Article

# Airborne Waveform Lidar Simulator Using the Radiative Transfer of a Laser Pulse 

Minsu Kim<br>KBR, Contractor to U.S. Geological Survey (USGS) Earth Resources Observation and Science (EROS) Center, Sioux Falls, SD 57198, USA; minsukim@contractor.usgs.gov

Received: 2 May 2019; Accepted: 13 June 2019; Published: 15 June 2019


#### Abstract

An airborne lidar simulator creates a lidar point cloud from a simulated lidar system, flight parameters, and the terrain digital elevation model (DEM). At the basic level, the lidar simulator computes the range from a lidar system to the surface of a terrain using the geomatics lidar equation. The simple computation effectively assumes that the beam divergence is zero. If the beam spot is meaningfully large due to the large beam divergence combined with high sensor altitude, then the beam plane with a finite size interacts with a ground target in a realistic and complex manner. The irradiance distribution of a delta-pulse beam plane is defined based on laser pulse radiative transfer. The airborne lidar simulator in this research simulates the interaction between the delta-pulse and a three-dimensional (3D) object and results in a waveform. The waveform will be convoluted using a system response function. The lidar simulator also computes the total propagated uncertainty (TPU). All sources of the uncertainties associated with the position of the lidar point and the detailed geomatics equations to compute TPU are described. The boresighting error analysis and the 3D accuracy assessment are provided as examples of the application using the simulator.


Keywords: airborne lidar simulator; radiative transfer; total propagated uncertainty

## 1. Introduction

Airborne lidar point clouds and the derived elevation products are essential geospatial datasets, along with airborne or spaceborne image data. Accordingly, the efforts to ensure high-quality lidar data became increasingly important [1,2]. Much research contributed to the quality assurance (QA) of lidar data [3-5]. The direct georeferencing geomatics equation is the backbone of lidar accuracy research. An airborne lidar simulator that computes point cloud solutions by the forward modeling of airborne lidar parameters using the geomatics equation is a useful tool in lidar research. A lidar simulation software can simulate the effect of lidar parameters on the point cloud as well as various advanced lidar research. An airborne lidar simulator consists of a laser module, a scanner module, a global navigation satellite system, an inertial measurement unit (GNSS/IMU) module, the flight parameters, and the environmental parameters. The environmental parameters include the optical property of the propagating medium and the 3D terrain, including 3D objects. The laser module is characterized by pulse repetition frequency (PRF), pulse width, the beam divergence angle at the $e^{-2}$ level, and the pulse distribution function. The scanner determines the overall scan pattern, such as the linear oscillating pattern from a Galvo mirror scanner and the circular pattern from a rotating prism. The scan frequency is also an important factor to determine the along-track and across-track lidar point distribution. The field of view (FOV) of a scanner along with the sensor above ground level (AGL) altitude determines the swath width. The DEM component describes the terrain over which the lidar system scans.

Several airborne lidar simulator software programs were created using mainly direct georeferencing [6-9]. The simple computation assumes that the beam divergence is effectively zero.

Thus, it assumes that the light interaction occurs only at the point where the beam center hits the terrain surface. In case the beam edge touches the target, but the beam center evades the target as shown in Figure 1, the simple georeferencing will not find a target hit. Instead, it will find the point at the background terrain.


Figure 1. Interaction between a laser delta-pulse and a cube.
If the beam spot is substantially large, only the waveform approach based on the radiative transfer of a laser pulse can describe the problem properly. An advanced simulator addresses the beam divergence using a simplified Gaussian beam spot model [10].

The approach presented here simulates the lidar scanner using the waveform from the solution of the radiative transfer equation of a laser pulse propagation, rather than a semi-analytical beam spot model. The governing lidar waveform equation is given by the following [11-16]:

$$
\begin{equation*}
S(t)=Q \Sigma \Omega \frac{c}{2 n} \int_{z(t)} d \boldsymbol{r} \beta(z, r) E_{E}(z, r) E_{R}(z, r) \tag{1}
\end{equation*}
$$

The general lidar equation models the returned optical power, $S(t)$. Here, $Q$ is the laser pulse energy, $\sum$ is the receiver pupil area, $\Omega$ is the solid angle subtended from a target to the receiver, $c$ is the speed of light, and $n$ is the refractive index. The governing equation models the optical return as an integral of the two irradiance distributions, emitter irradiance, $E_{E}$, and the receiver irradiance, $E_{R}$, in the presence of the interaction function $\beta$. The integration using Equation (1) is performed over a normal cross-section plane at distance $z(t)$ after the pulse propagation time $t$. The vector $r$ represents the two-dimensional (2D) position on the cross-section plane. As the emitted laser pulse propagates, the cross-section area expands due to the beam divergence and the light is attenuated by the medium, creating weaker emitter irradiance.

The concept of receiver irradiance may not be intuitive because the receiver does not send out any light. However, the receiver sensitivity distribution can be considered as if it is a radiance distribution. The sensitivity function experiences the same type of expansion and attenuation defined by the field of view. Indeed, the governing lidar equation calls for a different view of how lidar return is detected. In Figure 1, the narrow green beam represents the laser irradiance and the wider blue beam represents the receiver irradiance.

The interaction function, $\beta(z, r)$, for the first green return arrow represents the lidar return from the ambient medium, the second arrow represents the target reflection, and the third arrow represents the topographic surface reflection (Figure 1). The irradiance distribution of a delta-pulse beam plane
provides the chance of evaluating the time-varying interaction along the delta-pulse and the target surface interaction line as the beam propagates downward. Thus, the essential component in the radiative transfer-based waveform approach is how to compute irradiance. The theory and solutions are described in the following sections.

## 2. Theories and Modeling

### 2.1. Lidar Waveform Solution by Radiative Transfer Theory

Compared to the simple radiometric solution, the radiative transfer theory of a laser pulse and the numerical solution is quite challenging. Thus, only the essential components of the theory are briefly described here. Refer to References [11-16] for the full theories and the numerical solutions [16]. To solve the governing equation of a lidar system, the irradiance solutions are needed in Equation (1). The irradiance as a function of 2 D position vector, $r$, on the beam cross section at a given distance, $z$, is an integration of the radiance distribution over the hemisphere.

$$
\begin{equation*}
E(\boldsymbol{r})=\int d \omega(\boldsymbol{n}) I(\boldsymbol{r}, \boldsymbol{n}) \tag{2}
\end{equation*}
$$

Here, the radiance is integrated over infinitesimal solid angle, $d \omega(\boldsymbol{n})$, centered on the direction vector $\boldsymbol{n}$. Thus, we need a radiance distribution function $I(\boldsymbol{r}, \boldsymbol{n})$, which is obtained as a solution of the laser pulse propagation based on the radiative transfer theory.

$$
\begin{equation*}
\boldsymbol{n} \cdot \nabla I(\boldsymbol{\rho}, \boldsymbol{n})=-c I(\boldsymbol{\rho}, \boldsymbol{n})+\int_{4 \pi} I\left(\boldsymbol{\rho}, \boldsymbol{n}^{\prime}\right) \beta\left(\boldsymbol{n}^{\prime} \rightarrow \boldsymbol{n}\right) d \omega\left(\boldsymbol{n}^{\prime}\right) \tag{3}
\end{equation*}
$$

This is a 3D radiative transfer equation (RTE) that describes the net radiance balance at a given position and direction. The value $\rho$ is a 3 D position vector, c is an attenuation coefficient, and $\beta$ is a volume scattering function of the medium. Since there is no analytical solution for Equation (3) in 3D, it is reduced to a radiance distribution on a 2D plane perpendicular to the beam direction using small angle approximation because the volume scattering function of the natural air or water medium is strongly forward-oriented.

$$
\begin{equation*}
\frac{\partial}{\partial z} I\left(z, \boldsymbol{r}, \boldsymbol{n}_{t}\right)+\boldsymbol{n}_{t} \cdot \nabla_{t} I\left(z, \boldsymbol{r}, \boldsymbol{n}_{t}\right)=-c I\left(z, \boldsymbol{r}, \boldsymbol{n}_{t}\right)+b_{s} \int I\left(z, \boldsymbol{r}, \boldsymbol{n}_{t}^{\prime}\right) x_{s}\left(\left|\boldsymbol{n}_{t}-\boldsymbol{n}_{t}^{\prime}\right|\right) d \boldsymbol{n}_{t}^{\prime} . \tag{4}
\end{equation*}
$$

Here, $\boldsymbol{n}_{t}$ is the 2D direction vector defined on the 2D beam cross section and its norm is small according to the small angle approximation, $b_{s}$ is a small angle forward-scattering coefficient, and the small angle forward-scattering phase function, $x_{s}$, is defined as a function of difference between incoming and scattered 2D direction vectors. The radiance solution is derived via an analytical Fourier transform.

$$
\begin{equation*}
\widetilde{I}(z, \boldsymbol{k}, \boldsymbol{p})=\widetilde{I}\left(z_{0}, \boldsymbol{k}, \boldsymbol{p}+\boldsymbol{k}\left(z-z_{0}\right)\right) \exp \left(-\int_{z_{0}}^{z}\left\{c-2 \pi b_{s} \cdot \widetilde{x}_{s}(\boldsymbol{p}+\boldsymbol{k}(z-\varsigma))\right\} d \varsigma\right) . \tag{5}
\end{equation*}
$$

Here, $\boldsymbol{k}, \boldsymbol{p}$ is the conjugate frequency variable of the $\boldsymbol{r}, \boldsymbol{n}_{t}$ and $\widetilde{x}_{s}$ is the analytical Fourier transform of $x_{s}$. Converting Equation (5) into an inverse Fourier transform and plugging it into the Equation (2) yields the following:

$$
\begin{equation*}
E(z, \boldsymbol{r})=\int \widetilde{I}(0, \boldsymbol{k}, z \boldsymbol{k}) \exp (-i \boldsymbol{k} \cdot \boldsymbol{r}) \exp \left[-z^{\prime}(\boldsymbol{r}) a_{s}-a_{b s}(z k) z\right] d \boldsymbol{k} \tag{6}
\end{equation*}
$$

Here, $a_{s}$ is the effective absorption coefficient and $a_{b s}$ is the effective attenuation coefficient due to forward scattering. Two things are needed to solve Equation (6). First, the source laser pulse radiance
distribution in the Fourier transform, and, second, the effective attenuation coefficient. Assuming a Gaussian beam the radiance and its Fourier Transform are as follows:

$$
\begin{equation*}
\widetilde{I}(0, \boldsymbol{k}, z \boldsymbol{k})=\frac{1}{(2 \pi)^{2}} \exp \left(-k^{2}\left(1+\frac{z}{n H_{s}}\right)^{2} r_{0}^{2}\right) \tag{7}
\end{equation*}
$$

Here, $r_{0}$ is effective radius defined by multiplication of half of the beam divergence angle, $\Theta$, and slant altitude, $H_{s}$. The effective attenuation coefficient, $a_{b s}$, due to the small angle forward-scattering $b_{s}$ is given as follows using the analytical forward-scattering phase function model:

$$
\begin{equation*}
a_{b s}(z k)=b_{s}-\frac{\alpha b_{s}}{z k} \ln \left(\left(\frac{\alpha}{z k}\right)^{-1}+\left(\left(\frac{\alpha}{z k}\right)^{-2}+1\right)^{0.5}\right) \tag{8}
\end{equation*}
$$

Here, $\alpha$ is a volume scattering function shape parameter by Dolin [13]. The effective attenuation coefficient, $a_{b s}$, characterizes the contribution from small angle forward scattering to the attenuation of spatial harmonics of the spatial frequency, $k$, in the initial distribution of radiance within a light beam when propagated to the distance, $z$, in the medium. Finally, using Equations (7) and (8) the irradiance solution of Equation (6) becomes the following:

$$
\begin{gather*}
E(z, r)=\frac{\exp \left(-a_{s} z^{\prime}(r=)\right)}{2 \pi} \int f(k ; z, r) d k \\
f(k ; z, r)=J_{0}(k r) k \exp \left(-\left(\frac{k^{2}}{4}\right) P^{2}-a_{b s}(z k) z\right) \tag{9}
\end{gather*}
$$

Here the effective radius is defined as $P=\left(\frac{\Theta}{2}\right) \cdot\left(H_{s}+\frac{z}{n}\right)$ and $J_{0}$ is the 0 -th order first kind Bessel function and $z^{\prime}\left(r_{=}\right)$is the geometrical distance of the slant-incident beam projected along the coordinate parallel to the direction of pulse propagation. Equation (9) is a final irradiance solution for a point at $r$ away from the center on a circular plane at a distance, $z$, for a Gaussian laser pulse. Once another irradiance solution for the receiver is calculated in the same manner, the lidar waveform is formed by applying two irradiances in Equation (1) and by computing the numerical integration. Complete details of the theories can be found in [16].

These equations are most generic to be used for laser pulse propagation through any medium. This level of sophistication is critical when the laser pulse propagates through substantially dense medium, like water for bathymetric lidar. The high absorption and scattering by the natural water medium must be properly handled using these equations. However, if a lidar simulation is only for the topographic objects, the equations can be dramatically simplified. Equation (9) will be a concentric 2D Gaussian defined by the beam spot radius, which is determined by the slant sensor altitude, $H_{s}$, and the full beam divergence, $\Theta$, at the $e^{-2}$ level. Assuming the instantaneous field of view (IFOV) is significantly larger than the beam divergence, the two irradiance multiplication in the governing Equation (1) is still a Gaussian distribution, mainly determined by the beam divergence as follows:

$$
\begin{equation*}
E_{E}(z, r) E_{R}(z, r) \propto \exp \left(-\left(\frac{r}{\left(H_{S} \cdot \Theta / 2\right) / \sqrt{2}}\right)^{2}\right) \tag{10}
\end{equation*}
$$

The delta-pulse first touches down at a tiny line segment A (Figure 2). The length of the interaction line segment increases to the arrow direction and then decreases toward the corner.

$$
\begin{equation*}
S(t) \propto \int_{r_{2}(t)}^{r_{1(t)}} \beta(z(t), \boldsymbol{r}) E_{E}(z(t), \boldsymbol{r}) E_{R}(z(t), \boldsymbol{r}) d \boldsymbol{r} \tag{11}
\end{equation*}
$$



Figure 2. Interaction between a laser delta-pulse and the object. The 3D view on the left and the 2D top-view on the right.

The interaction function, $\beta(z(t), r)$ in this case, is the target surface reflectance. Assuming the homogeneous reflectance property of the target surface, the value is constant all over the target surface. The radiative transfer-based irradiance solution, $E_{E} E_{R}$, is also a 2D circular plane. However, the plane hits the target surface with a non-zero incident angle to make it a slant plane. Thus, the waveform Equation (1) will be non-zero only along the interaction line segment between the two position vectors $r_{1}$ and $r_{2}$, which results in the 2D integral in a one-dimensional (1D) line integral. The solution can be obtained only via numerical solution because the position of the target is over a random relative position and orientation of the target. The waveform solutions are demonstrated in the Results section.

### 2.2. Lidar Direct Georeferencing

While the radiative transfer-based waveform approach is the advanced component of the airborne lidar simulator, the direct georeferencing is the backbone of the simulator. A direction vector of an emitted laser pulse defined in sensor body frame (SBF) is defined first, then, by multiplying with the lidar range, $\rho$, to a point, $P$, on the surface, the position vector in the SBF is determined.

$$
\begin{equation*}
\boldsymbol{X}_{P}^{S B F}=\rho \cdot d_{P}^{S B F} \tag{12}
\end{equation*}
$$

Next, the position vector in SBF $X_{P}^{S B F}$ is converted to the position in the IMU body frame (IBF), $X_{P}^{I B F}$, using the following relationship:

$$
\begin{equation*}
\boldsymbol{X}_{P}^{I B F}=\boldsymbol{R}_{S B F}^{I B F} \cdot \boldsymbol{X}_{P}^{S B F}+\boldsymbol{X}_{O_{S B F}}^{I B F} \tag{13}
\end{equation*}
$$

where $X_{O_{S B F}}^{I B F}$ is the lever-arm vector to the SBF origin in the IBF and $R_{S B F}^{I B F}$ is a rotation matrix that transforms $\boldsymbol{X}_{P}^{S B F}$ to $\boldsymbol{X}_{P}^{I B F}$ and is defined using three boresighting angles, $\omega, \varphi, \kappa$, as follows:

$$
\boldsymbol{R}_{S B F}^{I B F}=\left[\begin{array}{ccc}
\cos \varphi \cos \kappa & \sin \omega \sin \varphi \cos \kappa-\cos \omega \sin \mathcal{\kappa} & \cos \omega \sin \varphi \cos \kappa+\sin \omega \sin \kappa  \tag{14}\\
\cos \varphi \sin \kappa & \sin \omega \sin \varphi \sin \kappa+\cos \omega \cos \kappa & \cos \omega \sin \varphi \sin \kappa-\sin \omega \cos \kappa \\
-\sin \varphi & \sin \omega \cos \varphi & \cos \omega \cos \varphi
\end{array}\right]
$$

Next, the position vector, $\boldsymbol{X}_{P}^{I B F}$, is transformed into a new position vector, $\boldsymbol{X}_{P}^{L T F}$, in a local tangential frame (LTF), as follows:

$$
\begin{equation*}
\boldsymbol{X}_{P}^{L T F}=\boldsymbol{R}_{I B F}^{L T F} \cdot \boldsymbol{X}_{P}^{I B F} . \tag{15}
\end{equation*}
$$

An LTF is defined in reference to a global geodetic frame, which follows a coordinate axes convention of a North-East-Down (NED) reference frame. The orientation of the IBF, with respect to the LTF, is described by three IMU angles, roll $(R)$, pitch $(P)$, and heading $(H)$, as follows:

$$
\boldsymbol{R}_{I B F}^{L T F}=\left[\begin{array}{ccc}
\cos P \cos H & \sin R \sin P \cos H-\cos R \sin H & \cos R \sin P \cos H+\sin R \sin H  \tag{16}\\
\cos P \sin H & \sin R \sin P \sin H+\cos R \cos H & \cos R \sin P \sin H-\sin R \cos H \\
-\sin P & \sin R \cos P & \cos R \cos P
\end{array}\right] .
$$

Along with the position of the GNSS antenna phase center, the lever-arm distances between the IMU origin and the antenna can be used to calculate the position of the O_IBF (which is the same as O_LTF). Using the latitude, $\phi$, longitude, $\lambda$, and ellipsoid height, $h$, from O_LTF, it is possible to form a rotation matrix that transforms an arbitrary position vector in LTF to the position vector in the Earth Centered Earth Fixed (ECEF) coordinate:

$$
\boldsymbol{R}_{L T F}^{E C E F}=\left[\begin{array}{ccc}
-\sin \varphi \cos \lambda & -\sin \lambda & -\cos \varphi \cos \lambda  \tag{17}\\
-\sin \varphi \sin \lambda & \cos \lambda & -\cos \varphi \sin \lambda \\
\cos \varphi & 0 & -\sin \varphi
\end{array}\right]
$$

In addition to the rotation matrix, the geodetic position $(\varphi, \lambda, h)$ of $O_{-}$LTF itself can be transformed into the position in the ECEF coordinate using the following relationship:

$$
\boldsymbol{R}_{O_{L T F}}^{E C E F}=\left[\begin{array}{c}
\left(r_{N}+h\right) \cos \varphi \cos \lambda  \tag{18}\\
\left(r_{N}+h\right) \cos \varphi \sin \lambda \\
{\left[\left(1-e^{2}\right) r_{N}+h\right] \sin \varphi}
\end{array}\right], r_{N}=\frac{a}{\sqrt{1-e^{2} \sin ^{2} \varphi}}
$$

where $a$ is the semi-major axis, $e$ is the eccentricity, and $r_{N}$ is the position in ECEF coordinates. Now the position vector can be solved in ECEF coordinates, as follows:

$$
\begin{equation*}
\boldsymbol{X}_{P}^{E C E F}=\boldsymbol{R}_{L T F}^{E C E F} \cdot \boldsymbol{X}_{P}^{L T F}+\boldsymbol{X}_{O_{L T F}}^{E C E F} \tag{19}
\end{equation*}
$$

Finally, the ECEF coordinates are converted to the map projection, such as the Universal Transverse Mercator (UTM) or a state plane.

### 2.3. Total Propagated Uncertainty (TPU)

The TPU in this simulator is calculated using the following direct georeferencing equation:

$$
\begin{equation*}
\boldsymbol{X}_{P}^{L T F}=\boldsymbol{R}_{I B F}^{L T F} \cdot\left(\boldsymbol{R}_{S B F}^{I B F} \cdot \rho \cdot \boldsymbol{d}_{P}^{S B F}+\boldsymbol{X}_{O_{S B F}^{I B F}}^{I B F}\right)+\boldsymbol{X}_{O_{I B F}}^{L T F} \tag{20}
\end{equation*}
$$

This is the equation that describes the position of a point $P(x, y, z)$ in the LTF coordinate. The direct georeferencing equation that calculates the position in LTF indeed includes all the major sources of errors, such as boresighting, scan angle, GNSS position, and IMU angles. Regarding TPU, it is called "total" because multiple error sources are combined and it is called "propagated" because the errors are propagated through Equation (20). The TPU as a statistical uncertainty is computed based on the linear approximation and by ignoring the covariances between multiple parameters.

Currently, the assumption of negligible covariances is accepted in the lidar community. As the TPU is still in its infancy, no major lidar manufacturers (e.g., Teledyne Optech, Hexagon Leica, Riegl) currently provide direct georeferencing proprietary software that computes TPU. The only known exception is the CZMIL bathymetric lidar system that computes TPU with support from the US Army Corps of Engineers (USACE). The TPU model in this research utilizes 14 parameters. Thus, the number of covariance combination is $14 \times 14 / 2$, considering the symmetry of the covariance matrix. While most covariances are likely to be negligible, potential exceptions of non-negligible covariances could be the position and orientation parameters. However, even the Applanix IMU systems do not have covariance specifications due to the measurement difficulties, but it is not impossible. Numerous U.S. Geological Survey (USGS) internal lidar accuracy assessments indicate that the uncertainties from airborne lidar data in terms of smooth surface precision [2] are very close to the computed TPU based on negligible covariances, which justifies the assumption that the negligible covariance is acceptable.

To formulate the TPU computation equations, Equation (20) is rewritten as follows:

$$
\left[\begin{array}{l}
x  \tag{21}\\
y \\
z
\end{array}\right]=\boldsymbol{R}_{I B F}^{L T F} \cdot\left(\boldsymbol{R}_{S B F}^{I B F} \cdot \rho \cdot \boldsymbol{d}_{P}^{S B F}+\left[\begin{array}{c}
x_{0} \\
y_{0} \\
z_{0}
\end{array}\right]\right)+\left[\begin{array}{c}
x_{1} \\
y_{1} \\
z_{1}
\end{array}\right] .
$$

The TPU in $x$ position in LTF is given by the following equation, which is derived from Equation (21) (The TPU for y and z components is also computed in the same manner):

$$
\begin{gather*}
\sigma_{x}^{2}=\left(\frac{\partial x}{\partial x_{0}} \sigma_{x_{0}}\right)^{2}+\left(\frac{\partial x}{\partial y_{0}} \sigma_{y_{0}}\right)^{2}+\left(\frac{\partial x}{\partial z_{0}} \sigma_{z_{0}}\right)^{2}+\left(\frac{\partial x}{\partial x_{1}} \sigma_{x_{1}}\right)^{2}+\left(\frac{\partial x}{\partial y_{1}} \sigma_{y_{1}}\right)^{2}+\left(\frac{\partial x}{\partial z_{1}} \sigma_{z_{1}}\right)^{2}+, \\
\left(\frac{\partial x}{\partial \rho} \sigma_{\rho}\right)^{2}+\left(\frac{\partial x}{\partial \theta} \sigma_{\theta}\right)^{2}+\left(\frac{\partial x}{\partial R} \sigma_{R}\right)^{2}+\left(\frac{\partial x}{\partial P} \sigma_{P}\right)^{2}+\left(\frac{\partial x}{\partial H} \sigma_{H}\right)^{2}+\left(\frac{\partial x}{\partial \omega} \sigma_{\omega}\right)^{2}+\left(\frac{\partial x}{\partial \varphi} \sigma_{\varphi}\right)^{2}+\left(\frac{\partial x}{\partial \kappa} \sigma_{\kappa}\right)^{2} . \tag{22}
\end{gather*}
$$

To solve Equation (22), two sets of data are needed. First, the uncertainty, $\sigma$, of each component, which is given as a priori standard deviation. The second is the partial derivative, which is the Jacobian matrix element.

The Jacobian matrix elements for lever-arm displacement and position vector of IBF origin can be derived as follows:

$$
\left[\begin{array}{lll}
\frac{\partial \boldsymbol{X}_{P}^{L T F}}{\partial x_{0}} & \frac{\partial \boldsymbol{X}_{P}^{L T F}}{\partial y_{0}} & \frac{\partial \boldsymbol{X}_{P}^{L T F}}{\partial z_{0}}
\end{array}\right]=\boldsymbol{R}_{I B F}^{L T F},\left[\begin{array}{ccc}
\frac{\partial \boldsymbol{X}_{P}^{L T F}}{\partial x_{1}} & \frac{\partial \boldsymbol{X}_{P}^{L T F}}{\partial y_{1}} & \frac{\partial \boldsymbol{X}_{P}^{L T F}}{\partial z_{1}} \tag{23}
\end{array}\right]=\boldsymbol{I} .
$$

The Jacobian matrix elements for lever-arm displacement and position vector of IBF origin are derived as follows:

$$
\begin{equation*}
\frac{\partial \boldsymbol{X}_{P}^{L T F}}{\partial \rho}=\boldsymbol{R}_{I B F}^{L T F} \cdot \boldsymbol{R}_{S B F}^{I B F} \cdot \boldsymbol{d}_{P}^{S B F}, \quad \frac{\partial \boldsymbol{X}_{P}^{L T F}}{\partial \theta}=\rho \cdot \boldsymbol{R}_{I B F}^{L T F} \cdot \boldsymbol{R}_{S B F}^{I B F} \cdot \frac{\partial \boldsymbol{d}_{P}^{S B F}}{\partial \theta} \tag{24}
\end{equation*}
$$

The Jacobian matrix elements for boresighting and the IMU angles are derived as follows:

$$
\begin{align*}
& \frac{\partial \boldsymbol{X}_{P}^{L T F}}{\partial \chi}=\rho \cdot \boldsymbol{R}_{I B F}^{L T F} \cdot \frac{\partial \boldsymbol{R}_{S B F}^{I B F}}{\partial \chi} \cdot \boldsymbol{d}_{P}^{S B F},(\chi=\omega, \varphi, \kappa) .  \tag{25}\\
& \frac{\partial \boldsymbol{X}_{P}^{L T F}}{\partial \chi}=\rho \cdot \frac{\partial \boldsymbol{R}_{I B F}^{L T F}}{\partial \chi} \cdot \boldsymbol{R}_{S B F}^{I B F} \cdot \boldsymbol{d}_{P}^{S B F},(\chi=R, P, H) . \tag{26}
\end{align*}
$$

The analytical derivative solutions in Equations (24)-(26) required at the next stage are straight forward, thus they are not described here. The TPU computed via the equations in this section computes the uncertainty associated to the position vector in LTF. This means when the TPU associated with the lidar point cloud in the commonly used 2D map projection coordinate, such as the UTM or state plane, is needed, then the additional uncertainty that occurs in the rest of the coordinate transform needs to be included. Those additional procedures convert the point from LTF to the point in the map projection. Since the major sources of error were already handled in the equations so far, only the uncertainty ( $\sigma^{\prime}$ ) caused by the projection conversion will be added in the following manner.

$$
\begin{equation*}
\sigma_{x} \leftarrow \sqrt{\sigma_{x}^{2}+\sigma_{x}^{\prime 2}}, \sigma_{y} \leftarrow \sqrt{\sigma_{y}^{2}+{\sigma^{\prime}}_{y}^{2}}, \sigma_{z} \leftarrow \sqrt{\sigma_{z}^{2}+\sigma_{z}^{\prime 2}} \tag{27}
\end{equation*}
$$

### 2.4. Airborne Lidar Simulator

An airborne lidar simulator was developed to implement direct georeferencing, target simulation, and the radiative transfer-based waveform modeling. The simulator consists of three main components, as follows:
(1) Laser module: PRF [Hz], pulse width [ns], beam divergence [mrad], pulse function;
(2) Scanner module: Scanner type, scan frequency [Hz], FOV [deg];
(3) Flight parameters: AGL [m], speed [kt], flight pattern, swath overlap.

The pulse function is the radiance distribution function (e.g., Gaussian or Top-hat) on the cross section of a pulse fired at PRF (pulse repetition frequency). The typical scanner types are liner Galvo-mirror, circular, and Palmer scanner. The scanning frequency is an important factor determining the lidar point cloud point density in PPM (points $/ \mathrm{m}$ ) or PPSM (points $/ \mathrm{m}^{2}$ ).

The point density is an important simulation condition because the lidar point cloud positional accuracy is constrained by the point density. The following are the descriptive formulae:

$$
\begin{aligned}
\mathrm{PPSM} & =\frac{P R F}{v \cdot S W}, S W=2 \cdot A G L \cdot \tan (F O V / 2), \\
\mathrm{PPM} \_\mathrm{al} & =\left(2 \cdot F_{S} / v\right), \mathrm{PPM} \_\mathrm{ac}=\mathrm{PRF} /\left(2 \cdot F_{S} \cdot S W\right) .
\end{aligned}
$$

Along track linear point density (PPM_al) is determined by the scanner frequency $\left(F_{s}\right)$ and the speed $(v)$. Across track density (PPM_ac) is related to the swath width (SW). Although the parameters affect the point density in an intuitive manner, creating an optimal set of parameters is a tricky issue. For instance, if we want higher PPSM, then the increase of PRF might come as a first option. However, if PRF is simply increased, it will result in the increased across track density with unchanging along track density, which will cause uneven point distribution. It is obvious that numerous cases like this can be investigated for an optimal setting. All these multiple parameters need to be carefully adjusted to achieve desired evenly distributed point density.

The airborne lidar simulation starts from determining the total number of pulses, because it is the foundation of the computation flow (Figure 3). Using several related parameters, the SW is determined and then it is multiplied by another input, the "Length to Width" ratio, which determines the flight distance. The distance divided by speed gives the flight time. Eventually the total number of pulses is obtained by the PRF times flight time.


Figure 3. Computation flow of the airborne lidar simulator.
The second step is to create an IMU and GNSS data array. Although any arbitrary IMU and GNSS position data can be generated, the practical strategy is to use a real SBET file (or other form of position and orientation data) and sample two sets of data out of a straight north flight segment: $\left(R_{1}, P_{1}, H_{1}\right)$ and $\left(R_{2}, P_{2}, H_{2}\right)$. The data come in a low temporal resolution (e.g., SBET in 200 Hz ). Thus, the time and $(R, P, H)$ data need to be interpolated to match the high-resolution time array (usually hundreds of KHz ) of the laser pulses. Several different flight patterns can be selected, including parallel, parallel opposite, cross, slant cross, etc. The following modifications in heading angles (Figure 4) are needed to accommodate the flight pattern.

The point clouds in Figure 4 demonstrate the resultant swaths for two examples (cross pattern with circular scanner and slant-cross pattern with linear scanner).


Figure 4. Preparation of orientation of simulated flight lines.
The GNSS position array is prepared for four preset flight patterns in Figure 5, where the x-axis and y-axis represent ground coordinates in meters. The coordinates are prepared in LTF. The swath boundary of the two swaths is illustrated in red and white. The GNSS position corresponds to the dashed line located in the center of each swath. This illustration is based on a $30 \%$ swath overlap ratio.


Figure 5. Preset flight patterns of the airborne lidar simulator.
The next step is to create a direction vector of the propagating pulse in the SBF. The scan angle $(\theta)$ array is prepared first using scanner type, $F_{s}, F O V$, and the scan angle offset, $(\Delta \theta)$. Then the direction vector in SBF for the Galvo-mirror and circular scanner is given, respectively, as follows:

$$
\begin{gathered}
d_{P}^{S B F}=[0, \sin \theta, \cos \theta] \\
\boldsymbol{d}_{P}^{S B F}=[\sin (F O V / 2) \cdot \cos \theta, \sin (F O V / 2) \cdot \sin \theta, \cos (F O V / 2)] .
\end{gathered}
$$

The final step is to estimate a range to the DEM surface. Unlike the active real lidar that measures the range, the simulator cannot measure the distance. The simulator can only estimate the range to the DEM surface by following the computation procedure described in the direct georeferencing section above, as follows: Starting from the direction vector in SBF $\rightarrow$ computing a position in SBF by introducing unknown range $\rho \rightarrow$ transforming the SBF position into IBF position and translating by a lever-arm displacement $\rightarrow$ transforming the position in IBF into the position in LTF using ( $R, P$, $H)$ angles. The solution for the position where the laser pulse hits the target can be computed in two methods. One approach is the analytical solution when a specific geometry of the target yields a forward solution. The other approach is a generic optimization using DEM to get the range solution.

Once the beam center interaction point is obtained, it is refined into the waveform solution using the radiative transfer-based irradiance distribution described in the above section. Several examples are presented to demonstrate how the waveform-based solution is different compared to the simple georeferencing of the beam center. The example shown in Figure 6 uses a cube and a beam spot size comparable to the cube dimension, which creates a non-negligible interaction effect. The upper panel in Figure 6 illustrates a sequence of interaction lines between the delta-pulse and the cube surface along the arrow direction. As pulse 1 propagates downward and a little toward the right side, the interaction line touches down on the left beam edge first, and then the interaction line makes a transition to the right side to create a full circle interaction area. Thus, the z-coordinate of the lidar point is exactly the altitude of the cube top surface, which is shown in the P1 waveform in Figure 7.


Figure 6. A cube target interacts with an incoming laser pulse at different locations. The 3D point cloud on the right side is colored by the intensity, which represents the relative energy given by the waveform integral in Figure 7. The color scheme follows the typical "rainbow" color table of the lidar point cloud visualization so that the yellowish color represents higher values and the bluish color represents lower values.



Figure 7. The beam interaction examples for 3 more points and the waveform near the target. The maximum intensity of each waveform corresponds to the $z$-altitude of the lidar point computed by direct georeferencing.

The estimated relative reflected energy of the pulse and the pulse center altitude for each point are as follows: $1.0(1.0 \mathrm{~m})$ for point $1,0.08(1.043 \mathrm{~m})$ for point $2,0.54(0.990 \mathrm{~m})$ for point 3, and 0.14 $(0.957 \mathrm{~m})$ for point 4.

Pulse 2 hits the upper-left edge as it is moving in a westerly downward direction. Thus, it hits on the right edge of the beam circle first and makes a transition toward the left. As it hits the small area of the corner, the interaction area is a small wedge giving only $8 \%$ of the relative energy, which means the total reflected energy, by integrating the low intensity, waveform is about $8 \%$, compared to the full circle interaction of point 1. Additionally, the computed altitude is 4.3 cm higher than the cube surface because the short duration of the interaction occurs when the beam center is still above the surface before moving down to the cube surface level. The rest of the interpretation can be understood in a similar manner. Point 3 interacts in the upper half circle area, resulting in $54 \%$ of relative energy, and the computed altitude is about 1 cm below the surface. The point 4 beam propagates in a westerly downward direction, making the first hit at the bottom edge of the small arc of the circle. The whole interaction area is a small arc area resulting in a relative intensity of only $14 \%$. Considering the interaction position is the left arc of the circle, the center of the beam is already lower than the top surface. That is why the altitude (z-component) of the computed lidar point position is 95.7 cm , which is lower than the cube top altitude of 1 m . This means the computed lidar point is suspended in air at 4.3 cm lower than the top surface. In fact, the laser pulse reflection occurs exactly on the surface, which is at a 1 m altitude. However, the waveform modeling simulates the pulse-target interaction accurately and simulates the situation that a real lidar system experiences.

A lidar system detects a waveform peak, but it has no idea on exactly which part of the cross-sectional area of a beam the light reflection occurs. Thus, it assumes that the interaction occurs at the beam center. Obviously, this error occurs because the radiative transfer and waveform-based rigorous computation approach is used. However, this error is exactly what must be simulated. As a result, the target will have slightly larger apparent dimension; in other words, a target enlargement effect. If the simple georeferencing was used, the cube hit position will be on the side wall of the cube, which does not happen in a real lidar system. Of course, if the beam divergence is very small and the sensor AGL is low, the target enlargement effect will be minimized accordingly. Finally, the simulator computes TPU according to the equations described above.

## 3. Results

Several examples of the airborne lidar simulator applications are presented here.

### 3.1. Boresighting Error Effect

The boresighting angles are the angular offsets between the SBF and IBF. A sensor optical model defines the 3D Cartesian coordinate system SBF. The coordinate axis of an IMU system is usually printed on the cover as arrows, and the lidar axis is aligned to the IMU axis as close as possible, but the boresighting (fine tuning) is required. The IMU system bridges between the "subjective" sensor coordinate to the "objective" local real-world coordinate system by telling the ( $R, P, H$ ) angles. Thus, if there are any misalignment angles, $(R, P, H)$ must be corrected by those misalignment (boresighting) angles. Frequently, those misalignment angles are called "roll, pitch, and heading". However, they are reserved terms for IMU angles. Strictly speaking boresighting angle means the angular offset between the SBF and IBF coordinates. The three angle sets $(\omega, \varphi, \kappa)$ are the angular offset defined by rotating $x$, $y$, and $z$ axes, respectively. The most commonly used convention is that the $x$-axis points to the nose of an aircraft, the $y$-axis points to the right-wing, and the $z$-axis points downward nadir. Thus, rotating the $x$-axis (axis that points the nose) results in the net effect such as a body roll, and so forth. Thus, the $\omega$ boresighting angle will effectively increase the IMU roll ( R ) angle by $\omega$, and the IMU ( $\mathrm{P}, \mathrm{H}$ ) angle adjustment in the same manner by $(\varphi, \kappa)$.

The data for boresighting correction is obtained by flying multiple swaths in different flight directions. If the perfect boresighting angles are found, the overlapped area by multiple flights
needs to be aligned. Thus, boresighting angles are iteratively optimized until the relative differences between multiple virtual planes become minimal. The airborne lidar simulator can achieve this via visual demonstration.

The net effect of the boresighting error is illustrated in Figure 8. The point cloud in the middle shows perfect boresighting, thus the point cloud from two swaths makes a perfect overlap. Figure 8a and b demonstrate the effect of an incorrect $\omega, \varphi$ angle for two parallel swaths to the same direction, while Figure $8 \mathrm{c}, \mathrm{d}$ show the effect for two parallel swaths to the opposite direction. Figure 8 shows the overestimation, which means the optimized boresighting angles are greater than the true value.


Figure 8. The effect of boresighting angle error: (a) Overestimated $\omega$ for parallel same direction, (b) overestimated $\varphi$ for parallel same direction, (c) overestimated $\omega$ for parallel opposite direction, (d) overestimated $\varphi$ for parallel opposite direction.

For the overestimated $\omega$ for parallel same direction swaths, the point cloud from both swaths shifts to the left side, as shown in Figure 8a, while the underestimated $\omega$ will send both swaths to the right side. For the overestimated $\omega$ for parallel opposite direction swaths, the point cloud from each swath is shifted to the opposite (one left and the other right) direction, as shown in Figure 8c. If underestimated, each swath will change direction but the end result will be the same, which is to be separated, as shown in Figure 8c.

For the overestimated $\varphi$ for the parallel same direction swaths, the point cloud from both swaths shifts forward, as shown in Figure 8b, while the underestimated $\varphi$ will send both swaths to the rear. For the overestimated $\varphi$ for parallel opposite direction swaths, the point cloud from each swath is shifted to the opposite (one forward and the other backward) direction, as shown in Figure 8c. If underestimated each swath will change direction but the end result will be the same, which is to be separated, as shown in Figure 8d.

As demonstrated in Figure 8, the net result of boresighting error cannot be explained by the boresighting angle error itself. The flight pattern plays a major role to amplify or lessen the boresighting angle error effect because the relative location of the target from one swath and the other swath is different. This is an example that a airborne lidar simulator can reproduce.

### 3.2. TPU Estimation

Using Equations (21)-(26), the TPU $\left(\sigma_{x}, \sigma_{y}, \sigma_{z}\right)$ associated with every lidar point is computed. The variation of the TPU values comes from the partial derivative, which is related to the magnitude of local variables. Consider the first term in TPU Equation (22) as an example. Although the uncertainty $\left(\sigma_{x_{0}}\right)$ is constant, the partial derivative of a Jacobian element $\frac{\partial x}{\partial x_{0}}$ can have a wide range of values depending on the variables used in the computation. Thus, the dynamic range of TPU is determined by the dynamic range of variables, such as lidar range, scan angle range, IMU angle range, and so on.

Figure 9 shows all the uncertainty sources and the lidar system parameters. The lidar system parameters on the right are the common input. Figure 9a represents horizontal TPU and Figure 9b represents vertical TPU using the high precision IMU system with circular scanner. The range of horizontal TPU is $14-15 \mathrm{~cm}$ and the vertical TPU range is $4-5 \mathrm{~cm}$, which is much smaller than the horizontal TPU. The overall magnitude of the TPU is determined mainly by the AGL ( 1 km ) and minor contributions come from the rest of the system specification. The minimal range $(1 \mathrm{~cm})$ is due to the IMU angle variation. Figure 9c represents the horizontal TPU and Figure 9d represents the vertical TPU using the low precision IMU system with a linear scanner. The range of horizontal TPU is $37-47 \mathrm{~cm}$ and the vertical TPU range is $1-15 \mathrm{~cm}$. The high TPU using low precision IMU is obvious compared to the low TPU using high precision IMU. Figure 9a,b use a circular scanner that has a constant off-nadir angle. Figure $9 \mathrm{c}, \mathrm{d}$, however, use a linear scanner with the off-nadir angle between $-20^{\circ}$ and $20^{\circ}$. Thus, in addition to the raised TPU baseline caused by the low precision IMU, the large off-nadir scan angle range adds a substantial TPU range.


Figure 9. The simulated TPU and the uncertainty input components from the lidar simulator GUI: (a) Horizontal TPU with a circular scanner, (b) vertical TPU with circular scanner, (c) horizontal TPU with linear scanner, (d) vertical TPU with linear scanner. The corresponding parameters on graphical user interface.

It is reasonable to assume that most of the uncertainties are constant numbers, at least during the flight time. However, the GNSS positional uncertainties ( $\sigma_{x_{1}}, \sigma_{y_{1}}, \sigma_{z_{1}}$ ) can be quite dynamic depending on the satellite conditions. Thus, this situation was simulated by creating a temporary short spike in $\left(\sigma_{x_{1}}, \sigma_{y_{1}}, \sigma_{z_{1}}\right)$. Figure 10 shows when the temporary poor position dilution of precision (PDOP) occurs in one swath, while the other swath is in good GNSS condition. The left panel in Figure 10, with a linear galvo-mirror scanner, shows the quick rise and fall of the GNSS uncertainty, while the right panel, with a circular scanner, shows a different pattern. The TPU provides a quality check of the lidar
data. When the abnormal TPU spike area is identified, a decision can be made about recollecting data over this area.


Figure 10. Horizontal TPU with temporary poor position dilution of precision (PDOP) with a linear (left) and circular (right) scanner.

### 3.3. Accuracy Assessmenr of 3D Point Cloud

The lidar simulator is an important research tool for 3D accuracy assessment. For example, consider a parking lot with painted lines on the pavement, as shown in Figure 11a. A common practice in performing a check point survey is to place a GNSS rover on the intersection point of two painted lines. For the 3D accuracy assessment, the intersection position $(x, y, z)$ in the airborne lidar point cloud must be identified. Once identified, the difference from the survey point yields an error vector $(\Delta x, \Delta y, \Delta z)$. Using multiple error vectors, a statistical estimate of the 3D accuracy $\left(\sigma_{x}, \sigma_{y}, \sigma_{z}\right)$ can be made. However, the position identification from the airborne point cloud is difficult, mainly because of the low point density. Thus, the usual practice is to ignore horizontal accuracy and find the location in the airborne point cloud using the surveyed horizontal position and interpolate the $z$-value for that position. The difference against the surveyed $z$-value is used to calculate the root mean square error in z ( RMSEz ).


Figure 11. Simulated intensity point cloud on the painted lines on the parking lot: (a) Parking lot paint line for 3D accuracy assessment, (b) simulated intensity point cloud of the 2 m line, (c) simulated intensity point cloud of the 6 m line.

Identifying the intersection point from two crossing lines is challenging. Figure 11b,c illustrates the random cases of lidar point cloud intensity over the modeled painted lines with the circular scanner
at 9 PPSM, which is the USGS quality level 1 category. The color intensity ranges from yellow (full spot hit), green, cyan, and blue (minimal hit at the line edge). Intensity is used because the elevation of the painted lines and the background pavement is not distinguishable. The background dotted lines are the painted lines with 20 cm width. Only the simulator, however, can show the true painted lines.

Figure 11b simulates a 2 m long line and Figure 11c simulates a 6 m long line. Depending on the location in a swath, the point density, and various other conditions, some cross lines will be under sampled, e.g., the upper right in Figure 11b. The 6 m line example in Figure 11c demonstrates that the number of points is enough to describe the cross line properly.

An airborne lidar simulator is a useful tool in addressing the required conditions (line length, width, point density, and intensity contrast) of the crossing lines for a 3D accuracy assessment. About 200 cross painted lines were distributed in the two-swath overlap area. As demonstrated in Figure 11, to pinpoint the intersection point requires an unrealistically large point density. Thus, directly identifying a conjugate point is not a correct approach. Instead, 3D line regression modeling is needed so that the intersection point can be determined from two regression lines. The difference ( $\Delta x, \Delta y, \Delta z$ ) is computed against the true intersection point. Using a large number of error vectors, a statistical estimate of 3D accuracy $\left(\sigma_{x}, \sigma_{y}, \sigma_{z}\right)$ can be made. The uncertainty plot (Figure 12b) illustrates the uncertainty of $x$ and $y$ error $\left(\sigma_{x}, \sigma_{y}\right)$ as a function of painted line length. As the length of the painted line increases, the uncertainty becomes smaller. Figure 12a,c represents the ( $\Delta x, \Delta y$ ) distribution for the 2 m and 6 m lines, respectively, and the uncertainty $\sigma$ values are 3 cm and 1.5 cm , respectively. The research results (like Figure 12b) obtained from the simulation will provide a guideline for a minimum line length to meet the uncertainty requirement in the 3D accuracy assessment. For instance, if the required uncertainty associated to locate a conjugate survey point in the airborne point cloud is 2 cm , then the length of the painted line must be longer than 4 m .


Figure 12. The point location uncertainty using airborne lidar simulator.

## 4. Discussion

An airborne lidar simulator was developed using a rigorous radiative transfer theory of a laser pulse. The simulator allows the realistic simulation of a laser pulse with a realistic beam divergence angle. Thus, it can reproduce the various interaction conditions between the laser pulse and a target. The simulator also computes the TPU associated to each lidar point. The lidar simulator also has the capability to perform target detectability research. Additionally, the requirement conditions for the 3D accuracy assessment can be investigated using this simulator, as demonstrated in the example showing the intensity point cloud from the painted line over the pavement. To the best of the author's knowledge, an airborne lidar simulator combining these rigorous concepts has not been available previously. Further lidar applications research will be performed in the future.

Funding: This research was funded by USGS National Geospatial Program (NGP) 3D Elevation Program (3DEP).
Acknowledgments: The author acknowledges the support from Jason Stoker and Cindy Thatcher from 3DEP, Greg Stensaas from the USGS Earth Resource Observation and Science (EROS) Center. Seonkyung Park, United Support Services, a contractor to EROS, contributed to the software development. Any use of trade, firm, or product names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

Conflicts of Interest: The author declares no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

## References

1. American Society for Photogrammetry and Remote Sensing (ASPRS). Vertical Accuracy Reporting for Lidar, version 1.0; ASPRS: Bethesda, MA, USA, 2004.
2. US Geological Survey. Lidar Base Specifrication; USGS National Geospatial Program (NGP); USGS Publications Warehouse: Reston, VA, USA, 2018. Available online: https://pubs.usgs.gov/tm/11b4 (accessed on 7 May 2019).
3. Skaloud, J.; Lichti, D. Rigorous approach to bore-sight self-calibration in airborne laser scanning. ISPRS J. Photogramm. Remote Sens. 2006, 61, 47-59.
4. Habib, A.; Kersting, A.; Bang, K.; Lee, D. Alternative Methodologies for the Internal Quality Control of Parallel LiDAR Strips. IEEE Trans. Geosci. Remote Sens. 2010, 48, 221-236. [CrossRef]
5. ASPRS. Digital Elevation Model, Technologies and Applications: The DEM Users Manual, 3rd ed.; Maune, D., Nayegandhi, A., Eds.; ASPRS: Bethesda, MA, USA, 2019.
6. Lohani, B.; Reddy, P.; Mishra, R. Airborne altimetric lidar simulator: An education tool, International archives of the photogrammetry. Remote Sens. Spat. Inf. Sci. 2006, XXXVI, 179-183.
7. Kim, S.; Min, S.; Kim, G.; Lee, I.; Jun, C. Data simulation of an airborne lidar system. In Proceedings of the SPIE Defense, Security, and Sensing, Orlando, FL, USA, 2 May 2009; pp. 1-10.
8. Bechtold, S.; Hofle, B. HELIOS: A multi-purpose lidar simulation framework for research, planning and training of laser scanning operations with airborne, ground-based mobile and stationary platforms. In ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences; XXIII ISPRS Congress: Prague, Czech Republic, 2016.
9. Lovell, J.; Jupp, D.; Newnham, G.; Coops, N.; Culvenor, D. Simulation study for finding optimal lidar acquisition parameters for forest height retrieval. For. Ecol. Manag. 2005, 214, 398-412. [CrossRef]
10. Kukko, A.; Hyyppa, J. Small-footprint Laser Scanning Simulator for System Validation, Error Assessment, and Algorithm Development. Photogram. Eng. Remote Sens. 2009, 75, 1177-1189. [CrossRef]
11. Kim, M.; Kopilevich, Y.; Feygels, V.; Park, J.Y.; Wozencraft, J. Modeling of Airborne Bathymetric Lidar Waveforms. J. Coast. Res. 2016, 18-30. [CrossRef]
12. Bremmer, H. Random volume scattering. Radio Sci. 1964, 68, 967-981. [CrossRef]
13. Dolin, L.S.; Levin, I.M. Reference Book on the Underwater Vision Theory; Gidrometeoizdat Press: Leningrad, Russia, 1991.
14. Ishimaru, A. Wave Propagation and Scattering in Random Media; Academic Press: New York, NY, USA, 1978.
15. Kopilevich, Y.I.; Surkov, A.G. Mathematical modeling of the input signals of oceanological lidars. J. Opt. Technol. 2008, 75, 321-326. [CrossRef]
16. Kim, M. Airborne Laser Hydrography II (Blue Book II); Philpot, B., Ed.; Cornell University Library: Ithaca, NY, USA, 2019.
