1 CAMERA FAULTS

1.1 Deflection and Displacement

Cameras on AVs are often calibrated to fixed positions and angles. Any misalignment (like bump) induced a shift or rotation, affecting the position and orientation of the camera on the AV. This model realistically captures these faults by perturbing the intrinsic and extrinsic parameters of the camera. The deflection and displacement of camera are represented as the combination of rotation and translation of the camera from its normal, calibrated position. The two faults are modeled as a 3D transformation matrix that modifies the camera's pose (position and orientation) in the world coordinate system. Specifically, For the camera projection matrix P, which transforms world coordinates X_w into pixel coordinates x_p:

$$x_p = PX_w \tag{1}$$

The modified projection matrix P' due to deflection is decomposed into a rotation matrix R_d and a translation vector t_d:

$$P' = T_d P; \quad T_d = \begin{bmatrix} R_d & t_d \\ 0 & 1 \end{bmatrix} \tag{2}$$

The projection of a deflected and displaced camera is:

$$x'_p = P'X_w = (T_d P)X_w \tag{3}$$

R_d represents the angular misalignment of the camera around its axes, which can be a perturbation in yaw ($R_y(\theta_y)$), pitch ($R_c(\theta_c)$), or roll ($R_o(\theta_o)$), each can be modeled as standard rotation matrices. t_d represents a small displacement along the x, y, or z axes.

$$R_y(\theta_y) = \begin{bmatrix} cos(\theta_y) & 0 & sin(\theta_y) \\ 0 & 1 & 0 \\ -sin(\theta_y) & cos(\theta_y) & 0 \end{bmatrix}; \quad t_d = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \tag{4}$$

1.2 Broken Len

The broken lens of cameras can affect the image distortion and light refraction properties. This model is modeled using a point spread function (PSF) [21], which describes how a point source of light spreads across the image due to len imperfections. The PSF due to lens breakage can be approximated as a Gaussian function with varying spread radius, the formula of the model for broken len is:

$$I_b = I_e \times H_{psf} + N \tag{5}$$

$$H_{psf}(x, y) = \frac{1}{2\pi\sigma^2} exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right); \quad N(x, y) \sim \mathcal{N}(0, \sigma^2) \tag{6}$$

where Ie is the original image captured by the camera without broken len and N represents additional disturbance such as irregular fracture fragments. σ represents the spread of light. The value of σ increases with more severe len breakage.

1.3 Len Brightness Change

To simulate the effect of brightness changes in a camera len due to a fault (such as a malfunctioning shutter or iris diaphragm), the model of len brightness change is built as the affects of brightness change in the pixel intensities of the captured images [4, 22], which accounts for the reduction in light reaching the sensor due to the component malfunctioning.

$$I_L = \alpha \cdot I_o(x, y, t) + \beta \cdot I_a(x, y, t) + \gamma \cdot I_g(x, y, t) + \eta \cdot \sigma(t) \tag{7}$$

where $I_o(x, y, t)$ is the original pixel intensity, $I_a(x, y, t)$ is the ambient light entering the lens to be exaggerated or diminished, $I_g(x, y, t)$ is the internal glare. $\eta \cdot \sigma(t)$ is a time-varying stochastic noise component. α is the brightness attenuation factor, determined by the degree of lens obstruction or diaphragm malfunction. β is the ambient light scaling factor, and γ is the glare intensity scaling factor, dependent on the severity of the component malfunctioning.

$$I_o(x, y, t) = k \cdot L_i(x, y, t) \cdot T(x, y, t) \tag{8}$$

$$I_a(x, y, t) = \int_\Omega L_a(u, v, t) \cdot A(x - u, y - v) dudv \tag{9}$$

$$I_g(x, y, t) = \int_\Omega L_n(u, v, t) \cdot G(x - u, y - v) dudv \tag{10}$$

$L_i(x, y, t)$ is the incident light intensity at the pixel (x, y) at time t, and $T(x, y, t)$ is the transmission factor of the len. k is a proportionality constant that accounts for the sensor's sensitivity and other camera-specific factors. $L_a(u, v, t)$ is the ambient light intensity at a point (u, v) at time t, and $A(x - u, y - v)$ is a spatial kernel that accounts for the spreading of ambient light over the image, representing the optical characteristics of how light enters the lens and diffuses across the sensor. $L_n(u, v, t)$ is the light incident at position (u, v) at time t, and $G(x - u, y - v)$ is the glare function, which models the scattering of light due to the optical effects of diffraction and internal reflection.

To model dynamic changes in brightness over time, the parameters α, β, γ can be made functions of time t:

$$\alpha(t) = \alpha_0 \cdot (1 + sin(wt + \phi)) \tag{11}$$

$$\beta(t) = \beta_0 \cdot (1 + sin(w't + \phi')) \tag{12}$$

$$\gamma(t) = \gamma_0 \cdot (1 + \delta(t)) \tag{13}$$

where w and w' are the temporal frequencies of the intermittent malfunctioning, and ϕ and ϕ' are the phase shifts of the intermittent malfunctioning. γ is a stochastic term that models random glare fluctuations.

1.4 Blur

To simulate how a camera's imaging and sensing are affected by blur, we can model the fault using a Gaussian blur kernel [14] and incorporate Gaussian noise [16] to represent real-world noise conditions. For the originally captured image $I_e(x, y)$, the blur for the camera is generated as:

$$I_b(x, y) = \sum_{i=-k}^{k} \sum_{j=-k}^{k} I_e(x - i, y - j) C(i, j) + N(x, y) \tag{14}$$

$$C(i, j) = \frac{1}{2\pi\sigma^2} e^{-\frac{i^2+j^2}{2\sigma^2}}; \quad N(x, y) \sim \mathcal{N}(0, \sigma^2) \tag{15}$$

where k is the kernel size and (x, y) pixel coordinates relative to the center of the kernel. $N(x, y)$ is the Gaussian noise added to the blurred image. σ is the standard deviation of the noise, which controls the intensity of the noise.

1.5 Internal Scatter

The internal scatters of camera fault are modeled as adding color noise to the image in the form of pixel-level distortions. Let I_e be the original image at pixel (x, y), with $R(x, y)$, $G(x, y)$, $B(x, y)$ being the red, green, and blue channel values, respectively. I_s is the image captured by the camera with internal scatter. The fault injection model of internal scatter is:

$$I_s(x,y) = \begin{bmatrix} R_s(x,y) \\ G_s(x,y) \\ B_s(x,y) \end{bmatrix} = \begin{bmatrix} R(x,y) + \alpha_s \cdot N_s(x,y) \\ G(x,y) + \alpha_g \cdot N_g(x,y) \\ B(x,y) + \alpha_b \cdot N_b(x,y) \end{bmatrix} \tag{16}$$

$$N_s(x,y) \sim \mathcal{N}(0, \sigma_s^2), \quad N_g(x,y) \sim \mathcal{N}(0, \sigma_g^2), \quad N_b(x,y) \sim \mathcal{N}(0, \sigma_b^2) \tag{17}$$

where $N_s(x, y), N_g(x, y), N_b(x, y)$ are random noise functions for each color channel, representing colored scatter noise. They can be modeled as Gaussian noise functions. $\alpha_s^2, \alpha_g^2, \alpha_b^2$ are scaling factors that control the intensity of the fault in each channel. $\sigma_s^2, \sigma_g^2, \sigma_b^2$ represent the standard deviations of the noise, indicating the severity of impact on each color channel.

1.6 Internal Dirt

To simulate the fault model where the imaging and sensing of the camera are affected by internal dirt accumulation, the model is built considering the area, absorption, refraction and scattering of dirt. I_d is the dirt area.

$$I_d = \int_\Omega f_t(w_i, w_0; \tau, \sigma_a, \sigma_s, g, n) \cdot L_i(w_i) dw_i \tag{18}$$

where $L_i(w_i)$ is the radiance from the direction w_i, and $f_t(w_i, w_0)$ represents the fraction of light from the direction w_i scattered to the outgoing direction w_0. $\tau, \sigma_a, \sigma_s, g, n$ are the absorption coefficient, scattering coefficient, parameter for the phase function, and refractive index, respectively, all of which are material properties of the contaminants.

1.7 Len Occlusion

The model of len occlusion is constructed by the polygonal mask, which is a balance of masked area and quality. I_u is the occlusion on the len and n represents the degree of occlusion. k is the number of pixels of the camera. The area of the polygonal mask is calculated using the shoelace formula.

$$I_u = (1 - \psi(M)) \cdot I_e + \psi(M) \cdot \psi(I_l), \quad M = \frac{|\Sigma_{i=1}^n (x_{i+1}y_i - x_i y_{i+1})|}{2|\Sigma_{i=1}^n (x_i y_i)(x_{i+1}y_{i+1})|} \tag{19}$$

where I_e is the original image. ψ is an unsupervised Cycle-GAN [24, 25] to scale the image based on the degree of occlusion. M is the polygonal mask and I_l is a random occluder pattern.

1.8 Overexposure

Overexposure occurs when the camera sensor receives too much light, causing bright areas in the image to lose detail, often leading to a "washed out" appearance [1]. This model captures the essential attributes of overexposure by excessive light, which leads to loss of detail, especially in bright areas. The model is defined by the the spatial frequency and ratio of signals that the image data can be transmitted. The overexposure area I_p of the camera is:

$$I_p = 2\pi \int_0^B log_2 \left(\frac{S(f)}{N(f)} + 1 \right) f df \tag{20}$$

$$S(f) = \frac{\mu}{\sigma}, \quad N(f) = \frac{MTF^2(f)}{NEQ(f_x, f_y)} \mu, \quad MTF(f) = \frac{C(f)}{C(0)}, \quad C(f) = \frac{V_{max} - V_{min}}{V_{max} + V_{min}} \tag{21}$$

where f is the spatial frequency and V is luminance. $S(f)$ is the signal power spectrum and $N(f)$ is the noise power spectrum. B is the Nyquist frequency. NEQ is the noise equivalent quanta. μ is the mean and σ is the standard deviation of the pixel values in a square patch.

1.9 Dust

The model of dust on the len of camera is designed based on transmission coefficient and diffraction principles, as dust obstructs and scatters incoming light. The image I_{sw} captured by the camera with dust on the len is:

$$I_{du}(x, y) = I_e(x, y) \times T(x, y) + S(x, y) \tag{22}$$

where $I_e(x, y)$ is the pixel (x, y) of the original image captured by the camera without dust on the len. $T(x, y)$ is a transmission coefficient map that represents the light attenuation due to dust coverage at the pixel (x, y). It ranges from 0 (full occlusion by dust) to 1 (no attenuation). $S(x, y)$ is the scattered light at pixel (x, y) caused by diffraction and scattering due to the dust particles. It introduces a form of Gaussian blur and brightness distortion around dust particles.

$$T(x, y) = 1 - \Sigma_{i=1}^{K} \alpha_i \cdot G(x - x_i, y - y_i; \sigma_i), \quad K \leq N \tag{23}$$

$$S(x, y) = \Sigma_{s=1}^{K} \beta_s \cdot G(x - x_s, y - y_s; \sigma_s), \quad K \leq N \tag{24}$$

where K is the number of dust particles and N is the number of pixels of the camera's image. α_i is the attenuation factor of the dust particle based on its size and density. $G(x - x_i, y - y_i; \sigma_i)$ is a Gaussian distribution centered at the dust particle location (x_i, y_i) with spread determined by σ_i which controls the size of the dust particle's effect. Meanwhile, β_i is the scattering intensity related to the dust particle's size and reflective properties. $G(x - x_s, y - y_s; \sigma_s)$ is a Gaussian function that spreads the scattered light around the dust location.

1.10 Rain

Rain lines act as random streaks of water on the lens, causing scattering, refraction, and partial occlusion of the camera's field of view. The shape of raindrops is not linear, under natural conditions, they usually exhibit a tilted shape [19]. Therefore, when simulating the effect of rainwater, the angle and shape changes of raindrops are introduced to simulate the shape of raindrops. Meanwhile, in order to make the raindrop effect more realistic, the transparency (or brightness) of raindrops are also adjusted [8]. The image I_r captured by the camera with raindrops on the len is:

$$I_r(x, y) = (1 - L_r(x, y)) \cdot I_e(x, y) + L_r(x, y) \cdot (t_r \cdot I_e(x, y) + (1 - t_r) \cdot N_r(x, y)) \tag{25}$$

$$t_r(x, y) \sim U(t_{min}, t_{max}), \quad N_r(x, y) \sim \mathcal{N}(0, \sigma_r) \tag{26}$$

where $I_e(x, y)$ is the original pixel intensity of the image captured by the camera without raindrops on the len. $N_r(x, y)$ is a random noise factor simulating refraction-induced distortion. t_r is a transparency factor, simulating the partial occlusion of the camera's view due to the rain streaks. $L_r(x, y)$ is the rain line mask.

$$L_r(x, y) = \Sigma_{i=1}^{n_r} \mathcal{H}\left(\frac{(y - y_i) - tan(\theta_r^i)(x - x_i)}{l_r^2}\right) \tag{27}$$

$$l_r \sim U(l_{min}, l_{max}), \quad \theta_r \sim U(\theta_{min}, \theta_{max}) \tag{28}$$

$L_r(x, y)$ represents the spatial distribution of the rain streaks on the lens. Each streak can be modeled as a linear segment on the image with a length l_r and angle θ_r. (x_i, y_i) is the starting position of a rain line. $\mathcal{H}(f)$ is a Heaviside function (or step function) that returns 1 if f is within the rain line length, and 0 otherwise, controlling the spatial extent of each rain line.

1.11 Snow

Snow grains on a camera can block parts of the lens and scatter light, causing occlusion, diffusion, and reduction in image clarity [7]. The fault injection model of snow grains on the camera is modeled by modifying the image formation process, considering both occlusion and diffusion caused by snow particles. Let I_e the original image captured by the camera without snow grains, the image I_{sw} captured by the camera with snow grains is:

$$I_{sw} = (1 - R_{a,\lambda}(x,y)) \cdot G(I_e(x,y)) + R_{a,\lambda}(x,y) \cdot N_{sw} \tag{29}$$

$$R_\lambda = exp\left(-K_0 b\sqrt{\gamma_\lambda D_0}\right), \quad N_{sw} \sim \mathcal{N}(\mu_{sw}, \sigma_{sw}^2) \tag{30}$$

where R_λ is the spectral albedo, at the wavelength λ. γ_λ is the absorption coefficient of snow grain which depends on the imaginary part of the refraction index. b represents the shape factor and accounts for various types of grains (ranges from 4.53 for spheres to 3.62 for tetrahedral grains). D_0 is the optical diameter of the snow grain and K_0 is the escape function and depends mainly on the incident lighting conditions. $G(I_e(x,y))$ is Gaussian intensity to simulate the light scattering due to the snow grain's semi-transparent properties based on snow grain size. N_{sw} is the noise term representing the scattered light due to snow's reflective and refractive properties. μ_{sw} is the mean intensity of the scattered light and σ_{sw}^2 is the variance of the scattered light intensity.

1.12 Mist

In the presence of mist on the len, its vision is within a large number of water droplets. During the AV's driving, the light from headlamps is attenuated by the dual phenomena of absorption and diffusion [6], which leads to characterizing mist on the len by means of an extinction coefficient k (equal to the sum of the absorption and diffusion coefficients). The predominant phenomenon is diffusion which acts to deviate light rays from their initial direction. The image capturing I_m of the camera with mist on the len is modeled as:

$$I_m(x,y) = I_e(x,y) \cdot L(x,y) + T(x,y) \cdot (1 - L(x,y)) \tag{31}$$

$$L = L_0 e^{-kd} + L_f(1 - e^{-kd}), \quad T(x,y) = e^{-\beta \cdot d(x,y) \cdot V(x,y)}, \quad V(x,y) = -\frac{1}{k} ln(\epsilon) \tag{32}$$

where $I_e(x,y)$ is the original pixel intensity at (x,y). L is the transmission function that accounts for the luminance attenuation due to mist. T is the ambient light scattering caused by mist, which is proportional to the intensity of surrounding light sources. L is the apparent luminance of the object with distance d and intrinsic luminance L_0. L_f is the luminance and k is the extinction coefficient of the weather. β is the attenuation coefficient related to mist density. ϵ is the mist density.

1.13 Ice

The model of ice on the len is similar to the mist model, both of which are caused by condensation water, but ice has stronger occlusion, light reflection and absorption, and weaker refraction [3]. In addition, for the same len surface, the thickness of ice varies. The image capturing I_{ce} of the camera with mist on the len is modeled as:

$$I_{ce}(x,y) = I_e(x,y) \cdot L_c(x,y) + S(x,y) \cdot (1 - L_c(x,y)) + A(x,y) \tag{33}$$

$$L_c(x,y) = exp(-\alpha \cdot \Delta_c(x,y)), \quad A(x,y) = G_{\sigma(x,y)} \times I_{ce}, \quad S(x,y) = \begin{bmatrix} 1 & k_1\Delta_c(x,y) \\ k_2\Delta_c(x,y) & 1 \end{bmatrix} \tag{34}$$

where $L_c(x,y)$ is the the distortion of light transmitted through the ice at pixel (x,y) and $S(x,y)$ accounts for the refraction of light caused by the irregular ice surface. $A(x,y)$ is the additive blur due to light scattering, representing how much light is diffused by the ice. This term can be modeled as a convolution of the clear image with a Gaussian blur kernel $G_{\sigma(x,y)}$, where the standard deviation $\sigma(x,y)$ depends on the local ice thickness $\Delta_c(x,y)$.

1.14 White Balance Shift

White balance in imaging adjusts the colors to make white objects appear neutral under different lighting conditions, which typically relate to the color temperature of the light source. A shift in white balance can cause the entire image to appear too warm (yellow/orange) or too cool (blue) [27]. The model of white balance shift is to simulate the alteration of color temperature in the image.

$$I_f(x,y) = clip(I_{wb}(x,y), 0, 255), \quad I_{wb}(x,y) = \begin{bmatrix} I_R(x,y) \cdot s_R(T_{wb}) \\ I_G(x,y) \cdot s_G(T_{wb}) \\ I_B(x,y) \cdot s_B(T_{wb}) \end{bmatrix} \tag{35}$$

$$clip(v, v_{min}, v_{max}) = \begin{cases} v_{min}, & v \leq v_{min}, \\ v, & v_{min} < v < v_{max}, \\ v_{max}, & v \geq v_{max}. \end{cases} \quad T_{wb} = T_0 + \Delta T \tag{36}$$

$$S_R(T_{wb}) = \left(\frac{T_{wb}}{T_0}\right)^{-\alpha_R}, S_G(T_{wb}) = \left(\frac{T_{wb}}{T_0}\right)^{-\alpha_G}, S_B(T_{wb}) = \left(\frac{T_{wb}}{T_0}\right)^{-\alpha_B} \tag{37}$$

White balance is described in terms of color temperature T. To inject a shift (positive for a warmer shift and negative for a cooler shift), we introduce a parameter ΔT, which modifies the color temperature away from the ideal (reference) white balance setting T_0. The RGB values need to be adjusted according to the faulty color temperature. The scaling factors $S_R(T_{wb}), S_G(T_{wb}), S_B(T_{wb})$ are to apply white balance correction is to scale the red, green, and blue channels by factors that depend on the inverse color temperature. These factors adjust how the camera sensor interprets light of different wavelengths, where $\alpha_R, \alpha_G, \alpha_B$ are sensitivity parameters to different color channels. Typically, α_R is higher than α_G, reflecting how the red channel is more affected by changes in warm temperatures, and α_B higher for cooler ones. Since extreme changes in color temperature may cause channel values to exceed the displayable range (0–255 for 8-bit images), the final step is to normalize the pixel values to fit within the acceptable range.

2 LIDAR

2.1 Deflection

The deflection model of LiDAR is introduced by the deflection of the vertical to the direct direction and scan angle direction. When the deflection of the vertical direction is perpendicular to the scan angle direction, the resulting vertical error will be negligible. Therefore, the induced vertical error ranges between a maximum value when the scan angle direction is parallel to the deflection of the vertical direction and a minimum value when the deflection direction is perpendicular to the scan angle direction [12]. The magnitudes of the rotations are defined as ξ and η components of the deflection of the vertical, and the resulting R_G is formalized as:

$$R_G = \begin{bmatrix} cos(\eta) & sin(\xi)sin(\eta) & sin(\xi)cos(\eta) \\ 0 & cos(\xi) & -sin(\xi) \\ -sin(\eta) & cos(\xi)sin(\eta) & cos(\xi)cos(\eta) \end{bmatrix} \tag{38}$$

2.2 Displacement

The displacement model of LiDAR is defined by the grid mean approximation method and the triangular grid approximation, which have been applied experimentally to generate reference data for 3D data in the spatial domain [9, 13, 17]. The grid mean approximation method includes grid point errors and forms grids on the x and y planes based on irregularly distributed spatial data. Thereafter, the z coordinates of the data in the grid are averaged to determine the representative point of each grid. The grid mean approximation uses a multiple regression analysis technique and calculates the displacement of a structure using structural information such as strain, stress, displacement, and z-coordinates.

$$P_j(x, y, z) = (\varepsilon, Z_j), \quad Z_j = \frac{1}{n_j}\Sigma_{i=1}^{n_j}\Sigma_{j=1}^{m} Z_{ji}, \quad \varepsilon = N_1\varepsilon_1 + N_2\varepsilon_2 + N_3\varepsilon_3 \tag{39}$$

where Z_{ji} represents the Z coordinate value of the i-th coordinate data included in the j-th space. The point P_j is set as reference data in the center of the grid. The shape functions N_1, N_2, N_3 are calculated through natural coordinates.

2.3 Beam Loss

The model of beam loss is simulated as the degradation of beam density and intensity. The beam density refers to the number of laser pulses emitted per unit area or time [23]. Over time, due to wear and tear, or aging, the beam density may reduce. The intensity of each laser pulse reduces over time due to the aging of the laser diode and optical components [26]. This reduction impacts the effective sensing range and resolution of LiDAR.

$$P_{bl}(t) = P_o(t) D_0 I_0 e^{-(\alpha+\beta)t} \tag{40}$$

where α is the decay rate on density and β is that on intensity. P_o is the original point cloud of the LiDAR without fault injection. D_0 is initial beam density and I_0 is the initial beam intensity.

2.4 Line Fault

Gaussian noise is a realistic approximation of random noise encountered in electronics due to thermal fluctuations, shot noise, and component degradation [10]. The zero mean reflects the unbiased nature of the noise, while the variance reflects the severity of the malfunction. The key of internal component fault injection for LiDAR is to represent its point cloud with added Gaussian noise to simulate the erratic behavior of malfunctioning components. This noise will be added to both the range and intensity measurements for each point.

$$P_{lf}^i(t) = (x_t^i + \mathcal{N}(0, \sigma_X^2), y_t^i + \mathcal{N}(0, \sigma_Y^2), z_t^i + \mathcal{N}(0, \sigma_Z^2), I_i + \mathcal{N}(0, \sigma_I^2)) \tag{41}$$

where (x_t^i, y_t^i, z_t^i) are the spatial coordinates of the $i - th$ point at time t. I_i is the intensity value of the returned signal at the $i - th$ point. $\sigma_X^2, \sigma_Y^2, \sigma_Z^2$ are the variances of the Gaussian noise in the spatial dimensions, representing the level of distortion in the distance and position measurements. σ_I^2 is the variance of the noise in the intensity measurement, representing errors in the received signal strength.

2.5 Electromagnetic Interference

The key of electromagnetic interference model for LiDAR is to model the weakening of the LiDAR signal as it encounters strong electromagnetic waves [20]. The fault injection model of electromagnetic interference is as follows:

$$P_{ei}(t) = P_o(t) \cdot e^{-d_{A-E}(t)} + \mathcal{N}(0, \sigma_E(t)) \tag{42}$$

$$d_{A-E}(t) = \sqrt{(x_{AV} - x_{EM})^2 + (y_{AV} - y_{EM})^2 + (z_{AV} - z_{EM})^2} \tag{43}$$

where P_o is the original point cloud of the LiDAR without electromagnetic interference. $d_{A-E}(t)$ is the distance between the AV and the electromagnetic source at time t. $\mathcal{N}(0, \sigma_E(t))$ is the standard deviation of the noise, which increases with the proximity to the electromagnetic source.

(x_{AV}, y_{AV}, z_{AV}) is the coordinate of the AV at time t, and (x_{EM}, y_{EM}, z_{EM}) is the is the coordinate of the electromagnetic source.

2.6 Crosstalk

LiDAR is subject to shot noise due to external laser sources, and Poisson processes are widely used in shot-noise modeling for systems with random and independent signal emissions [15, 18]. Since the emission of signals from NPC vehicles' LiDARs are random and occur independently across time, the crosstalk injection realized by Poisson distribution can effectively capture the random occurrence of these noise points.

$$P_{ct}(N(t) = n) = \frac{(\lambda t)^n e^{-\lambda t}}{n!} \tag{44}$$

where $N(t)$ the number of interfering points (shot-noise points) observed at time t. λ is the rate of the Poisson process, representing the average number of interfering points generated per unit time or distance. This rate depends on the number of NPC vehicles using LiDAR and their relative proximity to the AV.

2.7 Rain and Snow Pollution

During laser propagation, snow particles and raindrops can cause scattering, absorption, and multiple refractions of laser signals, reducing the power of the laser reaching the receiver and resulting in signal attenuation [2]. This fault injection model is built as:

$$P_{lf} = P_o \frac{\xi \cdot C(\alpha, \beta)}{R^2} exp\left(-2\int_{r=0}^{R} \varphi(r)dr\right), \quad C(\alpha, \beta) = \int_0^\alpha \int_0^\beta R^2 sin(\beta) d\beta d\alpha \tag{45}$$

where P_o is the original point cloud of the LiDAR without fault injection. R is the radius of LiDAR's emitted laser. ξ is the efficiency of laser reflection, α is the horizontal field of view angle and β is the vertical field of view angle. φ is the LiDAR signal's extinction coefficient.

2.8 Strong Light Interference

As the intensity of strong light increases, more detector units are occupied by erroneous signals from the strong light, leading to the exponential decay in point-cloud density and a lower density of valid point returns [5]. The fault injection model of strong light interference is constructed by the calculation of the normalized multiple-scattering [11] (all considered orders) LiDAR returning per unit receiver aperture area.

$$P_{sl} = \chi_s(z) \cdot \mathcal{A}^2, \quad \chi_s(z) = \frac{1}{(H+z)^2} \frac{c\tau}{2} \sigma_s P(\pi) exp(-2\sigma_e z) \tag{46}$$

where τ is the pulse length and $c\tau$ is the spatial resolution of the LiDAR. χ_s is the single scattering return per unit area and $\mathcal{A}$ is the aperture area. z is the range inside the cloud and H is the distance to the base of the cloud. σ_s is the cloud-scattering coefficient and σ_e is the extinction coefficient. P is the scattering phase function.

REFERENCES

[1] [n. d.]. *Understanding Overexposure Underexposure.* Retrieved September 16, 2024 from https://shotkit.com/overexposure-underexposure/

[2] Emmanuel Alozie, Abubakar Abdulkarim, Ibrahim Abdullahi, Aliyu D Usman, Nasir Faruk, Imam-Fulani Yusuf Olayinka, Kayode S Adewole, Abdulkarim A Oloyede, Haruna Chiroma, Olugbenga A Sowande, et al. 2022. A review on rain signal attenuation modeling, analysis and validation techniques: Advances, challenges and future direction. *Sustainability* 14, 18 (2022), 11744.

[3] Kundan Biswas and Vivek Joshi. 2021. *Prediction of Vehicle Headlamp Condensation Phenomenon Using Computational Fluid Dynamics.* Technical Report. SAE Technical Paper.

[4] Vladimir Brajovic. 2004. Brightness perception, dynamic range and noise: a unifying model for adaptive image sensors. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, Vol. 2. IEEE, II–II.

[5] Weichen Dai, Shenzhou Chen, Zhaoyang Huang, Yan Xu, and Da Kong. 2022. LiDAR intensity completion: Fully exploiting the message from LiDAR sensors. *Sensors* 22, 19 (2022), 7533.

[6] Pierre Duthon, Michèle Colomb, and Frédéric Bernardin. 2019. Light transmission in fog: The influence of wavelength on the extinction coefficient. *Applied Sciences* 9, 14 (2019), 2843.

[7] Xiangsuo Fan, Dachuan Xiao, Qi Li, and Rui Gong. 2024. Snow-CLOCs: Camera-LiDAR Object Candidate Fusion for 3D Object Detection in Snowy Conditions. *Sensors* 24, 13 (2024), 4158.

[8] Kshitiz Garg and Shree K Nayar. 2006. Photorealistic rendering of rain streaks. *ACM Transactions on Graphics (TOG)* 25, 3 (2006), 996–1002.

[9] Pelagia Gawronek, Maria Makuch, Bartosz Mitka, and Tadeusz Gargula. 2019. Measurements of the vertical displacements of a railway bridge using TLS technology in the context of the upgrade of the polish railway transport. *Sensors* 19, 19 (2019), 4275.

[10] Craig Glennie and Derek D Lichti. 2010. Static calibration and analysis of the Velodyne HDL-64E S2 for high accuracy mobile scanning. *Remote sensing* 2, 6 (2010), 1610–1624.

[11] Robin J Hogan. 2008. Fast lidar and radar multiple-scattering models. Part I: Small-angle scattering using the photon variance–covariance method. *Journal of the Atmospheric Sciences* 65, 12 (2008), 3621–3635.

[12] Christopher Jekeli. 2019. Deflections of the vertical from full-tensor and single-instrument gravity gradiometry. *Journal of Geodesy* 93 (2019), 369–382.

[13] DS Kang, HM Lee, Hyo Seon Park, and I Lee. 2007. Computing method for estimating strain and stress of steel beams using terrestrial laser scanning and FEM. *Key Engineering Materials* 347 (2007), 517–522.

[14] Yu-Qi Liu, Xin Du, Hui-Liang Shen, and Shu-Jie Chen. 2020. Estimating generalized gaussian blur kernels for out-of-focus image deblurring. *IEEE Transactions on circuits and systems for video technology* 31, 3 (2020), 829–843.

[15] Torben Neumann and Franz Kallage. 2023. Simulation of a Direct Time-of-Flight LiDAR-System. *IEEE Sensors Journal* 23, 13 (2023), 14245–14252.

[16] Tuan-Anh Nguyen, Won-Seon Song, and Min-Cheol Hong. 2010. Spatially adaptive denoising algorithm for a single image corrupted by Gaussian noise. *IEEE Transactions on Consumer Electronics* 56, 3 (2010), 1610–1615.

[17] J Notbohm, A Rosakis, S Kumagai, S Xia, and G Ravichandran. 2013. Three-dimensional Displacement and Shape Measurement with a Diffraction-assisted Grid Method. *Strain* 49, 5 (2013), 399–408.

[18] Klaus Pasquinelli, Rudi Lussana, Simone Tisa, Federica Villa, and Franco Zappa. 2020. Single-photon detectors modeling and selection criteria for high-background LiDAR. *IEEE Sensors Journal* 20, 13 (2020), 7021–7032.

[19] Martin Roser and Andreas Geiger. 2009. Video-based raindrop detection for improved image registration. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops.* IEEE, 570–577.

[20] Alexander V Ryzhkov, Dusan S Zrnic, Alexander V Ryzhkov, and Dusan S Zrnic. 2019. Polarization, Scattering, and Propagation of Electromagnetic Waves. *Radar Polarimetry for Weather Observations* (2019), 1–18.

[21] Yoav Shechtman. 2020. Recent advances in point spread function engineering and related computational microscopy approaches: from one viewpoint. *Biophysical reviews* 12, 6 (2020), 1303–1309.

[22] HL Sneha. 2010. Pixel intensity histogram characteristics: basics of image processing and machine vision. *Viitattu* 26 (2010), 2018.

[23] Jorge Souto, José Luis Pura, and Juan Jiménez. 2018. Thermal and mechanical issues of high-power laser diode degradation. *MRS Communications* 8, 3 (2018), 995–999.

[24] Hao Tang, Dan Xu, Gaowen Liu, Wei Wang, Nicu Sebe, and Yan Yan. 2019. Cycle in cycle generative adversarial networks for keypoint-guided image generation. In *Proceedings of the 27th ACM international conference on multimedia.* 2052–2060.

[25] Yuan Yuan, Siyuan Liu, Jiawei Zhang, Yongbing Zhang, Chao Dong, and Liang Lin. 2018. Unsupervised image super-resolution using cycle-in-cycle generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops.* 701–710.

[26] Mark S Zediker and Erik P Zucker. 2022. High-power diode laser technology XX: a retrospective on 20 years of progress. *High-Power Diode Laser Technology XX* 11983 (2022), 1198302.

[27] Yiying Zhang, Yangfei Gao, Yeshen He, Yancui Shi, and Kun Liang. 2017. Research on the color temperature & white balance for multimedia sensor. *Procedia Computer Science* 107 (2017), 878–884.