AN OBJECT-ORIENTED SOFTWARE DEVELOPMENT APPROACH TO DESIGN SIMULATOR FOR AIRBORNE ALTIMETRIC LIDAR

Rakesh Kumar Mishra, Bharat Lohani

Geoinformatics division, Indian Institute of Technology Kanpur, Kanpur 208016, INDIA – rkmishra@iitk.ac.in

KEY WORDS: Object-oriented design, Software development, class, object, LiDAR, Research and education tool

ABSTRACT: Topographic data are fundamental for several applications. The latest technique for topographic data collection is Airborne Altimetric LiDAR. However, LiDAR data is not available for research and education as required. This paper describes the object oriented design methods used to develop a software system to simulate the functioning of an airborne altimetric LiDAR instrument. Our objective is to model functions of a LiDAR instrument by using object oriented software development approach so that the developed software could be maintainable, reliable and scalable. The object-oriented software development processes, viz., object–oriented analysis, object-oriented design and object-oriented programming have been done to realize the software. The simulator is conceived with three components, namely terrain component, sensor component and trajectory component. Each component has been divided into their sub modules and designed independently. The terrain component deals with generation of bare terrain and objects on top of the terrain. The sensor component deals with design of commercially available sensors or a generic sensor. And the trajectory component deals with modelling of platform parameters, viz., velocity, roll, pitch, yaw and accelerations. Numerical methods are used to solve complex problems for generating LiDAR data from simulated terrain and flights. LiDAR data files are generally very large. Considering this, special data structures and file formats have been designed to improve the performance and to solve the memory problems. This user friendly GUI based simulator, developed in JAVA programming language, is an ideal tool for research and education.

1. INTRODUCTION

1.1 Background

The last decade has seen manifold growth in the use of airborne altimetric LiDAR (Light Detection and Ranging) technology. Due to the main advantage of measuring topography through highly dense and accurate data points which are captured at high speed, the LiDAR technology has found several interesting applications (Lohani, 2001; Queija, et al., 2005).

1.2 Object-oriented software design

Object-oriented software design is a design strategy where system designers think in terms of ‘things’ instead of operations or functions. The executing system is made up of interacting objects that maintain their own local state

and provide operations on that state information (Figure 1). They hide information about the representation of the state and hence limit access to it. An object-oriented design process involves designing the object classes and the relationships between these classes. When the design is realized as an executing program, the required objects are created dynamically using the class definitions.

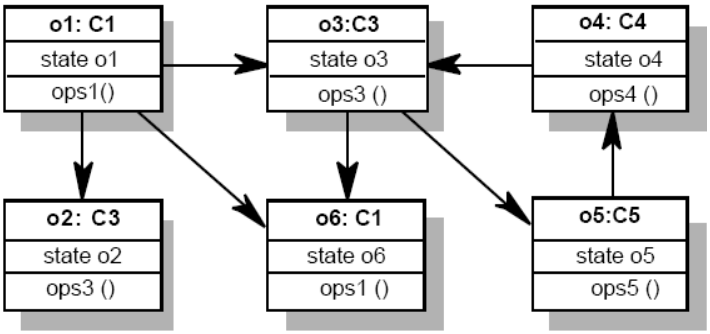

Figure 1. System with interacting objects (IAN Sommerville-2005)

Object-oriented analysis and design is more cost-effective and a faster way to develop software and systems. This technology cuts development time, overhead and enables software engineers to make reusable, reliable and easily maintainable applications.

In addition, this technology offers a new and powerful model for writing software. OOP (Object-oriented programming) allows decomposition of a problem into a number of objects and then builds data and functions around these objects.

1.3 LiDAR simulator

A LiDAR simulator is aimed at faithfully emulating the LiDAR data capture process with the use of mathematical models under a computational environment. Basically, data generated by simulator should exhibit all characteristics of data acquired by actual LiDAR 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 the effect of only single parameter on one kind of object (Holmgren et al., 2003) or inaccurate scanning pattern (Beinat and Crosilla, 2002). Another attempt is made (Kukko and Hyyppa, 2007) using MATLAB however, the simulator has limitations as it is designed only for test purpose and does not offer flexibility and completeness. 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 Statement of requirements

Considering the significance of LiDAR technology there is a need to introduce LiDAR technology to students at undergraduate and postgraduate level. Since the 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, testing the information extraction algorithms for their performance over a wide variety of data is conveniently possible with simulated data. It can also be used for LiDAR flight planning.

2. BENCHMARK FOR LIDAR SIMULATOR

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

- Should employ a user-friendly GUI (Graphical User Interface.)
- Should be easily extendable and maintainable.
- Should be designed for wider distribution over various computational platforms.
- It should simulate a generic LiDAR sensor and some widely available sensors in market.
- 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.
- Should facilitate data generation for actual earth-like surfaces.
- The output data should be available in commonly used LiDAR data formats.
- Should display the generated LiDAR data.
- Should come along with a help/tutorial system which can explain concepts of LiDAR using user-friendly multimedia techniques.

3. SOFTWARE DEVELOPMENT

3.1 Object-oriented analysis

On the basis of above requirements the use case document is made. In analysis, key classes are identified from the use cases. The interaction between the objects belonging to these classes is determined from the each use case. The objects reflect entities and operations that are associated with the problem to be solved. Analysis culminates in identifying classes implied by the use cases, and documenting them using an Analysis Class Diagram.

3.2 Object-oriented design

Based on the object oriented analysis the object-oriented model of the simulator is designed to implement the identified requirements. In this the objects are related to the solution of the problem that is being solved. The simulator is divided into three basic components as shown in Figure 2. Each component

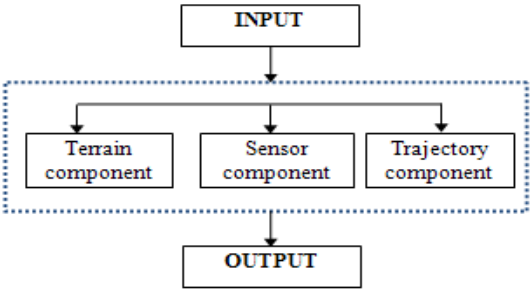

Figure 2. Basic components of simulator

is divided into their sub modules and classes are identified from these modules. As per the requirement relationship between classes, sub classes within class, abstraction behaviour of the class and common behaviour of the classes are identified. Functions of the classes are designed for each basic task. For algorithmic detail please refer to (Lohani and Mishra,2007). The following paragraphs describe the components of the software.

3.2.1 Terrain component

Terrain component (Figure 3) is combination of three modules, Polynomial surface, Raster and fractal. These three modules are designed to simulate bare earth surfaces and above ground objects.

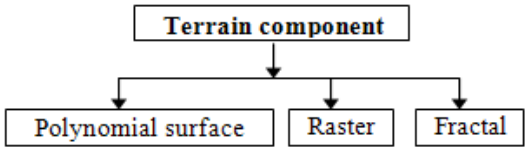

Figure 3. Terrain component

3.2.2 Sensor component

Two commercially available sensors (ALTM and ALS 50) and a generic sensor (Figure 4) are designed as independent modules. Independent classes are designed for each sensor.

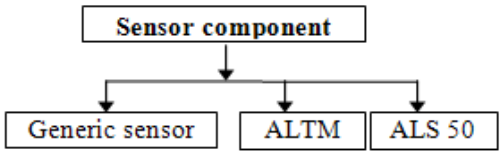

Figure 4. Sensor component

3.2.3 Trajectory component

This component (Figure 5) is divided into two modules, location and attitude. Location module gives the position of aeroplane at each instance on the basis of its sub modules, velocity and acceleration. And the altitude module is responsible to generate pshuedo random (controlled) Roll, Pitch, and Yaw. Integration of altitude, location modules and trajectory component generates trajectory coordinate (GPS coordinates) of aeroplane at each instance and provides the attitude parameters of the platform.

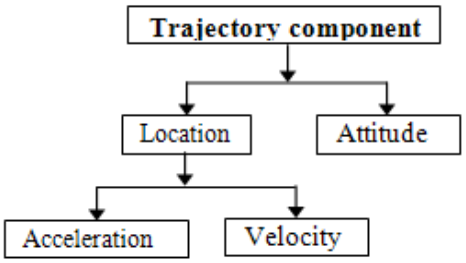

Figure 5. Trajectory component

3.2.4 Integration of the components

All components function independently to accomplish the task given to them by making use of other components wherever it is required. The classes of each component take input data from other classes. By making use of abstraction, inheritance and polymorphism whole system is integrated to produce the desired output.

3.3 System Implementation

To realize the aforesaid design JAVA programming language is used. Object-oriented design can be implemented

directly using JAVA. Being a cross-platform language the developed software can be run on any operating systems. Further, it supports excellent GUI features and the developed software will be robust. The industry standard are followed in coding the programs. The arrangement of classes is made in such a way that the entire set of code and modules can be easily understood and modified.

Figure 6 shows the GUI for terrain component. The user has option to choose terrain type i.e polynomial surfaces, raster terrain or fractal terrain. In the case of raster terrain one can place many objects on the top of the surface, drag-drop , rotate and resize these objects as per the requirment.

In figure 7 the GUI for sensor component is shown. It facilitates selection of commercial (ALTM and ALS 50) or generic sensors and set their parameters which are desired for LiDAR data generation, viz., scan angle, scan frequency, firing frequency, type of scanning etc. The range of parameters is constrained in commercial sensors as per their specifications. The generic sensor permits selection of any range of parameters. Software has GUI for controlling acceleration and attitude. GUI for output generates output data in common LiDAR data formats and a report on the LiDAR flight. Only few GUI are being shown for the sake of space.

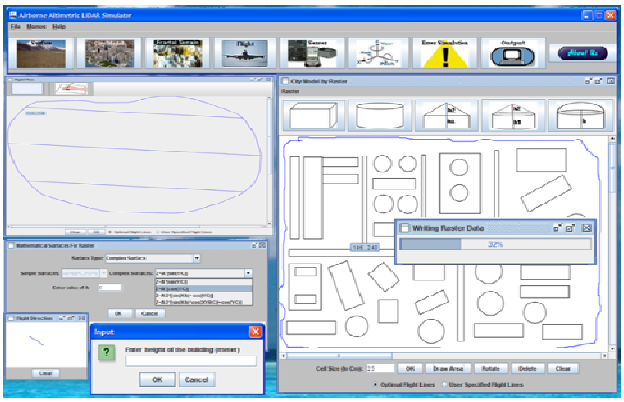

Figure 6. GUI for terrain generation

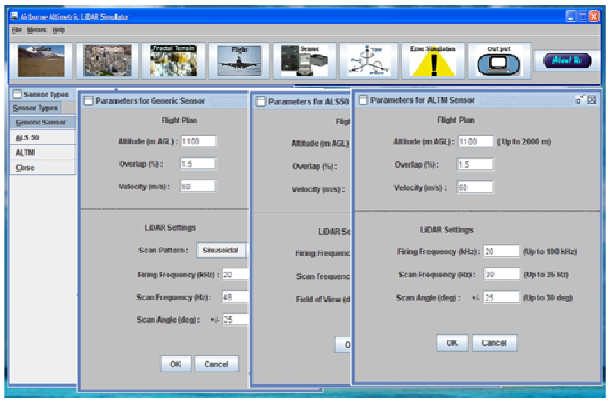

Figure 7. GUI for sensor selection

4. RESULT AND DISCUSSION

A hypothetical terrain (400 m by 600 m) is created by the simulator over a flat surface and is populated with

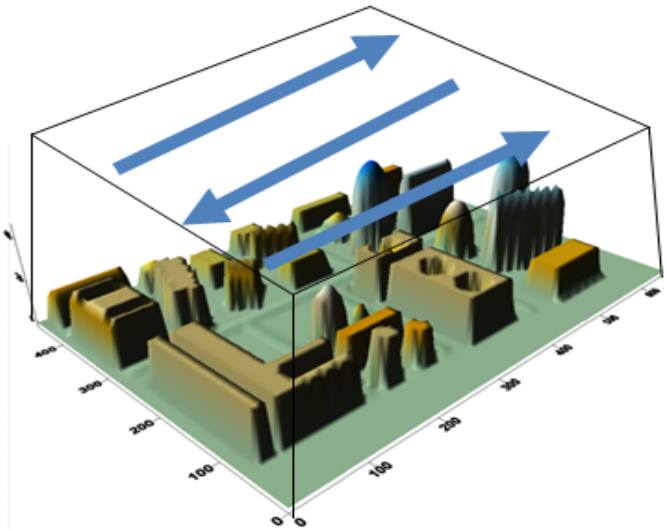

Figure 8. raster data display in Surfer

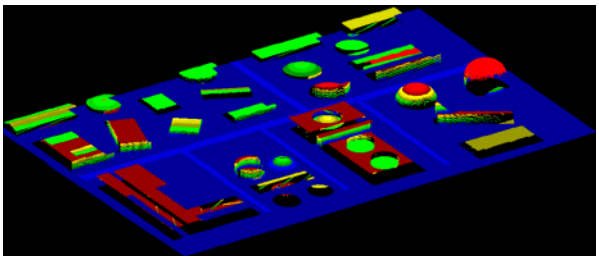

Figure 9. Lidar data of raster display in terrasacn

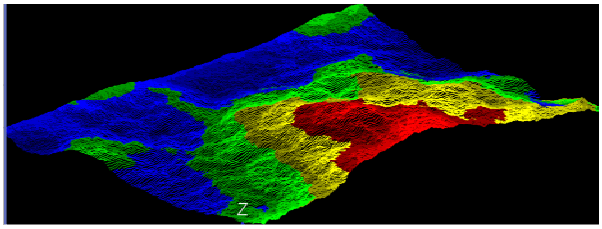

Figure 10. LiDAR data of fractal terrain

building like shapes are rasterized. A view of this is shown in Figure 8, which is displayed using surface feature of Surfer. Lidar data are generated for this terrain with the parameters: Flight velocity: 60 m/s;Altitude: 190m; Firing frequency: 20 KHz; Scan frequency: 48Hz; Scan angle 50^0; No. of flight lines: 3; Overlap 10%. Resulting LiDAR data are imported in Terrascan software and displayed (figure 9). The generated LiDAR data of a fractal terrain is shown in figure 10. Only few views of LiDAR data are being presented for the sake of space. (More such views of results and GUI screens and algorithms can be seen at "http://home.iitk.ac.in/~blohani/Limulator/Lim_index.html" which is site developed specially for this software.

5. CONCLUSION

The present version of software offers a user friendly GUI based interface so that the user can use this software with ease. It simulates the process of LiDAR data collection for LIDAR sensor as well as the user has freedom to set the parameters according to his/her choice which is not possible in the case of actual LIDAR. Different data sets can be generated for the same terrain by setting the different parameters. The simulator can be useful to generate LiDAR data for research to test algorithms. It is also useful in a classroom for demonstrating LiDAR data capture process and understanding the effect of flight and sensor parameters and their errors. The software is developed using object-oriented methodology so it can be extended further very easily. JAVA programming language is used to develop the software which makes this software robust and platform independent.

References

Marciniak, J.J., 2001, Process models in software engineering, *encyclopedia of software engineering*. Walt Scacchi, Institute for Software Research, University of California, Irvine.

Mishra, R. K., Lohani, B.,2007, Object oriented software engineering for designing an aerial survey LiDAR, IT and GIS strategy for infrastructure 2007, Jaipur, INDIA, October 25-26.

Lohani, B., Mishra, R. K., 2007, Generating LiDAR data in laboratory: LiDAR simulator, International Archive of Photogrammetry , and Remote Sensing XXXVI(3)W52 of Laser Scanning 2007 and SilviLaser 2007, Espoo, Finland, September 12-14.

Kukko, A. and Hyypa J., 2007, Laser scanner simulator for system analysis and algorithm development: A case with forest measurements, ISPRS Workshop on LASER Scanning 2007 and SiviLaser 2007, Espoo, September 12-14, 2007, Finland.

Lohani, B., Reddy, P., and Mishra, R.K., 2006, Airborne Altimetric LiDAR Simulator: An education tool, *InternationalArchives of the Photogrammetry, Remote Sensing and Spatial Information Science*, XXXVI(6), Tokyo, Japan.

Lohani, B., 2001, Airborne altimetric LiDAR for topographic data collection: Issues and application., *Proc. of International conference MAPINDIA-2001*, 7-9 February 2001, New Delhi. Queija, V. R., Stoker, J. M., and Kosovich, J. J., 2005, Recent U.S. geological survey applications of LiDAR, *PE&RS*, 71(1), pp. 5-9.

5.1 Acknowledgements

This work is supported under RESPOND programme of ISRO.