NEAR REALISTIC AND THREE DIMENSIONAL VISUALIZATION OF TERRAIN USING LIDAR DATA

Suddhasheel Ghosh and Bharat Lohani

Geoinformatics Division, Department of Civil Engineering,
Indian Institute of Technology Kanpur, Kanpur, UP 206018
suddhasheel@gmail.com, blohani@iitk.ac.in

Commission IV/4

KEY WORDS: Immersive Visualization, Geovisualization, LiDAR

ABSTRACT:

Visualization in scientific computing has been a thrust area in the field of computer science since the last century. This has triggered the development of technology in terms of memory, processing speed and storage capacity. LiDAR technology is an emerging field of research where the topography of the terrain is captured in the form of X, Y and Z coordinates. In the light of the developments in scientific visualization and LiDAR technology, systems could be designed which allow three dimensional immersive visualization of terrain. LiDAR data prove ideal for 3D visualization, particularly in an immersive environment, owing to their 3D nature and ability to integrate with other remotely sensed data at high resolution. Traditional methods of visualization of LiDAR data involve feature extraction in terms of roads, buildings and trees followed by 2.5D (perspective) visualization. In this paper, we attempt to develop a pipeline which enables processing of raw LiDAR data to enable visualization with texture information. The texture information is obtained from a digital colour aerial photograph, which is first co-registered with the LiDAR points. Facets of the terrain are then generated by performing a Delaunay triangulation on the LiDAR points. This is coded in the form of a module named FACET. Detection of planes and other surfaces in the terrain is performed by the module named DETECTOBJECT. In this process, smoothing of lidar points is also done to eliminate visually non-significant details. Finally, the entire scene is converted to a SGI OpenInventor compatible view file by modules named TEXTUREMAP and TEXTURE2IV. These modules are programmed using Java Software Development Kit. In our work till date, we have been able to generate views of points plotted in 3D space, facets with tones picked up from the centers of the facets and texture wrapped views of the terrain, using the outputs from FACET and TEXTURE2IV. The system facilitates stereo-viewing of terrain from any viewpoint dynamically. In this paper, we focus more on our attempt to make the views more realistic by the use of smoothing and objects. Various views of the results are presented and discussed in the paper.

1 INTRODUCTION

LiDAR stands for Light Detection and Ranging and is an active remote sensing technique, which involves sending laser pulses directed towards the ground and measuring the time of pulse return. The time measured, is processed to compute the variable distances between the sensor and the ground. With the help of GPS (Global Positioning System) observed aircraft location and IMU (Intertial Measuring unit) observed attitude values, coordinates of the points on ground where LiDAR pulse hit are computed. Modern LiDAR data acquisition systems consist of the airborne GPS and IMU (for measuring sensor location and angular orientation), a rapid pulse laser support with a high measuring rate (up to 200,000 pulses per second), a highly accurate clock, onboard computer support, reliable electronics for the sensor and robust data storage. Modern sensors in the LiDAR systems are able to record up to five returns per pulse, which enables capturing 3D structure of terrain as a dense set of points accurately and efficiently (Lillesand and Kiefer, 1999). After initial post processing, LiDAR data are filtered for noise removal and become available as a file of X, Y, Z points either in text or LAS format (`http://www.lasformat.org/`). LiDAR have found several applications and many more are being attempted. Availability of 3D structure of terrain in the form of LiDAR data points makes this data ideal for use in visualization applications.

2 GEOVISUALIZATION

The concept of visualization in scientific computing was introduced by McCormick et al. (1987). Advances in scientific visualization have been spurred by several developments in terms of algorithms, memory and high-end graphic cards. Real time 3D computer graphics has become a major field in Computer Science and is being applied in fields like virtual reality, video games, computer aided design etc.

Geovisualisation represents a fundamental area of application of 3D computer graphics. Recent developments in geoinformatics research especially point to those areas which need planning and decision support and express the requirement of systems enabling the visualization of geoscientific data. Spatial planning could be aided if these systems allow the visualization of data not only from an aerial view, but also from various perspectives and scales (Döllner, 2005). Visualization of the third dimension has been of great interest and use for geo-applications. Various commonly used methods of visualization of geographic data, which include third dimension, are planar maps with height represented by contour or colour, perspective views, hillshade views etc. These views are suitable for a limited use as the third dimension is either inexplicitly interpreted or an illusion of third dimension is created on a 2D surface. However, a human is trained to look at the terrain in actual 3D and it is desirable that the user is presented with such a model for easy of visualization and decision making. An example of this kind of viewing is photo-stereopairs, which facilitates a 3D view of terrain. However, photo-stereopairs have limitations as the view is available from only a single viewpoint and is also difficult for interactive measurements.

The aim of the technique outlined here is to develop a system where views of the terrain are available in 3D at several scales and from different viewing angles. Further, a facility to carry out measurements interactively on the 3D model is desirable.

3 LIDAR DATA FOR GEOVISUALIZATION

As seen above, LiDAR technology captures 3D structure of the terrain in the form of coordinates of millions of points spread all over the terrain. Further, in general it is a practice to fly a digital or analog camera along with LiDAR to capture simultaneous aerial photographs.

Therefore it is only natural that LiDAR data be selected for development of the system which allows 3D visualization at multiple scales and perspectives with the aid of aerial photographs.

4 PREVIOUS WORK AND PRESENT OBJECTIVES

Previously, Ghosh and Lohani (2007b) were able to generate stereo scenes using LiDAR data and digital aerial photographs. Professional software like MATLAB and ERDAS Imagine were used for performing computations and image processing. The remaining coding was done in GNU Compiler Collection (GCC) and Qt Open Source Edition. The scenes generated were displayed only with points rendered, tone rendered (tones picked up from centroids of generated facets) and texture rendered (Texture wrapped over the generated facets). However, the scene suffered with a drawback that the walls actually present in the scene (for residential or office areas), were not satisfactorily generated.

In this paper, we attempt to develop a pipeline which is cross platform, less expensive, faster, and generates a more realistic version of the lanscape from all the angles and at different scales.

5 DATA USED

In 2004, Messrs Optech Corporation conducted a LiDAR flight over the Niagra Falls with the ALTM sensor. The airplane flew at an average height of 1190m, with a DSS 301 SN0039 camera on board for the aerial photographs. The average density of the LiDAR data is 2.74 points per square metre. For computational convenience, the data was subset into an area of 100m $\times$ 100m.

6 TOOLS USED

THe following tools were used in the development of a system for near-realistic visualization of LiDAR data:

1. Fedora Core 6 operating system
2. Java Software Development Kit 1.6
3. SGI Open Inventor 2.1.6 on the Linux Platform
4. ERDAS Imagine 8.7
5. QHULL Binaries for Linux. (http://www.qhull.org).

The choice of the tools were based on economising the cost of development as well as developing a cross platform utility.

7 CONCEPTUAL FRAMEWORK

7.1 Stereoscopic visualization

Stereoscopy is possible because of binocular vision, which requires that the left-eye view and the right-eye view of an object be perceived from different angles. It is possible by an arrangement where the right-eye image is presented to the right eye and the left-eye image to the left (www.stereoscopy.com).

7.2 Generation of stereo

LiDAR data present the actual 3D model of terrain generalized in the form of a point cloud. As shown in figure 1 each LiDAR point can be mapped onto two 2D image planes thus giving their position with desired x-parallax.

To generate a perspective view from a 3D model, let us assume a point $M(x_m, y_m, z_m)$ on the model, a photo plane $PP : z = ax + by + c$ and a camera position $P(x_c, y_c, z_c)$. The equation of the line PM is given as

$$\frac{x - x_c}{x_m - x_c} = \frac{y - y_c}{y_m - y_c} = \frac{z - z_c}{z_m - z_c}$$

We can then determine the point of intersection of the line PM and the plane PP by solving the following two equations simultaneously:

$$\frac{x - x_c}{x_m - x_c} = \frac{ax + by + c - z_c}{z_m - z_c} \quad (1)$$

$$\frac{y - y_c}{y_m - y_c} = \frac{ax + by + c - z_c}{z_m - z_c} \quad (2)$$

This method projects the 3D model of the terrain to the plane PP. This calculation may be thus used to generate the perspective view with respect to a viewport using the traditional computer graphics techniques.

Now if there were two perspective centers placed at a certain distance, the mapping of entire point cloud to the two planes would generate stereo-pairs which when seen through stereoscope would show point cloud in 3D. It is possible to generate infinite sets of such stereo-pairs from different view angles and at varied scales. Dynamically generating these stereo-pairs and viewing through a stereo-viewing system will provide the desired visualisation. The main details for realisation of this concept are presented below.

8 METHODOLOGY

8.1 Co-registration of Image and LiDAR point data

A raster DEM was produced by interpolating the LiDAR data and this was then used to geocode the high resolution digital aerial photograph using ERDAS Imagine 8.7.

8.2 Data tiling

For computational convenience, the LiDAR point data is divided into smaller overlapping tiles. A data structure was designed such that it could handle the overlapping areas. Since the tiles are designed to be overlapping, a single LiDAR point can also fall into more than one tile (which can vary from 1 to 4). This data was then written to a file in the binary format.

8.3 Tile-wise triangulation

Since the LiDAR data displays a lot of redundancy in terms of storage of points, the repeated points were removed. A 2D delaunay triangulation of the points in each tile was then performed using the x and y coordinates of the LiDAR data and the QHULL binary. This algorithm was coded as a module named FACET.

8.4 Elimination of Spurious Triangles

A quick preview of the collected LiDAR data gives an idea that the points are least captured on the walls. In such a scenario, the triangulation process generated very long triangles at the sides of the buildings. These triangles are detected and eliminated by using a longest median criteria. A threshold value is keyed by the user in the designed interface (see figure 2).

8.5 Identification of planar and non-planar objects

Definition: The *neighbourhood criteria* of two triangles T_i and T_j is satisfied if

- The dihedral angle (angle between the normals of two planes) between the triangular planes is less than θ_t where θ_t is the threshold provided by the user.
- Either T_j has a common edge with T_i or with another triangle T_k which has a common edge with T_j and also satisfies the first condition.

If the neighbourhood criteria between two triangles T_j and T_k is satisfied, then we express it as follows:

$$nbd(T_j, T_k) = true$$

Terminology: We say that a triangle is *in the neighbourhood of another triangle* if it satisfies the neighbourhood criteria.

Since facets have already been calculated, it is trivial to calculate the equations of the planes corresponding to each of the facets and also the direction cosines of the normal to the planes. Coplanar surfaces can thus be determined, using the equations of the normals and the neighbourhood criteria. This procedure can be summarized in the following algorithm:

1. Let $\{T_i\}_{i=0}^{n}$ be a collection of triangles.
2. $\forall\ g < n - 1$, we compare T_g with T_h, where $h = g + 1, \ldots, n$
3. If $nbd(T_h, T_g) = true$ with a given threshold θ_t, we put T_h into a collection C_λ.
4. This process will partition the set of triangles into separate collections of triangles $\{C_\lambda\}_{\lambda=0}^{m}$ where m denotes the number of collections.

8.6 Generalization

Facets identified to be as co-planar, are then processed to generate a least squares equation of the plane. This process is repeated for all the tiles in the data. The overlapping areas help in determining the continuity of the planes. Further, owing to the limitations of the aerial LiDAR data with respect to its ability in capturing the completely vertical surfaces, not very vertical walls are found along the sides of the buildings. In such a scenario, for example, building sides are supposed to be vertical and near vertical facets are readjusted.

For generalization we have the following steps:

1. $\forall \lambda = k$, the collection C_k is considered.
2. The vertices $\langle v_i \rangle$ of the collection are taken and the convex hull is constructed using the QHULL Binary.
3. A vertical bounding plane is calculated based on the positions of the vertices.

8.7 Texture Wrapping

The geocoded image produced in the first step was then converted to a SGI RGB (Haeberli, 1988) format using a combination of MATLAB and Java codes. The facets generated by FACET and the RGB file were combined to generate an OpenInventor scene file. This module was coded using Java and was named FACET2IV.

8.8 Stereo Display and Anaglyph

Stereo visualization is possible using the stereo-buffering technique wherein left and right perspective views of the terrain are alternately generated on the screen. Since stereo buffering consumes a lot of memory, it has now been coded into hardware. These alternate scenes can be then viewed as a real stereo using shutter glasses.

9 RESULTS AND DISCUSSION

For the convenience of computation, the lidar data was first converted to a tiled format which was based on personally developed data structure. The computations would increase if the number of the tiles were too high or too less. As a result, an optimum number of tiles was chosen as 25. The interface designed to allow this conversion is seen in 1.

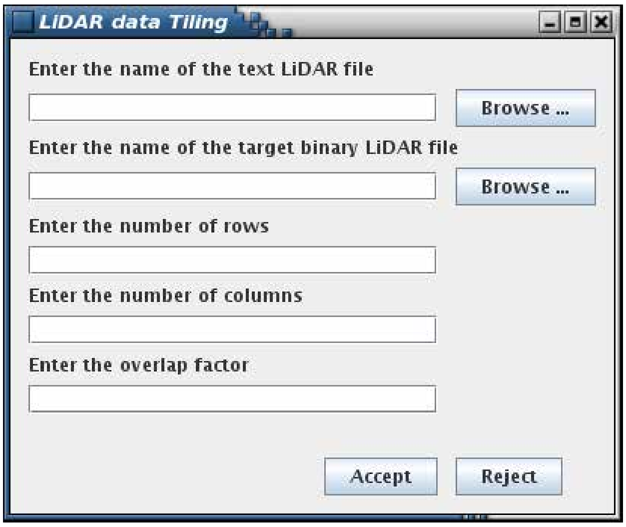

Figure 1: Interface for Tiling

The interface designed to generate the scenes from LiDAR data and the geocoded aerial photograph is shown in Figure 2. Using this interface, the lidar data is first triangulated tilewise and then each of the tiles is converted to a scene. This interface, therefore combines the FACET algorithm and the IVSCENE This interface generates the OpenInventor compatible scene file. The scene file can be then visualized in OpenInventor in different modes.

The point view generated is shown in Figure 3. It was seen that the features available in the original were clear only when the view was rotated to a certain angle. The aid of 3D NuVision Shutter glasses could not also help the perception of the third dimension.

The scene generated by open inventor could also be seen in a wireframe mode (see Figure 4). It was seen again that the presence and the visibility of lines beyond some of the facets succeeded in confusing the perceiver. As for the previous one, this view was clear only when rotated to a particular angle.

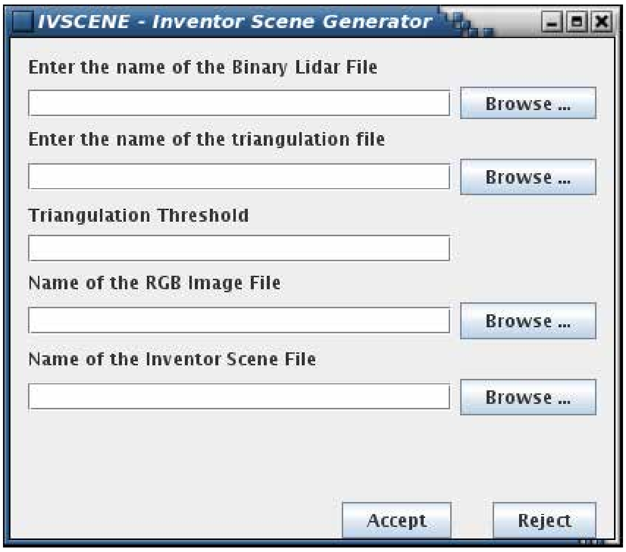

Figure 2: IVSCENE Interface

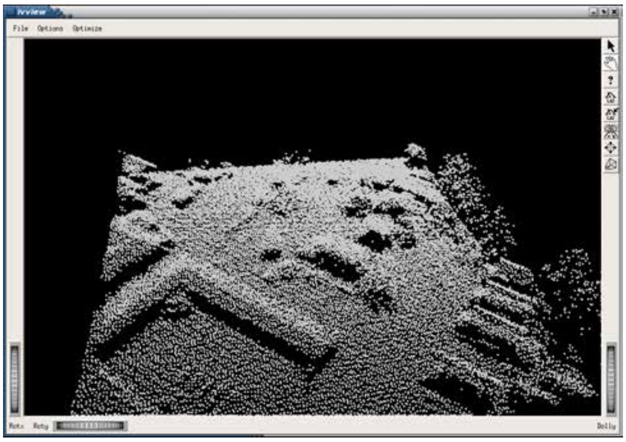

Figure 3: Point display in three dimensional space

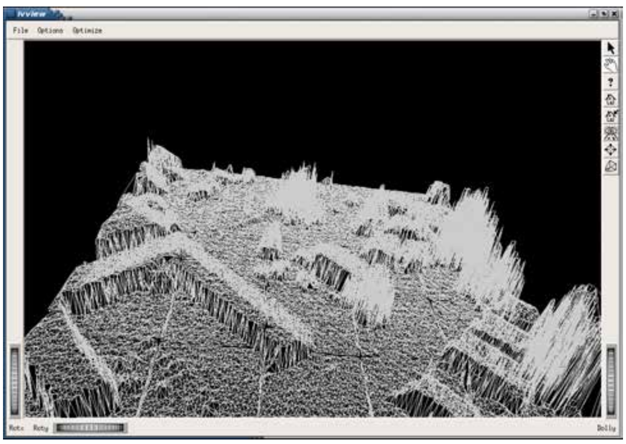

Figure 4: Wireframe based display in three dimensional space

The next step in visualization was with the filled up FACETS which hid the other facets and consquently the features behind. In the facet display, it was seen that the perception of the third dimension could be realised effectively. With a few adjustments for camera separation and viewer distance, the user can visualize the third dimension from multiple angles with the aid of the 3D shutter glasses. Figure 5 shows the tiled faced display.

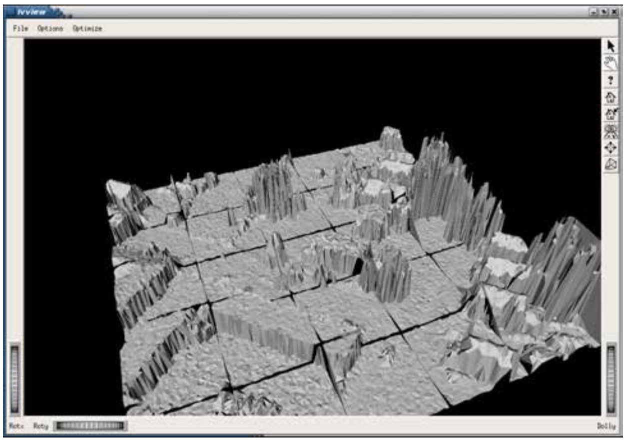

Figure 5: Tiled facet display in three dimensional space

It is seen in Figure 5 that the walls of the buildings in the space are not displayed properly. The threshold distance provided by the user is used to trim triangles off the Inventor Scene. The trimmed scene is seen in figure 6.

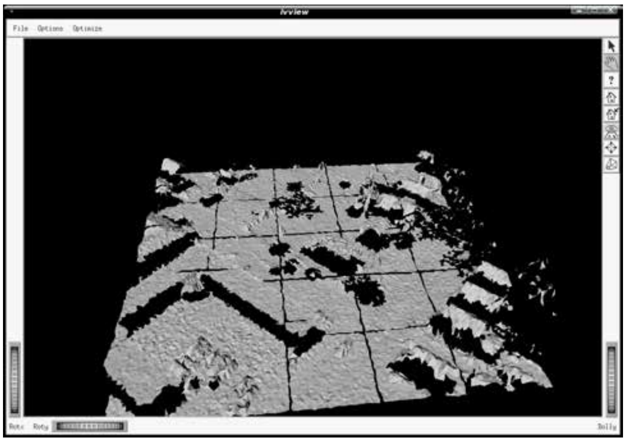

Figure 6: Thresholded triangles in three dimensional space

In the next step, the texture is wrapped on the scene in the options of OpenInventor on the thresholded scene. We can easily appreciate that, the scene appears cleaner but of course without the vertical facets (mostly walls) present.

We now run the DETECTOBJECT algorithm, which is presently under continuous development in the context of coding and fine tuning in terms of the parameters and restriction of the conditions. The data, when passed through the DETECTOBJECT algorithm generates a scene as shown in figure 8.

We can similarly, also turn on the option of overlaying the texture on the new scene generated, and then the scene is perceived as shown in figure 9. It is seen that small wall like artifacts, comparatively smoother than shown in figure 5 have come up where there were no facets present as in figure 6.

A fine tuning of the algorithm, with parameters and conditions is being continuously done for improving the restitution of the

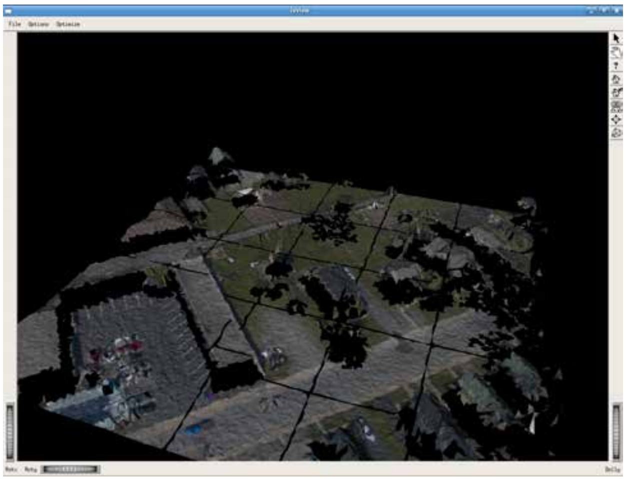

Figure 7: Texture Wrapped on thresholded triangles in three dimensional space

Figure 8: Initial object detection in the scene

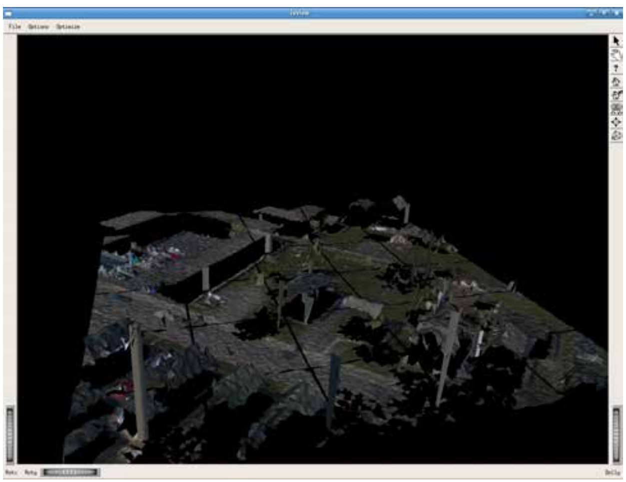

Figure 9: Texture Wrapped on Object detected scene

buildings initially. Following this stage, all planes which are present in the scene would be dealt with. The scene, thus would become more smoother and the speed of restitution would also be faster.

10 FUTURE DIRECTIONS

Our aim is to develop a set of algorithms which would enable realistic visualization of the terrain using LiDAR data and aerial photographs (Ghosh and Lohani, 2007a). The algorithms presented in the paper are the first step towards the end result wherein buildings and trees would be more realistic. Our approach is unlike the conventional approach where the data are classified in different object categories and then visualised (Haala and Brenner, 1997; Fujisaki et al., 2003; Forlani et al., 2003; Brenner, 2005). In our method we do not classify data in object types but aim to generalise data in geometric entities which are key attributes to visualisation. The presence of such systems with the aid of stereoscopic visualization techniques will enable making measurements and development of strategies for urban and emergency planning. Such systems will further cater to research like pattern exploration and spatial cognition.

ACKNOWLEDGEMENT

The authors are thankful to the Prof. Raj K. Thareja, Dean of Academic Affairs, IIT Kanpur, Prof. C. V. R. Murty, Head, Department of Civil Engineering and Prof. S. K. Jain, Dean Resource Planning and Generation, IIT Kanpur for arranging partial funds to support this research. The first author is also thankful to Mr. Rakesh Kumar Mishra, Project Associate at IIT Kanpur for his timely help in fine tuning the code.

References

Brenner, C., 2005. Building reconstruction from images and laser scanning. International Journal of Applied Earth Observation and Geoinformation 6, pp. 187–198.

Döllner, J., 2005. Geovisualization and real time 3d computer graphics. In: J. Dykes, A. M. MacEachren and M. J. Kraak (eds), Exploring Geovisualization, Elsevier Ltd., chapter 16, pp. 325–342.

Forlani, G., Nardinocchi, C., Scaioni, M. and Zingaretti, P., 2003. Building reconstruction and visualization from lidar data. In: The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XXXIV, Part 5/W12, Ancona, Italy.

Fujisaki, I., Evans, D. L., Moorhead, R. J., Irby, D. W., Aragh, M. J. M. and Roberts., S. D., 2003. Lidar based forest visualization: Modeling forest stands and user studies. In: ASPRS Annual Conference, Fairbanks, Alaska.

Ghosh, S. and Lohani, B., 2007a. 3d visualization of LiDAR data: state of the art and proposed system. GIM International.

Ghosh, S. and Lohani, B., 2007b. Development of a system for 3d visualization of lidar data. In: Map World Forum, Hyderabad, India.

Haala, N. and Brenner, C., 1997. Generation of 3-d city models from airborne laser scanning data. EARSEL Workshop on LIDAR Remote Sensing of Land and Sea, Tallinn, Estonia.

Haeberli, P., 1988. The SGI image file format. ftp://ftp.sgi.com/graphics/SGIIMAGESPEC. Last accessed April 15, 2007.

Lillesand, T. M. and Kiefer, R. W., 1999. Remote Sensing and Image Interpretation. 4 edn, John Wiley and Sons. ISBN 9971-51-427-3.

McCormick, B., DeFanti, T. and Brown, M. (eds), 1987. Visualization in scientific computing. Vol. 21(6), Computer Graphics.