AIRBORNE ALTIMETRIC LIDAR SIMULATOR: AN EDUCATION TOOL

Bharat Lohani, Parameshwar Reddy, Rakesh Kumar Mishra

Department of Civil Engineering, Indian Institute of Technology Kanpur, Kanpur 208016, INDIA – blohani@iitk.ac.in

Commission VI

KEY WORDS: LiDAR, Simulator, Modelling of sensor, Education, Software, Tutorial

ABSTRACT:

This paper describes development of a simulator for airborne altimetric LiDAR. Main aim of simulator is to replicate the functioning of LiDAR sensor and generate data for a given terrain with specified parameters of the sensor and trajectory. The simulator has three components: 1) Terrain component, 2) Sensor component, and 3) Platform component. The terrain component is formed using multiple mathematical surfaces for bald terrain and for objects on top of the surface. The sensor component is designed to account for various parameters that can be set and varied in a LiDAR sensor. The third component attempts to replicate the platform parameters, viz., velocity, roll, pitch, yaw and accelerations. Simulated data should be similar to the data which might have been collected by actual sensor when operating with specified sensor and trajectory parameters over the given terrain. This is realized by first finding the time and space varying equation of laser vector and then determining the point of intersection of this vector with the mathematical surface representing terrain. Numerical methods are used to solve complex problems of determining the intersection of laser vector and ground surfaces. This GUI based simulator, developed in JAVA, will help in classroom and in producing data as required for conducting laboratory exercises.

1. INTRODUCTION

1.1 Background

The last decade has seen manifold growth in the use of airborne altimetric LiDAR (Light Detection and Ranging) technology. LiDAR data are complimenting several other remotely sensed data for various applications. At the same time LiDAR is competing with existing topographic data collection techniques. The main advantage of LiDAR is that highly dense and accurate topographic data are captured at high speed. In view of its typical characteristics and ease in data capture LiDAR is also finding applications in several new areas which were not considered feasible hitherto with conventional techniques.

1.2 Principle of LiDAR

LiDAR works by measuring time of travel of laser pulses fired from an airborne platform. The location and attitude of laser head are determined using onboard GPS (Global Positioning System) and IMU (Inertial Measuring Unit). The range computed using time of travel measurement is combined with the laser scan angle and attitude values to yield the laser vector. Transformation of laser vector from the laser head coordinate system to WGS-84 coordinate system is carried out to determine the WGS-84 coordinate of the point on ground hit by laser. The forward motion of platform combined with scanning in transverse direction measures a large number of points across a wide swathe on the ground.

1.3 Concept of simulator

A LiDAR simulator is aimed at faithfully replicating the LiDAR data capture process as discussed above with the use of mathematical models under a computational environment. Basically, data generated by simulator should exhibit all characteristics of the data acquired by an actual LiDAR sensor. Further, the simulator should possess flexibility of operation as in case of actual sensor.

Literature reveals that only a few attempts have been made by researchers to develop simulator for LiDAR instrument. These efforts are limited in their scope as either these consider effect of only single parameter on one kind of object (Holmgren et al., 2003) or inaccurate scanning pattern (Beinat and Crosilla, 2002). More focused and comprehensive efforts have been made to simulate the return waveform from a footprint (Sun and Ranson, 2000; Tulldahl and Steinvall, 1999).

1.4 Need of simulator

Considering the significance of technology there is a need to introduce LiDAR technology to students at undergraduate and postgraduate level. LiDAR instrument and data are costly. Collecting LiDAR data with varied specifications, as are desired for classroom teaching and laboratory exercises, may not be viable considering the cost and availability of instrument. A simulator can generate various kinds of data, as and when desired, at minimal or no cost. This data could be very useful for conducting laboratory exercises. User control over the entire data generation process in simulator can also help students in understanding the functioning and limitation of LiDAR instrument. Further, error sources and their effect on LiDAR data can be understood. In any laboratory exercise availability of ground truth is fundamental. While it may be difficult and expensive to collect ground truth in case of actual LiDAR data, for simulated data the ground truth is readily and accurately available.

In addition to education, there are several other applications where data generated by simulator can be employed. In particular, for testing the information extraction algorithms for their performance over a wide variety of data is conveniently possible with simulated data.

2. DESIGN CONSIDERATION FOR A SIMULATOR

Considering the aforesaid applications the following benchmarks are set for a simulator:

1. Simulator should employ a user-friendly GUI (Graphical User Interface.)
2. Simulator should be designed for wider distribution over various computational platforms.
3. The simulator should come along with a help/tutorial system which can explain concepts of LiDAR using user-friendly multimedia techniques.
4. It should simulate a generic LiDAR sensor and some widely available sensors in market.
5. The simulator should facilitate selection of trajectory and sensor parameters as in actual case along with the facility of introducing errors in various component systems of LiDAR.
6. Simulator should facilitate data generation for actual earth-like surfaces.
7. The output data should be available in commonly used LiDAR formats.

3. DEVELOPMENT OF SIMULATOR

In view of the aforesaid requirements programming language JAVA has been chosen for developing the simulator. This language offers good numerical and graphical programming besides, and most importantly, being platform independent. The simulator is composed of three basic components in addition to input and output modules as shown in Figure 1.

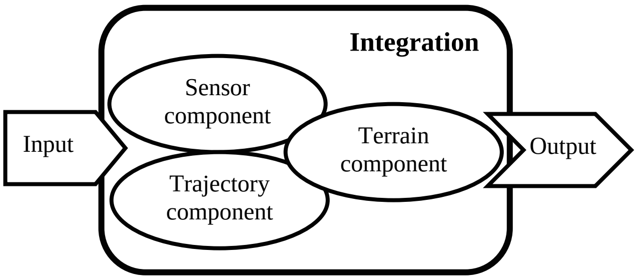


Figure 1. Basic components and their integration in simulator

These components take the form as per the user input, while their integration generates LiDAR data as desired by user. Each of these components is realised using a mathematical model. The following paragraphs describe development of these components as implemented in present version of simulator.

3.1 User input

Through a GUI the user is prompted to input various parameters as required to define the three individual components and output format. To help the user in understanding various terms and processes in LiDAR technology a help module is activated at any instance during the parameter input.

3.2 Trajectory component

As shown in Figure 2 two coordinate systems are considered. The first coordinate system is (X, Y, Z), which is absolute. All trajectory and terrain coordinates are determined in this system A system which translates with platform and remains parallel to absolute coordinate system is considered at the laser head and is referred to as gyro coordinate system henceforth. The second coordinate system is the body coordinate system, which has its origin at laser head and is affected by roll, pitch and yaw rotations. The laser vector at any instance is defined using direction cosines and coordinate of laser head in gyro coordinate system. The location and attitude parameters, as required at the instance of each laser pulse firing, are simulated as following.

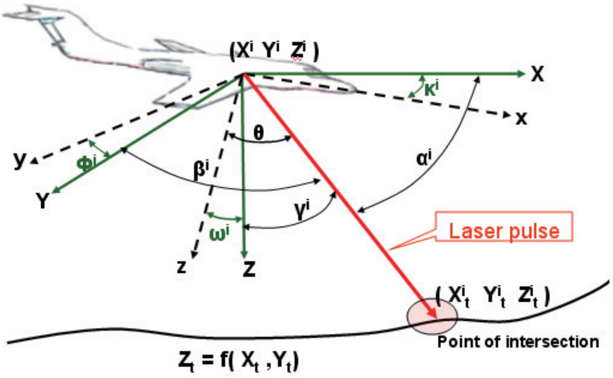


Figure 2. Schematic of laser vector intersection with a surface

3.2.1 Location

A trajectory is defined by the location of laser mirror centre (point of origin of laser vector) in the absolute coordinate system at each instance of firing of laser pulse. To simulate the trajectory and to incorporate a possibility of introducing errors in trajectory parameters the following procedure is employed.

Let:
Firing frequency = F.
Time interval between firing of successive pulses= $d_t = 1/F$
Total number of points on trajectory wherefrom pulses are fired = n
Velocity of platform in flight direction (which is considered here as X axis) at i^{th} point on trajectory = $u_x^{\ i}$
Laser head coordinate at i^{th} point on trajectory = (X^i, Y^i, Z^i)

At each successive d_t interval one needs to compute the location of laser head. The aerial platform is subject to internal and external force fields. While the external forces result in accelerations in X, Y and Z directions the internal force attempts to maintain a uniform speed and the designate trajectory. The net effect is that the platform is subject to random accelerations in three axes directions. The following system is employed to simulate accelerations. This system ensures a pseudo-random generation of acceleration values.

$$a_x^i = \sum_{j=1}^{J} A_j \sin(B_j(\frac{2\pi}{T}(id_t))) + \sum_{k=1}^{K} C_k \cos(D_k(\frac{2\pi}{T}(id_t))) \quad (1)$$

Where $a_x^{\ i}$ is the acceleration at i^{th} point in X direction. T is the total duration of a trajectory. The parameters of this equation J, K, A, B, C and D control the nature of curve. Developed software permits selection of these parameter values within ranges that generate accelerations as may be observed in a normal flight. Similarly, a_y^i and a_z^i are also generated with different values of parameters in above equation. Using the acceleration values at i^{th} point the new location of laser head (i.e., X^{i+1}, Y^{i+1}, Z^{i+1}) after d_t interval is computed using equations similar to:

$$X^{i+1} = X^i + u_x^i d_t + \frac{1}{2} a_x^i d_t^{\,2} \qquad (2)$$

3.2.2 Attitude

As in case of acceleration, due to internal and external force fields, the attitude will change within certain limits and may exhibit a random behaviour. To realise this the attitude values (i.e. ω^i, φ^i, κ^i) at any i^{th} point are determined using the equation 1. Similar to the case of acceleration, the developed simulator permits selection of these parameter values in the ranges which generate attitude values as in case of a normal flight.

The outcome of aforesaid is that at each point wherefrom a laser pulse is fired the attitude values and coordinate of point are known in the absolute coordinate system.

3.3 Sensor component

The sensor component aims to replicate the scanning mechanism of sensor as per the user input parameters. The following differential equation is employed to simulate sinusoidal scanning pattern.

$$\frac{d^2\theta}{dt^2} + \frac{k}{R}\theta = 0 \qquad (3)$$

Where θ is the instantaneous scan angle measured from the z axis of body coordinate system. k and R are the driving acceleration and radius of the scanning mirror, respectively. Solution of equation (3) gives value of θ at any time t as:

$$\theta = \frac{\frac{S}{2}\sin(t\sqrt{\frac{k}{R}})}{\sin(\frac{1}{4f}\sqrt{\frac{k}{R}})} \qquad (4)$$

Where S is full scan angle and f is the scanning frequency as input by the user. At $t=0 \Rightarrow \theta=0$, i.e. beginning of scan, i.e. laser vector along z axis in body coordinate system. At *t=0.25*time period =>* $\theta=S/2$.

3.4 Terrain component

Two approaches have been investigated for simulating the bald earth surfaces and above ground objects.

3.4.1 Vector approach

In this, a terrain is represented using mathematical equations, which yield earth like surfaces. A few of these are:

$$\begin{aligned} Z &= AX + BY + C \\ Z &= A[\sin(X/B) - \sin(XY/C)] + D \\ Z &= A[\sin(X/Y) - \sin(XY/B)] + C \end{aligned} \qquad (5)$$

The GUI permits selection of these surfaces and their parameters.

3.4.2 Raster approach

In this the surfaces resulting from above equations are rasterized. Most importantly, this approach permits populating the raster with above ground objects so as to create the terrain like a 3D city model. Those cells where an over ground object is desired to be placed take new values as per the height and shape of object.

3.5 Integration of components

In the heart of simulator lies generation of the laser vector and its intersection with simulated terrain. The point of intersection yields the coordinate of terrain point. As shown in Figure 2, for any i^{th} point on trajectory there exists a laser vector. The equation of laser vector is given as:

$$\frac{X - X^i}{a^i} = \frac{Y - Y^i}{b^i} = \frac{Z - Z^i}{c^i} \qquad (6)$$

Where a^i, b^i, and c^i are direction cosines ($\cos\alpha^i$, $\cos\beta^i$, and $\cos\gamma^i$, respectively) of laser vector with respect to gyro coordinate system at i^{th} point. The values of α^i, β^i, and γ^i are determined from known values of attitude (ω^i, φ^i, κ^i) and instantaneous scan angle (θ).

The point where laser hits the terrain, following the above laser vector, is computed by solving for intersection of equation (6) and equation (5) or the rasterized terrain. Solution is realised using specially formulated numerical methods. These methods differ for vector and raster terrain and also depend upon the basic equations employed to create terrain. Full description of these methods is beyond the scope of this paper. At this stage coordinates of all points of intersection (X_t^i, Y_t^i, Z_t^i) are known.

3.6 Error introduction in data

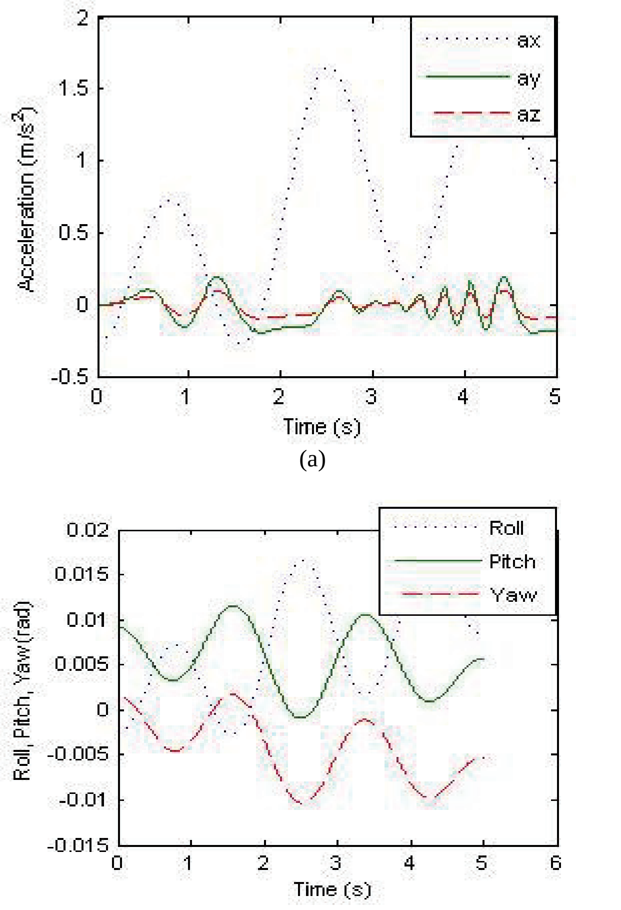
In present version of simulator a normal error is introduced in the terrain coordinates computed in the above step in X, Y and Z directions separately. The system for introducing error in X direction is shown below:

$$X_T^i = X_t^i + N(\mu_X, \sigma_X^{\,2}) \qquad (7)$$

Where X_T^i is the X coordinate value with error. Similar systems with different values of parameters are used for Y and Z coordinates. It is assumed that errors in X, Y and Z directions follow normal distribution. Further, when introducing these errors it is ensured in algorithm that there is no spatial auto-correlation of error. The parameters of this distribution are known from field experience and are reported by the vendors of sensors. The simulator facilitates variation of these parameters.

4. RESULT AND DISCUSSION

Simulated trajectory and attitude parameters are shown for a duration of 5 seconds (Figure 3, a and b). Though, it is statistically difficult to compare simulated data with any set of actual flight data, as these two represent two different populations, the former amply exhibit the random nature of parameters as in any normal flight.

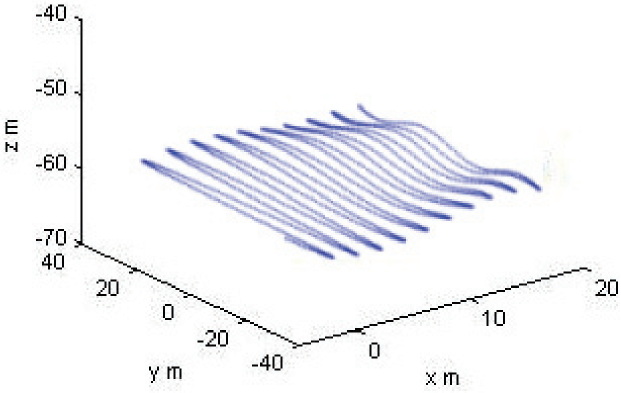


(a)

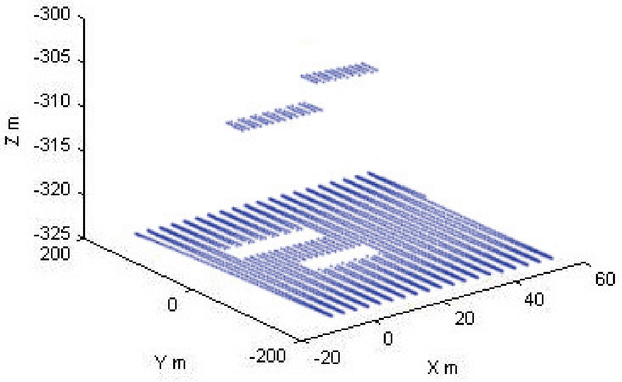
(b)

Figure 3: Acceleration (a) and (b) attitude values for simulated flight.

LiDAR data generated for two terrain types are shown in Figure 4 (a and b) along with the corresponding terrain parameters. For both of these cases the flight parameters are: u_x^0 =*100 m/s, F= 10kHz, f=48Hz and S=50°.*



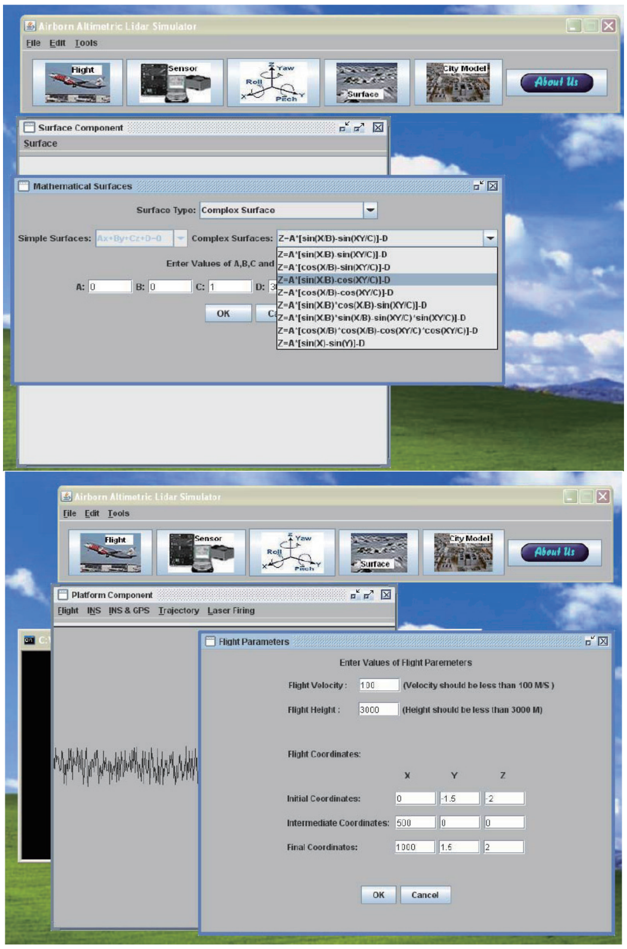
(a)



(b)

Figure 4: (a) Vector terrain *Z=sin(X/10)-sin(XY/90)-60* for *0<X<20* and *-40<Y<40* with flying height 60m; (b) Raster terrain at *Z=-325* for *0<X<50* and *0<Y<400* with flying height 0 m. First roof plane (20 m by 30 m) with left-bottom corner at (0 m, 30 m, 15 m) and second roof plane (15m by 30 m with slant height 5m) with left-bottom corner at (10 m, -40 m, 18 m)

The simulator is GUI based. Figure 5 shows example screenshots of user interface. The actual performance of software can be demonstrated when a user works on it.



Surface Type: Complex Surface

Simple Surfaces: Ax+By+Cz+D=0 Complex Surfaces: Z=A*[sin(X/B)-sin(XY/C)]-D

Z=A*[sin(X.B)-sin(XY/C)]-D
Z=A*[cos(X/B)-sin(XY/C)]-D
Z=A*[sin(X.B)-cos(XY/C)]-D
Z=A*[cos(X/B)-cos(XY/C)]-D
Z=A*[sin(X.B)*cos(X.B)-sin(XY/C)]-D
Z=A*[sin(X.B)*sin(X/B)-sin(XY/C)*sin(XY/C)]-D
Z=A*[cos(X/B)*cos(X/B)-cos(XY/C)*cos(XY/C)]-D
Z=A*[sin(X)-sin(Y)]-D

Enter Values of A,B,C and

A: 0 B: 0 C: 1 D: 3

Enter Values of Flight Paremeters

Flight Velocity : 100 (Velocity should be less than 100 M/S)

Flight Height : 3000 (Height should be less than 3000 M)

Flight Coordinates:

	X	Y	Z
Initial Coordinates:	0	-1.5	-2
Intermediate Coordinates:	500	0	0
Final Coordinates:	1000	1.5	2

Figure 5: Two example screenshots showing user interface and input of parameters.

5. CONCLUSION

The developed simulator facilitates generation of LiDAR data for the user created terrain as per the specified sensor and trajectory parameters. The simulator can be highly useful in a classroom for demonstrating LiDAR data capture process and the effect of changes in flight-sensor parameters and error. Different kinds of data that can be generated along with accurate and full ground truth will help in conducting laboratory exercises.

The present version of simulator is being modified to incorporate more faithfulness by introducing separate GPS and IMU data collection and their integration; by introducing concepts of separation of GPS antenna, laser head and INS; by incorporating atmospheric effects and by facilitating for multiple returns, etc. Further efforts are on to generate a full waveform digitization. The main stumbling block in this development is simulation of bald earth and above-ground objects. Voxel representation of terrain will be attempted to handle this problem.

References

Beinat, A. and Crosilla, F., 2002. A generalized factored stochastic model for optimal registration of LIDAR range images, *International Archives of photogrammetry an remote sensing and spatial information sciences*, 34(3/B), pp. 36-39.

Holmgren, J., Nilsson, M., and Olsson, H., 2003. Simulating the effect of lidar scanning angle for estimation of mean tree height and canopy closure. *Canadian Journal of Remote Sensing*, 29(5), pp. 623-632.

Sun, G. and Ranson, K. J., 2000. Modeling Lidar returns from forest canopies, *IEEE trans. On geosciences and remote sensing*, 38(6), pp. 2617-2626.

Tulldahl, H. M. and Steinvall, K. O., 1999, Analytical waveform generation from small objects in lidar bathymetry, *Applied optics*, 38(6), pp. 1021-1039.

5.1 Acknowledgements

This work is supported under RESPOND programme of ISRO.