Werman (2003).

The first step in rendering the rain streaks is calculating the number density of raindrops as a function of rainfall rate. This is accomplished using correlation performed by Feingold and Levin (1986), in which they found the number density of raindrops is related to the rainfall rate by

$$N_T = 172R^{0.21} \tag{7.1}$$

Similarly, Feingold and Levin (1986) found the geometric mean of the raindrop diameter (in meters) is given by

$$D_g = \frac{0.75}{1000}R^{0.21} \tag{7.2}$$

The terminal velocity of the raindrops is given by fitting the data from Table 7-1, taken from Horstmeyer (2015). Best fit of the data from Table 7-1 gives the following equation for terminal velocity with respect to drop diameter.

$$v_{\text{terminal}} = 3.1654\ \ln D_g + 26.006 \tag{7.3}$$

Table 7-1 Terminal velocities of raindrops as a function of drop size.

Drop Diameter (m)	Terminal Velocity (m/s)
0.0005	2.06
0.001	3.98
0.0012	4.60
0.002	6.44
0.0026	7.20
0.004	8.76
0.005	9.03

A uniform wind direction is also added to the model, as defined by the user, so the final raindrop velocities are given by

$$v_{\text{drop}} = v_{\text{terminal}} + v_{\text{wind}} \tag{7.4}$$

Finally, in order to add a degree of randomness and non-uniformity to the raindrop simulation, a uniform random variation of ±15% is added to each component of the velocity.

To render the appearance of rain streaks in the image, a set of raindrops is randomly generated in the camera's view frustum. The view frustum is cut off at a maximum range of

$$r_{max} = f\ \frac{D_g}{0.02d_{\text{pix}}} \tag{7.5}$$

where f is the focal length of the camera in meters and d_{pix} is the dimension of the pixels on the imaging plane. This equation limits the rain such that raindrops with projected dimension on the focal area of less than 2% of the pixel size will not be rendered.

The total number of drops generated is equal to the volume of the view frustum times the number density from Equation 7.1. Each raindrop's initial pixel position (u, v) and velocity direction on the image frame is calculated using a simple camera projection based on the camera's focal length and pixel dimensions. The length of the streak is then calculated using the exposure time, t_{exp} of the camera. For a drop at an orthogonal distance Z from the imaging plane, the magnification of the camera in pixels is $M_P = \left(\frac{f}{Z}\right)/d_{pix}$. The length of the streak in pixels in then given by

$$l_{\text{streak}} = M_p v_{\text{drop}} t_{\text{exp}} \tag{7.6}$$

The width on the image plane, in pixels, for each streak is determined according to the equation

$$w_{\text{streak}} = M_p D_g \tag{7.7}$$

After the location, width, and length of each streak are determined, the streaks are rendered to a mask. The mask modifies the intensity of the background image while taking into account the depth map of the original image so that rain streaks behind objects are not rendered. The approach of Starik and Werman (2003) was applied and an adaptive additive model used, which has the following assumptions (quoted from Starik and Werman [2003]).

1. Adding a rain stroke to the image increases the intensity of the background pixels.
2. The intensity increases in inverse proportion to the background intensity (which means that more rain color needs to be added to a dark background and less to a bright one).
3. The drops are transparent, meaning that the original pixel color is mixed with the raindrop color.
4. A rain-stroke blurs the background behind it.

Example rain masks for light (1.02 mm/h), medium (6.35 mm/h), and heavy (25.4 mm/h) rain are shown in Figure 7-1. The application of these masks to an image is shown in Figure 7-2.

Figure 7-1 Rain masks for different rain rates.

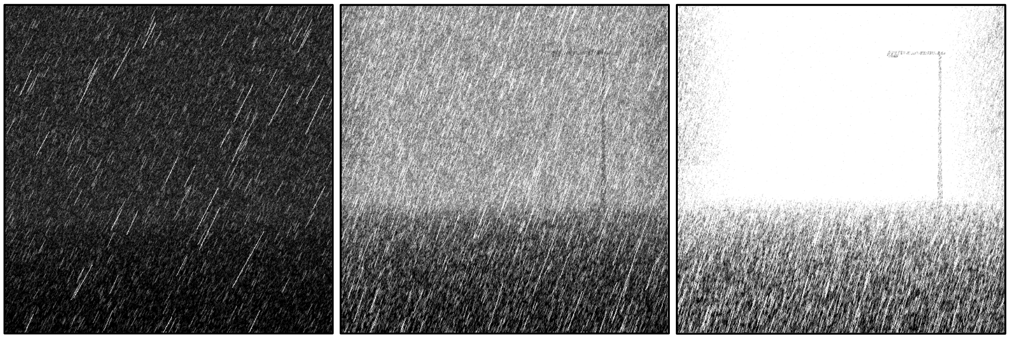
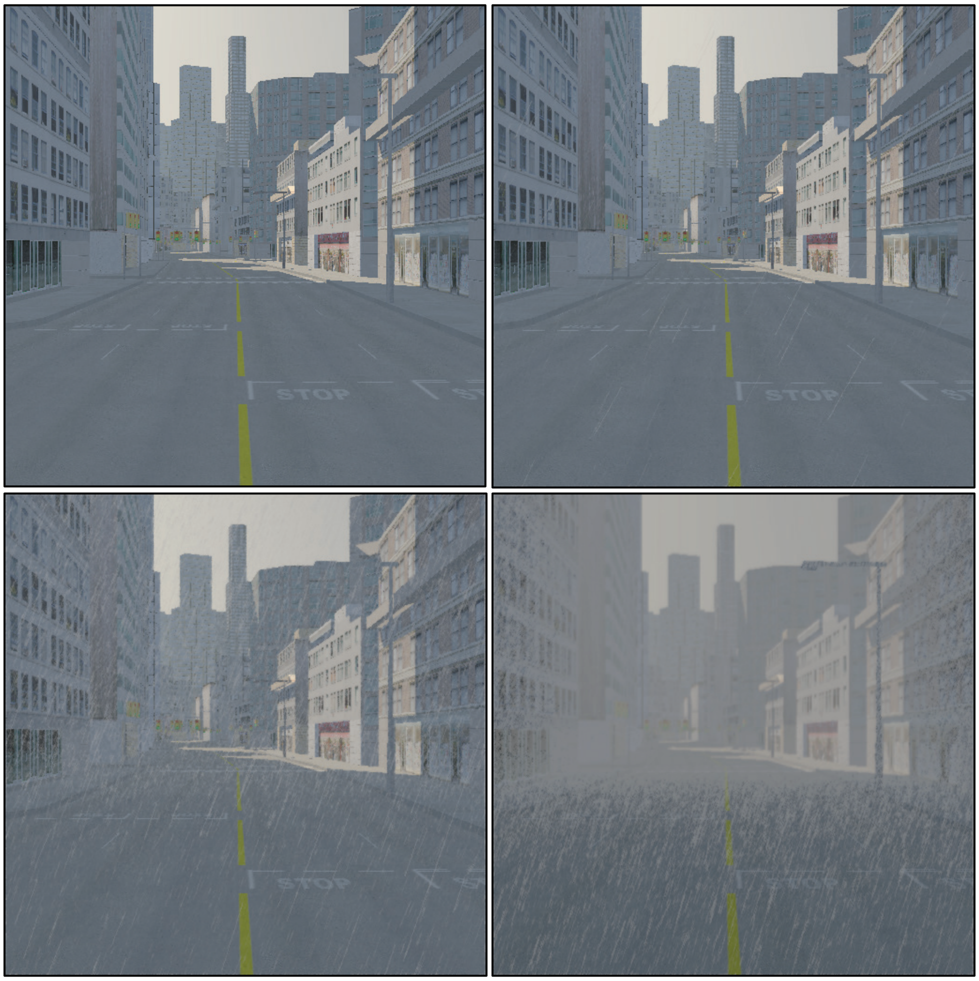


Figure 7-2. Example images for different rain rates.



For a pixel $I(u, v)$ in the background image, the rain mask $RM(u, v)$ is added to the final pixel $I_{rain}(u, v)$ according to the formula

$$I_{\text{streaked}}(u, v) = I(u, v) + RM(u, v)(1 - I(u, v))(1 - I_{\text{avg}}) \qquad (7.8)$$

where I_{avg} is the average intensity of the background image. Blurring is introduced according to the formula

$$I_{\text{rain}}(u, v) = I_{bs}RM(u, v) + I_{\text{avg}}(1 - RM(u, v)) \qquad (7.9)$$

where I_{bs} is a local average of the streaked image from Equation 7.8. Following Starik and Werman (2003), the average is taken over a 3x3 pixel stencil.

8 Modeling Dust with Particle Systems

Dust is simulated in the VANE using a particle system. The primary influence of dust is to serve as an obscurant around moving vehicles that generate dust during travel (Tong 2014). Unlike rain (Chapter 7), there is very little published material that relates the density of airborne dust to the scattering coefficients for the cases of relatively large particles entrained close to the ground. While there are published measurements of the scattering properties of fine-scale dust that is entrained in the atmosphere (Shimizu et al. 2004) and (Liu et al. 2002), these measurements are not relevant for the VANE because the dust particles generated by moving vehicles are orders of magnitude larger and the clouds have several orders of magnitude higher number density. Therefore, a semi-empirical approach was adopted for modeling dust in the VANE.

8.1 Dust particle dynamics

In VANE, dust is generated by moving vehicles, with the number of particles added at each time step being proportional to the vehicle velocity. The EPA AP-42 model was used, as presented in Tong (2014) to estimate the amount of dust generated by the vehicle. In the model, the emission factor, EF, is given by

$$EF = \frac{k(s/12)^a(W/3)^b}{(M/0.2)^c} \tag{8.1}$$

where EF has units of pounds of dust per vehicle mile traveled (lb/VMT), k is a dimensionless scale factor, s is the % silt of the road surface, W is the vehicle weight in pounds, and M is the surface moisture content. As noted by Tong (2014), the values of k, a, b, and c vary depending on which class of particle matter (PM) is being considered. Typically, $PM_{2.5}, PM_{10}, or\ PM_{30}$ are used for calculations, depending on the application, where the subscript is the size in microns of the maximum particle size of the classification. In VANE, calculations were performed for PM_{30} because we are interested in the larger particles entrained in the near field of a moving vehicle. In this case, $k = 10, a = 0.8, b = 0.5, and\ c = 0.4$ (Tong 2014).

Although Equation 8.1 is a cumulative total, it is used dynamically in the VANE by multiplying by the vehicle velocity and time step. For a given VANE time step with length dt seconds, the mass in kg (m_{dust}) of dust added to the simulation is given by

$$m_{\text{dust}} = 0.000282(EF)v_{\text{veh}}dt \tag{8.2}$$

where v_{veh} is the vehicle velocity in m/s and the factor 0.000282 converts mph to m/s and pounds to kg. As Tong (2014) points out, there is very little information about the localized distribution of entrained particles around a moving vehicle. In VANE, it is simply injecting particles behind the rear wheels of the vehicle. The total number of particles injected at each time step is related to the mass of the particles by

$$n_{\text{injected}} = m_{\text{dust}} + m_{\text{particle}} \tag{8.3}$$

where $m_{particle}$ is the mass of the individual particles in the particle system.

8.1.1 Particle Dynamics Model

As discussed above, particles are added to the simulation behind the rear wheels of the vehicle. The particles are randomly added in a spherical volume with radius equal to half the vehicle width directly behind the tire. The initial velocities of the particle are related to the velocity of the vehicle by

$$v_0^{\text{dust}} = [0.5v_x^{\text{veh}}, 0.5v_y^{\text{veh}}, 0.75|v^{\text{veh}}|] \tag{8.4}$$

Some randomness is added to the initial velocities by choosing a randomly oriented vector and a random magnitude between 0 and 0.5 of the calculated initial velocity.

$$v_0^{\text{dust}} = v_0^{\text{dust}} + \text{rand}\left(0{,}0.5\left|v_0^{\text{dust}}\right|\right)\ \text{randomVector}() \tag{8.5}$$

At each time step, the acceleration of each particle is calculated using gravity and the drag coefficient.

$$a^{\text{dust}} = g - a_{\text{drag}}\, v^{\text{dust}}|v^{\text{dust}}| \tag{8.6}$$

Where a_{drag} is the drag factor, to be discussed below. The velocity of the position of the particle is then updated according to the equation

$$p^{\text{dust}} = p^{\text{dust}} + v^{\text{dust}}dt + 0.5a^{\text{dust}}dt^2 \tag{8.7}$$

And finally, the velocity of the dust for the next time step is updated by

$$v^{\text{dust}} = v^{\text{dust}} + a^{\text{dust}}dt \tag{8.8}$$

In VANE, the drag force on the dust particle is time dependent. This is to account for the natural diffusion of the dust over time. Each particle is given a maximum lifetime; for dust particles this is typically 10 sec. This is necessary from a computational perspective to keep the total number of particles from continually increasing during the simulation. It also allows for dust density diffusion to be modeled. Given that the time dependent term in the diffusion equation can be modeled $f(t) = \ e^{-\lambda^2 t}$, where λ is a constant, the drag increase is taken proportionally to the % diffusion, with a maximum diffusion occurring at the end of the particles life:

$$1 - e^{-\lambda^2 t_{\text{max}}} = 0.99 \tag{8.9}$$

This implies that $\lambda^2 \ = 4.6/t_{\text{max}}$. Therefore, the drag factor is then given by

$$a_{drag} = C_d(1 - e^{-4.6t/t_{max}}) \tag{8.10}$$

Physically, the drag coefficient, *Cd*, should account for a variety of factors including air viscosity, particle mass, and particle shape. However, since the VANE particle system drag is essentially an empirical model, $C_d \ = 1$ is adequate for realistic results.

8.2 Volume ray-tracing

In order to perform ray-tracing on the particles, they are first assigned to a density grid. The grid is comprised of cubic, axis-aligned bounding volumes, the dimensions of which are set to ten times the radius of the dust particles. All the particles are placed into one of the cells and the number of particles in each cell is counted. Each cell is assigned a number density per unit volume, N, based on the particle count in each cell. Theoretically, the Beer-Lambert law for the intensity of a ray passing through the medium is given by

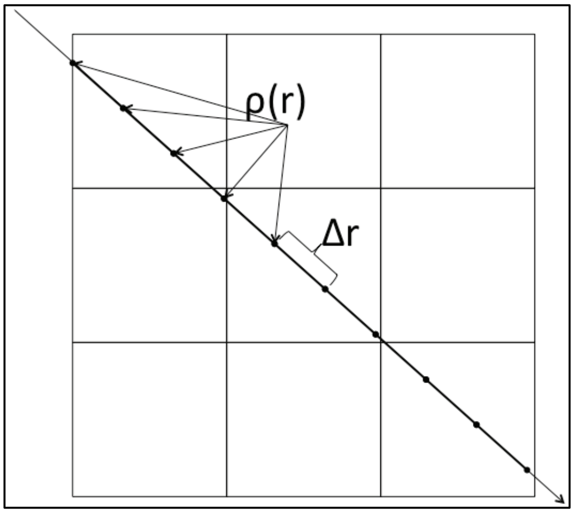
$$I = I_0 e^{-\sigma_m L \int_0^L \rho(L')dL\prime} \tag{8.11}$$

where σ_m is the mass-specific extinction cross section of the material (m^2/kg), L is the path length through the medium (meters), and ρ(L) is the spatially varying mass concentration of dust (kg/m^3) along the path length L. In VANE, the intensity reduction, I/I$_0$, of a ray passing through a dust cloud is calculated by replacing the integral in the exponent of Equation 8.11 with a summation for numeric integration along the path length. Equation 8.11 then becomes

$$\frac{I}{I_0} = \tau_{\text{dust}} = e^{-\sigma_m \Delta r \sum \rho(r)} \quad (8.12)$$

where ρ(r) is the spatial variation of the mass density in the dust cloud. For a given ray passing through the volume grid of the dust cloud, the ray is subdivided into segments and the integration performed at the endpoint of each segment, with ρ(r) calculated by interpolating to the centers of the grid cells. Figure 8-1 below provides an illustration of the volume rendering approach in the VANE.

Figure 8-1. 2-D illustration of the volume rendering algorithm in VANE.



In VANE, Δr is set to half the dimension of the grid cells. The mass-specific extinction cross section, σ_m, can vary considerably depending on the mineralogical composition of the dust (Sokolik and Toon 1999). Gerasopoulos et al. (2009) found that $\sigma_m \approx 650$ m^2/kg for visible and NIR wavelengths. This was the value for relatively small particles entrained in the atmosphere, and it has been found through visual comparison to field tests that $\sigma_m \approx 2600$ m^2/kg (a factor of 4 higher than [Gerasopoulos et al. 2009]) gives more realistic results for vehicle-generated dust.

8.2.1 Dust in the camera simulation

In the VANE camera simulation, the image is first rendered without the particle system. The particle system is then rendered on top of the existing frame using Equation 8.12 to calculate the attenuation of the light passing through the dust cloud. Since the dust particle also reflects light, light is added to the pixel based on the optical depth of the cloud

$$L_{\text{dust}} = E_{\text{sun}}(1 - \frac{I}{I_0})\rho_{\text{dust}} \tag{8.13}$$

where L_{dust} is the light reflected from the dust, E_{sun} is the irradiance of the sun, and ρ_{dust} is the bulk reflectance of the dust particles so that the total radiant flux for the pixel is given by

$$I_{\text{dust}} = \tau_{\text{dust}} I_{\text{clear}} + L_{\text{dust}} \tag{8.14}$$

where τ_{dust} is defined in Equation 8.12.□

8.2.2 Dust in the LiDAR simulation

Previous work has shown the probability of a LiDAR returning from a cloud of dust is related to the optical depth of the dust cloud (Goodin et al. 2013). Specifically, the probability that a given scan will return from a dust cloud is linear with the optical depth up to a certain point, after which the LiDAR will always return from the dust cloud and nothing behind it.

In VANE, the initial ray-tracing is performed on the scene without the dust. First, Equation 8.12 is used with the volumetric ray-tracer to calculate the one-way transmission loss (Goodin et al. 2013). From this, the optical depth, μ is found by

$$\mu = \ln\left(\frac{I}{I_0}\right) = -\sigma_N \sum N(r)\Delta r \tag{8.15}$$

If the optical depth is above the threshold for the sensor μ_{thresh}, then a return is generated from within the dust cloud. The threshold parameter is an input to the LiDAR model and must be supplied by the user. Typical values range from $0.1 - 0.3$ for TOF systems.

If the optical depth is below μ_{thresh}, then the return is calculated using the Monte-Carlo technique. First, a test value is generated between 0 and μ_{thresh}. If μ is greater than the test value, a LiDAR return from the dust cloud is generated. If μ is less than the test value then no return is generated, but the intensity of the existing return (if there was one) is modified by Equation 8.12.

9 Conclusion

The purpose of this document is to give a detailed description of the physics models implemented in the VANE for simulating sensors and the environmental impact on sensors, including any simplifying assumptions or implementation details. Several additional reports are planned for future publication to include the development and handling of terrain databases for the VANE; verifying and validating the sensor and environment models within the VANE; the performance implementations and analysis of the ray-tracing engine within the VANE; and utilizing and extending the VANE for integrating, training, and analyzing autonomy algorithms.

References

Bouget, J. Y. 2015. *Camera Calibration Toolbox for MATLAB*. October 14. Accessed October 2, 2017. http://www.vision.caltech.edu/bouguetj/calib_doc/.

Brown, D. C. 1966. "Decentering Distortion of Lenses." *Photometric Engineering* 32 (3): 444-462.

Durst, P. J., C. Goodin, C. L. Cummins, B. Q. Gates, G. B. McKinley, T. R. George, M. M. Rohde, M. A. Toschlog, and J. Crawford. 2012. "A real-time, interactive simulation environment for unmanned ground vehicles: The autonomous navigation virtual environment laboratory (ANVEL)." *Fifth International Conference on Information and Computing Science (ICIC).* USA: IEEE. 7-10.

Durst, P. J., and C. Goodin. 2012. "High Fidelity Modelling and Simulation of Inertial Sensors Commonly used by Automous Mobile Robots." *World Journal of Modelling and Simulation* 8 (3): 172-184.

Feingold, G., and Z. Levin. 1986. "The Lognormal Fit to Raindrop Spectra from Frontal Convective Clouds in Israel." *Journal of Climate and Applied Meteorology* 25 (10): 1346-1363.

Fersch, T., A. Buhmann, A. Koelpin, and R. Weigel. 2016. "The Influence of Rain on Small Aperture LiDAR Sensors." *2016 German Microwave Conference*. Germany: IEEE. 84-87.

Filgueira, A., H. Gonzalez-Jorge, S. Laguela, Diaz-Vilarino, and P. Arias. 2017. "Quantifying the Influence of Rain in LiDAR Performance." *Measurement* 95: 143-148.

Gerasopoulos, E., P. Kokkalis, V. Amiridis, E. Liakakou, C. Perez, K. Haustein, K. Eleftheratos, M. O. Andreae, T. W. Andreae, and C. S. Zerefos. 2009. "Dust Specific Extinction Cross-Sections Over the Eastern Mediterranean Usin gthe BSC-Dream mdoel and Sun Photomoter Data: The Case of Urban Environments." *Annales Geophysicae* 27: 2903-2912.

Goodin, C., P. J. Durst, Z. T. Prevost, and C. J. Compton. 2013. "A Probabilistc Model for Simulating the Effecto fo Airborne Dust on Ground-Based LiDAR." *SPIE Defense, Security, and Sensing*. USA: SPIE. 87340.

Goodin, C., J. T. Carrillo, D. P. McInnis, C. L. Cummins, P. J. Durst, B. Q. Gates, and B. S. Newell. 2017. "Unmanned Ground Vehicle Simulation with the Virtual Autonomous Navigation Environment." *International Conference on Military Technologies*. Bruno: IEEE. 160-165.

Goodin, C. 2013. "Analytic Expressions for the Black-Sky and White-Sky Albedos of the Cosine Lobe Model." *JOSA A* 30 (5): 854-858.

He, X. D., K. E. Torrance, F. X. Sillion, and D. P. Greenberg. 1991. "A Comprehensive Physical Model for Light Reflection." *ACM SIGGRAPH* 25: 175-186.

Heikkila, J., and O. Silven. 1997. "A Four-Step Camera Calibration Procedure with Implicit Image Correction." *Computer Vision and Pattern Recongition*. USA: IEEE Computer Society. 1106-1112.

Horstmeyer, S. L. 2015. *Typcial Raindrop Sizes*. January 1. Accessed October 2, 2017. http://www.shorstmeyer.com/wxfaqs/float/rdtable.html.

Hosek, L, and A. Wilkie. 2013. *IEEE Computer Graphics and Applications* (IEEE) 33 (3): 44-52.

______. 2013. April 20. Accessed October 2, 2017. http://cgg.mff.cuni.cz/projects/SkylightModelling/.

Hosek, L, and A. Wilkie. 2012. *ACM Transactions on Graphics* 31 (4): 95.

Hwang, C., and L. Hwang. 2002. "Satellite Orbit Error Due to Geopotential Model Error Using Perturbation Theory: Applications to ROCSAT-2 and COSMIC Missions." *Computers and Geosciences* 28 (3): 357-367.

Kim, H. I., J. H. Ha, K. D. Park, and S. U., Kim, J. H. Lee. 2009. "Comparson of Tropospheric Signal Delay Models for GNSS Error Simulation." *Journal of Astronomy and Space Sciences* 26 (2): 211-220.

Koenig, N., and A. Howard. 2004. "Design and Use Paradigms for Gazaebo, and Open-Source Multi-Robot Simulator." *IROS*. USA: IEEE. 2149-2154.

Lewandowski, P. A., W. E. Eichinger, A. Kruger, and W. F. Krajewski. 2009. "LiDAR-Based Estimation of Small-Scale Rainfall." *Journal of Atmospheric and Oceanic Technology* 26 (3): 656-664.

Lewis, R. R. 1993. "Mskinh Shaders More Physically Plausible." *Computer Graphics Forum* 13: 109-120.

Liu, Z., N. Sugimoto, and T. Murayama. 2002. "Extinction-to-Backscatter Ratio of Asian Dust Observed with High-Spectral-Resolution LiDAR and Raman LiDAR." *Applied Optics* 41 (15): 2760-2767.

Meagher, D. 1980. *Octree Encoding: A New Technique for the Representation, Manipulation and Display of Arbitrary 3-D Objects by Computer*. Image Processing Laboratory, Renssalaer Polytechnic Institute, Troy, New York: RPI, IPL-TR-80-111.

Michal, D. S., and L. Etzkorn. 2011. "A Comparison of Player/Stage/Gazebo and Microsoft Robotics Developer Studio." *Proceedings of the 49th Annual Southeast Regional Conference*. USA: ACM. 60-66.

Montenbruck, O., and M. Garcia-Fernandez. 2005. *Ionospheric Path Delay Models for Spaceborne GPS*. DLR-GSOC TN: German Space Operations Center.

Niell, A. E. 2000. "Improved Atmospheric Mapping Functions for VLBI and GPS." *Earth, Planets, and Space* 52 (10): 699-702.

Peters, J. F., and C. Goodin. 2015. *Software Implementation of a Polarized Bidirectional Reflectance Distribution Funcion Model for Ray-Tracing Applications.* Vicksburg: U.S. Army Engineer Research and Development Center.

Platt, U., K. Pfeilsticker, and M. Vollmer. 2007. "Radiation and Optics in the Atmosphere." *Springer Handbook of Lasers and Optics* 1165-1203.

Preetham, A. J., P. Shirley, and B. Smits. 1999. "A Practical Analytical Model for Daylight." *Proceedings of the 26th Annual Conference on Compter Graphics and Interactive Techniques.* USA: ACM. 91-100.

Rasshofer, R. H., M. Spies, and H. Spies. 2011. "Influences of Weather Phenomena on Automtove Laser Radar Systems." *Advances in Radio Science* 9 (2): 49-60.

Rymes, M. 2004. *Bird Simple Spectral Model: spectrl2.* March 17. Accessed October 3, 2017. http://rredc.nrel.gov/solar/models/spectral/spectrl2/.

Shimizu, A., N. Sugimoto, I. Matsui, K. Arao, I. Uno, T. Murayama, N. Kagawa, K. Aoki, A. Uchiyama, and A. Yamazaki. 2004. "Continuous Observations of Asian Dust and Other Aerosols by Polarization LiDARs in Chana and Japan During ACE-Asia." *Journal of Geophysical Research: Atmospheres* 109 (D19): 0.

Sokolik, I. N., and O. B. Toon. 1999. "Incorporation of Mineralogical Composition into Models of the Radiative Properties of Mineral Aersol from UV to IR Wavelengths." *Journal of Geophysical Research: Atmospheres* 104 (D8): 9423-9444.

Stanislas, L., and T. Peynot. 2015. "Characterisation of the Delphi Electronically Scanning Radar for Robotics Applications." *Australasian Conference on Robotics and Automation 2014.* Melbourne, Australia: Australian Robotics & Automation Association. 0-0.

Starik, S., and M. Werman. 2003. "Simulation of Rain in Videos." *Texture Workshop, ICCV* 2: 406-409.

Tong, X. L., E. A. Luke, and R. Smith. 2014. "Numerical Validation of A Near-Field Fugitive Dust Model for Vehicles Moving on Unpaved Surfaces." *Proceedings of the Institution of Mechanical Engineers, Part D.* Journal of Automobile Engineering. 747-757.

Velodyne. 2017. "velodynelidar.com." *velodynelidar.com.* January 1. Accessed October 6, 2017. http://velodynelidar.com/docs/datasheet/63-9229_Rev-F_Puck%20_Spec%20Sheet_Web.pdf.

Wallace, J. R., M. F. Cohen, and D. P. Greenberg. 1987. "A Two-Pass Solution to the Rendering Equation: A Synthesis of Ray Tracing and Radiosity Methods." *ACM* 21: 0-0.

Wang, B., G. D. Tong, and J. X. Lin. 2013. "Monte Carlo Simulation of Laser Beam Scattering by Water Droplets." *Fifth International Symposium on Photoelectronic Detection and Imaging.* International Society for Optics and Photonics. 89052.

Unit Conversion Factors

Multiply	By	To Obtain
atmosphere (standard)	101.325	kilopascals
bars	100	kilopascals
cubic feet	0.02831685	cubic meters
degrees (angle)	0.01745329	radians
degrees Fahrenheit	(F-32)/1.8	degrees Celsius
feet	0.3048	meters
hectares	1.0 E+04	square meters
inches	0.0254	meters
miles per hour	0.44704	meters per second
pounds (force) per square inch	6.894757	kilopascals
pounds (mass)	0.45359237	kilograms
yards	0.9144	meters

Acronyms

VANE	Virtual Autonomous Navigation Environment
ODOA	obstacle detection and avoidance
DoD	Department of Defense
RDE	Robotic Development Environment
ANVEL	Autonomous Navigation Virtual Environment Laboratory
UGV	unmanned ground vehicle
NDVI	normalized differential vegetation index
CCD	charge-coupled device
CMOS	complementary metal-oxide semiconductor
BRDF	the bidirectional reflectance distribution function
HTSG	He-Torrance-Sillion-Greenberg
VIS/NIR	visual near-infrared
HDRF	hemispherical reflectance distribution function
API	application programming interface
NREL	National Renewable Energy Laboratory
ECEF	earth-centered, earth-fixed
ENU	East-North-Up
TOF	time-of-flight
PM	particle matter

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE ***(DD-MM-YYYY)*** August 2020	**2. REPORT TYPE** Final	**3. DATES COVERED** ***(From - To)***

4. TITLE AND SUBTITLE Sensor and Environment Physics in the Virtual Autonomous Navigation Environment (VANE)	**5a. CONTRACT NUMBER**
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Justin T. Carrillo, Christopher T. Goodin, and Juan Fernandez	**5d. PROJECT NUMBER** 471937
	5e. TASK NUMBER 2
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Geotechnical and Structures Laboratory U.S. Army Engineer Research and Development Center 3909 Halls Ferry Road Vicksburg, MS 39180-6199	**8. PERFORMING ORGANIZATION REPORT NUMBER** ERDC/GSL TR-20-32
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) U.S. Army Corps of Engineers Washington, DC 20314-1000	**10. SPONSOR/MONITOR'S ACRONYM(S)**
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This report documents the physics models that are implemented in the Virtual Autonomous Navigation Environment (VANE), a sensor simulator that uses physics-based ray tracing to simulate common robotic sensors such as cameras, LiDAR, GPS, and automotive RADAR. The report will provide information about the underlying assumptions and implementation details regarding the physics models used in VANE simulations. These include surface reflectance and texture models, atmospheric models, weather effects, and sensor properties.

The purpose of this report is to provide information for VANE users, developers, and analysts who would like to use the VANE for sensor simulations.

15. SUBJECT TERMS

	Digital cameras	Automotive sensors
Automated vehicles	Global positioning system	Detectors – Simulation methods
Autonomous robots	Optical radar	

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT Unclassified	**b. ABSTRACT** Unclassified	**c. THIS PAGE** Unclassified	SAR	54	**19b. TELEPHONE NUMBER** ***(include area code)***