

IMMERSIVE 3D VISUALIZATION OF REMOTE SENSING DATA

¹Surbhi Rautji, ²Deepak Gaur, ³Karan Khare

¹Oracle Technologies, Oracle 3C Building, Sector 127, Noida (UP), India

²Department of Computer Science and Engineering, AMITY University, Noida, India.

³Gradestack Pvt. Ltd, TLABS, Times Tower, Noida (UP), India

ABSTRACT

Immersive 3D Visualization is a java based Engine for viewing the Data of aerial images in the Three Dimensional with provision for simulation and fly through. This application is based on java Technology and works as standalone application, applet or in the browser. Immersive 3D Visualization is a good application to use where the area of interest is to view the interested regions rather than the big images. The Immersive 3D Visualization is a application that is designed on Java 3D Technology and Open Graphics Libraries. The Java3D Technology of envisages the three Dimensional view of the picture and Open Graphics Libraries allows one to view the rendered data on to the screen. This visualization application is java based hence no portability issues. The Data for this work is collected from the various sites like Google earth, USGS web site for the data viewing. This work takes an advantage of modelling the 3D DEM data on the visualized portion of the screen, and thus this approach is optimized. We are assumed that the DEM is coming from the other source or it is simulated Data in order to view the three-dimensional pictures. The Process of Collecting the Data and projecting on to the screen is done by collectively Open Graphics Libraries and Java 3D Technology. In this work any image can be viewed as 3D with the use of DEM data, which can be created with use of certain selected values, where as in the case of Google Earth it is not possible to see your own image in 3D. The work done here can be used for selected region of interest, unlike Google Earth, which is used for continuous streaming of data. Our work on 3D immersive visualisation can be useful to GIS analyst to view their own images in 3D.

KEYWORDS

3D Visualisation, Immersive, Image Representation, Indexed Color Mode, JAVA 2D, JAVA 3D, OpenGL

1. INTRODUCTION

With the rising number of sensors carried on board of planes and satellites and the increasing performance of these sensors new tasks originates, which are handled more efficiently by remote sensing techniques than by standard procedures. These tasks include data acquisition for geographic information system, refinement of data using higher level of details like DEM, update and verification of existing GIS database.

Software systems in yesteryears have gone through a great evolution and have allowed various highly developed techniques to facilitate the work spheres. One such significant development is generation of 3D city models. Still there is a continuing need for an improvement of these systems to enable a further reduction of human interaction, with an idea of fully automatic systems. However, it is becoming even more important to develop newer and more updated and efficient applications and tools to view and manage the acquired 3D city models. This paper is an example which gives an overview of the latest updates on generation and visualization of 3D site models.

A 3D site model generally includes a description of the terrain, of the streets, buildings and vegetation of the concerned built-up area. Although for several applications additional information is required, mainly buildings and man-made objects are important factors of a 3D site model. For instance, a realistic representation of such a virtual reality applications can be acquired only when the texture of the ground, roofs and facades is available as well as other significant details such as trees, footpaths and fences.

The purpose of this given work is technology of 3D-site model generation in PC-level computers. In this technology digital terrain models on large territory are created using space images, and man made objects and terrain models containing them are created using large scale aerial images.

1.1 Image Representation

The data that is acquired from various sources like satellites, aerial based images are of two dimensional in nature. These two dimensional image is a raster image where in it consists of pixels. Pixel is considered as a smallest element. The image is formed with the pixels. The data that is acquired from various aerial image will have the two dimensional representation of the image.

1.2 Two dimensional picture of an image

The image which was shown above is the two dimensional picture of the image. The image has gray values that are shown above. The image will have different colors. The various colors of the image are RGB – Red Green Blue, Indexed Color Mode. The image can be of 8 bit, 16 bit or 24 bit image. 8 bit image occupies the 1 byte, 16 bits is 2 bytes and 24 bit image is 3 bytes.

1.3 Three dimensional picture of an image

In order to represent the Three dimensional picture of the image, one needs the two dimensional picture of the image and a height information that is constructed with the elevation of the image. The elevation of the image is being constructed with the models Digital Elevation models and Digital Terrain Models. In the DEM the elevation is computed without surface information but in DTM surface models are taken into account.

1.4 Remotely Sensed Satellite Imagery

Remote sensing is the acquisition of information about an object or phenomenon, without making physical contact with the object. In contemporary parlance, the remote sensing generally refers to the use of aerial sensor technologies to detect and classify objects on Earth (both on the surface, as well as in the atmosphere and oceans) through the transmitted signals (e.g. electromagnetic radiation emitted from aircrafts or satellites).

These satellite images serve many purposes in the fields of meteorology, agriculture, geology, forestry, landscape, biodiversity conservation, regional planning, education, intelligence and warfare.

There exist two key categories of remote sensing: Passive Remote Sensing and Active Remote Sensing. The Passive Sensors are able to detect natural radiation that emit or reflect by the object or the surrounding area that is being observed. The most common source of radiation that passive sensors measure is Reflected Sunlight. Some examples of passive remote sensors are film photography, infrared charge-coupled devices and radiometers.

Entirely opposed to that, Active Remote Sensing emit energy that scan objects and areas and then a sensor detects and measures the radiation reflected or backscattered from the target. RADAR is one of the examples of Active Remote Sensing in which the time spent between the emission and return is measured, which establishes location, height, speed and direction of an object.

The Remote Sensing technology has allowed man to acquire data on and from hazardous and inaccessible areas. Remote sensing applications have been used to monitor deforestation in areas such as the Amazon Basin, glacial features in Arctic and Antarctic regions, and collecting information by means of waves (possibly sound) in coastal and ocean depths. During the cold war, the data collected ta about dangerous border areas was used by the Military. Remote sensing also replaces costly and slow data collection on the ground, ensuring in the process that areas or objects are not disturbed.

The quality of remote sensing data depends on its spatial, spectral, radiometric and temporal resolutions. The size of a pixel recorded in a raster image is called Spatial Resolution. The pixels match up to the square area which ranges in side length from 1 to 1,000 metres (3.3 to 3,300 ft). The wavelength width of the different frequency bands recorded is called Spectral Resolution. Generally, this is related to the number of frequency bands that the platform records. The number of different intensities of radiation that the sensor is able to differentiate between is called Radiometric Resolution. Usually, its range is between 8 to 14 bits, which corresponds to 256 levels of the gray scale and is up to 16,384 intensities or "shades" of colour, in each of the band. It also depends on the instrument noise.

1.5 Digital Elevation Model

A digital representation of ground surface topography or terrain is called its digital elevation model (DEM). Generally, it is also known as a digital terrain model (DTM). A DEM can be symbolised by a raster (a grid of squares) or by a triangular irregular network. Frequently DEMs are constructed using remote sensing techniques, but, they may also be made from land survey. Geographic Information Systems use DEMs quite often and are the most regularly used for digitally creating relief maps.

The data preparation for this work depends on the selection of the raster image and its selection of DEM Data. The DEM data is also raster data. This Data will give the height information. Data preparation model for this work depends on the proper selection of the Digital Elevation Model for the image.

In Order to find the correct height information value one should study about the Digital Elevation model generation. The height information for the Raster images is generated using the satellite imageries. There are various mathematical modelling techniques are available for the DEM generation.

DEMs may be created in a several ways, but often they are acquired by Remote Sensing rather than using direct survey. Interferometric Synthetic Aperture Radar is one of the dominant techniques for generating digital elevation models. Two passes of a radar satellite (for example RADARSAT-1) are required to generate a digital elevation map for converting from a resolution of ten meters through ten kilometres. One can also acquire an image of the surface cover.

Digital image correlation method is also a very significant and powerful technique for creating a Digital Elevation Model. It involves two optical images obtained with different angles that have been taken from the same pass of an airplane or an Earth Observation Satellite (for example the HRS instrument of SPOT5).

Former methods of creating DEMs generally involve interpolating digital contour maps that could have been generated by direct survey of the land surface; the method is still being used in mountain areas where superimposition of waves or interferometry is not always as desired. It is important to keep in mind that the contour data or any other sampled elevation datasets (by either GPS or ground survey) do not fall under the category of DEMs, they can be accepted as Digital Terrain Models. A DEM signifies the continuous availability of elevation at each location in the area of the study.

The quality of a DEM represents the accuracy of elevation at each pixel (absolute accuracy) and the accuracy of morphology presented (relative accuracy). The quality of DEM derived products depend on various factors that such as roughness of the terrain, sampling density (elevation data collection method), grid resolution or pixel size, interpolation algorithm, vertical resolution and terrain analysis algorithm. The common uses of DEMs include creation of relief maps, modeling water flow or mass movement (for example avalanches), extracting terrain parameters, rendering of 3D visualizations, creation of physical models (including raised relief maps), rectification of aerial photography or satellite imagery, reduction (terrain correction) of gravity measurements (gravimetry, physical geodesy), terrain analyses in geomorphology and physical geography.

1.6 OpenSceneGraph

For development of high performance graphics applications viz. flight simulators, games, virtual reality and scientific visualization, the OpenSceneGraph is an Open Source library and cross-platform graphics toolkit. It is based on the concept of a SceneGraph, which provides an object oriented framework on top of OpenGL. This gives the developer the scope to have many additional utilities for a fast development of graphics applications as well as sets the developer free from implementing and optimizing low level graphics calls.

1.6.1 Features

The main objective of the OpenSceneGraph is to make the benefits of scene graph technology freely available to everyone, including commercial and non-commercial users. The program is written completely in Standard C++ and OpenGL, and it fully utilizes the STL and Design Patterns. It also capitalizes on the open source development model to provide a development library that is legacy free and focuses on the needs of end users.

The principal strength of OpenSceneGraph is its performance which includes supporting view-frustum culling, occlusion culling, small feature culling, Level of Detail (LOD) nodes, OpenGL state sorting, vertex arrays, vertex buffer objects, customization of the drawing process, and OpenGL Shader

Language and display lists is one feature of core scene graph while Productivity is the feature of core scene graph that lets encapsulates the majority of OpenGL functionality including the latest extensions, provides rendering optimizations features such as culling and sorting, and set of add-ons generally libraries which makes it possible to develop high performance graphics applications very rapidly, scalability with the scene graph is that it will not only run on portables all the way up to high end multi-core, multi-GPU systems and cluster, portability of core scene graph has been designed to have minimal dependency on any specific platform.

2. RELATED WORK

Interest in 3D site models has risen significantly in the past years. Originally, it was considered one of the main application areas to simulate the circulation of electromagnetic waves. These are

used by network operators for the planning of antenna locations. Although, it is quite likely that there may be further need in the coming times due to the approaching introduction of UMTS networks in Europe, there are other areas evolving as well, such as 3D navigation systems and visualization for the city and building planning or architectural contests.

Geographic Information System (GIS) are widely accepted tools for geodata visualization and analysis. Many GIS application suffer from the lack of basic geospatial data. As acquiring geodata is quite time intensive and resource intensive methods for automated object recognition in raster images are indisputably beneficial and therefore of interest for any GIS application. Along with this point of view goes the effect that the basic functionality of a GIS is not only visualization but also interaction and analysis on geospatial objects in connection with highly sophisticated scientific visualization methods for large datasets. Many approaches simply cover 2D aspects only (common GIS approach) or are designed to support 3D data output. But, recently more and more GIS are combined with virtual reality environment, taking into account the GIS typical data analysis and decision support functionality, which has to involve highly sophisticated 3D interaction facilities.

Common to all these GIS applications is that they suffer from the lack of suitable set of basic geospatial data. Especially in applications which involve decision support in urban planning, up-to-date information about buildings, traffic infrastructure and other man made features.

The difference that exists between this work and Google Earth is that, Firstly; Google Earth requires only its own specific height information data and cannot use different height information while in our work this is no limitation in our work since any height data can be used i.e. Digital Elevation Modal. Secondly in our work any image can be used to get its 3D output with the use of DEM data while in the case of Google earth not every image can be used to produce the 3D output.

3. CLASSIFICATION IN 3D IMMERSIVE VISUALIZATION

3.1 JAVA 2D Programming

Java 2D is a client side graphics programming. The Java 2D programming is meant for two dimensional graphics representations. The various forms in which this programming is done is either Frame based applications which can be run as a standalone applications and Applet Based applications which can be run in the browser.

The advantages of Java 2D programming is of its ease of use and java's platform and architectural neutral. An application programming Interface (API) was developed at Sun Micro Systems called Java 3D for rendering interactive 3D graphics employing Java Programming language. Java 3D is a client side Java API and it includes the abstract Windows Toolkit (AWT) and Java Foundation classes(JFC/Swing), which are both java class libraries for creating applications with a Graphical User Interface(GUI).

Java 3D depends on OpenGL or DirectX for its functioning of native rendering, whereas the 3D scene description, application logic, and scene interactions are located in Java code. When Sun decided to design Java 3D, although they did not have the required means and resources or the support from the industry to substitute OpenGL, they still wanted to pull more of its strengths as an object oriented programming (OOP) language in place of merely entrusting it to a procedural language such as C. Whereas OpenGL's description for a 3D scene comprises of points, lines and abstraction, Java 3D is able to describe a scene as a collection of Objects. In raising the level of description and abstraction, Sun applied OOP principles to the graphics domain on one hand and

on the other, also brought in scene optimizations that can make up for the involvement and increasing the overhead by means of calling through JNI.

3.2 JAVA 3D API

The Java 3D API is a ladder of Java classes which act as the interface to a refined three dimensional graphics rendering and sound rendering system. The programmer employs high-level concepts for creating and manipulating 3D geometric objects. These geometric objects are located in a virtual space, which is then rendered. The API is planned with the elasticity to create precise virtual spaces of a great variety of sizes from astronomical to subatomic.

A Java 3D program produces instances of Java 3D objects and situates them into a scene graph data structure. The scene graph is an arrangement of 3D objects in a tree structure that fully specifies the content of a virtual space, and how it is to be rendered.

3.3 Building a Scene Graph

A Java 3D virtual space is created from a *scene graph*. A scene graph is produced using instances of Java 3D classes. The scene graph is put together from objects to define the geometry, sound, lights, location, orientation, and appearance of visual and audio objects.

A common definition of a graph is a data structure composed of nodes and arcs. A node is a data element, while arc is the relationship between data elements. The nodes in the scene graph are the instances of Java 3D classes. The arcs represent the two kinds of relationships between the Java 3D instances.

The most universal relationship is a *parent-child relationship*. A group node can have any number of children but only one parent, whereas leaf node can have one parent and no children. The other relationship is a *reference*. A reference associates a *Node Component* object with a scene graph Node. The geometry and appearance attributes used to render the visual objects are defined by the Node Component objects.

A Java 3D scene graph is built of Node objects in parent-child relationship in a tree structure. In a tree structure, one node is the root. Other nodes are accessible following arcs from the root. The arcs of a tree form no cycles. A scene graph is formed from the trees rooted at the Locale objects. The Node Components and reference arcs are not part of the scene graph tree.

In a tree, there is only one passageway from the root of a tree to each of its leaves; similarly, there is only one passageway from the root of a scene graph to each leaf node. The passageway from the root of a scene graph to a specific leaf node is called the leaf node's scene graph path. Since a scene graph path leads to exactly and only one leaf, there exists one scene graph path for each leaf in the scene graph.

Each scene graph path in a Java 3D scene graph specifies fully the state information of its leaf. State information includes the following: location, orientation, and size of a visual object. As a result, the visual attributes of each visual object depend only on its scene graph path. The Java 3D renderer capitalizes on this fact and renders the leaves in the order it considers to be most efficient. The Java 3D programmer usually does not have control over the rendering order of objects.

To design a Java 3D virtual space a scene graph is made using the standard set of symbols. After the design has been completed, that scene graph drawing is the specification for the program.

After the program is complete, the same scene graph is a concise representation of the program. The scene graph can be created through a program which can be generated using an existing scene graph. A scene graph drawn from an existing program document the scene graph the program creates.

3.4 JAVA 3D Advantages and Strengths

Java 3D allows the programmers to program in 100 percent java code. In any significant 3D application, the rendering code composes only a part of the total application. It is very good to have the application code, persistence, and user interface (UI) code in an easily portable language, such as Java. Although Sun's promise of Write-Once-Run-Anywhere was considered more to be marketing strategy than a realistic claim. Especially for the client side programming, java has done significant work to enable development of applications that can be easily moved between platforms. The platforms of a wide range of interest today are Microsoft Windows 98/NT/2000, Sun Solaris, LINUX, and Macintosh OS X.

Many of the latest 3D graphics applications being developed with java 3D today are capitalizing on the strengths of java as the language for the internet.

Java 3D includes a view representation designed for use with head-mounted displays (HMDs) and screen projectors. By protecting the programmer from much of the complex trigonometry required for such devices, java 3D simplifies the transition from a screen centric rendering model to a projected model, where rendering in stereo allows for greater practicality.

Java 3D also includes in-built support for sampling 3D input devices and rendering 3D spatial sound. By combining the above elements into a unified API, java 3D has achieved a design uniformity that few other APIs can claim.

Java 3D's higher level of concept of rendering the scene also has the scope of interactive 3D graphics for a new class of audience people who would typically have been considered 3D content creators. It may appear the 3D graphics as a spectrum, with functional resources and capabilities distributed across a variety of tasks.

3.5 Limitations of Java 3D

There are some OpenGL features that are difficult or not possible to achieve within Java3D for some programmers coming from OpenGL. Some of them may actually miss the total control they normally have over the scene and the rendering process. Many others, however, quickly learn mapping from OpenGL functions to java 3D Objects and even appreciate the output gains they can have using java 3D.

A skilled developer using OpenGL and native C code will be able to achieve higher performance than a Java Programmer who uses Java 3D despite the fact that java 3D includes some clever optimizations. However, if absolute rendering performance is the top-main concern of the application then it would be advisable to use OpenGL or another native rendering API.

4. DESIGN AND IMPLEMENTATION

3D lets users view regularly gridded DEM data as a continuous topographic surface with colored contours, shading, or overlaid physical, cultural or environmental data. The program facilitates for the users viewing of the texture of the gridded surface and the variation of relief. The map area can be examined with different light sources, from varying directions and heights above the

horizon. It is possible to rotate the maps 360 degrees around a vertical axis and 90 degrees around a horizontal axis. Map projections can be exaggerated or zoomed.

Our work has been implements and designed on Java 3D Technology. The implementation works on Java 3D and uses the Open Graphics Libraries for visualization.

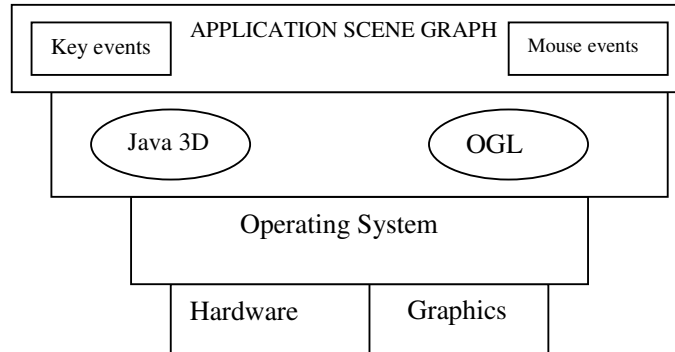


Fig: 1 - Architecture of the application

4.1 3D Terrain Visualization with Flythrough Scenes

To appraise the condition of the terrain in areas whether remote or urban , to detect any potential operational problems, and to ensure that radio communication and telemetry signals are maintained, DEMs and high-resolution Orthorectified Satellite Images are employed to generate a representation of the local terrain conditions. One can manage all this without having to step out of one's office.

Satellite Imaging Corporation is able to provide its customers with 5m digital surface models (DSMs), 2m digital elevation models (DEMs), and a detailed orthorectified satellite image mosaic at 0.8m resolution by utilizing stereo IKONOS satellite image data.

DEMs can be generated from the HRS SPOT-5 satellite sensor with a resolution of 20m and with a resolution of 15m from the stereo ASTER satellite sensor. Explorers and operators can achieve a substantial improvement in the visualization of surface and exploration targets by this data that provides 3D imagery.

The benefits are quite evident, such as in the field of oil and gas exploration. A combination of the described process and the interpreted seismic data in the vertical dimension can provide a greater understanding of the relationship between surface and subsurface structures at both basin and prospect scale. When viewed in an immersive 3D visualization environment, this understanding is more enhanced. These services are exceptionally helpful for project planners, operation managers, and logistics managers in order to plan field operations in a computer environment, ensuring the best access and project objectives' achievement for engineering and construction of pipeline, transmission, road design, oil and gas exploration and production activities, 2D/3D seismic data acquisition, environmental studies, motion pictures, aviation, media and so on.

High resolution geometrically corrected Satellite Image Data has to be available or obtained from the 0.62m QuickBird, 0.81m IKONOS or 2.5m SPOT-5 satellite sensors so as to provide 3D Terrain Modeling and visualization capabilities in a GIS or mapping computer environment. In certain instances, use of digital orthoairial photography is also possible.

Availability of an accurate digital terrain model (DTM) is a must if a 3D terrain modeling and visualization application is expected to have good details about the terrain features and terrain-slopes to facilitate critical project decisions. If, in the Area of Interest (AoIs), there is no appropriate DTM available then a DTM can be obtained by certain satellite and aerial sensors such as LiDAR (Light Detection and Ranging), Stereo Aerial Photography, IFSAR (Interferometric Synthetic Aperture Radar) and Stereo High Resolution Satellite Image Data. These remote sensors have been listed in an orderly manner based on their resolution, capability and accuracy.

4.2 Image Data or Texture Data

The data that is to be projected on the application is given by the Image Data or Texture Data. Sample Data Sets of the Image or Texture Data is shown below



Fig: 2- Texture Sample 1



Fig: 3- Texture Sample 2

The Simulated Digital Elevation data is shown in the below picture.

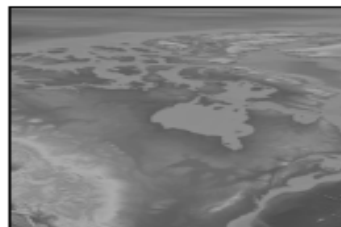
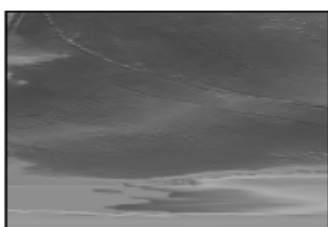


Fig: 4- Simulation Digital Elevation data

After the Data is prepared, the next process involves the draping and rendering process. The Diagram of the draping the Texture image on the Elevation is shown below.

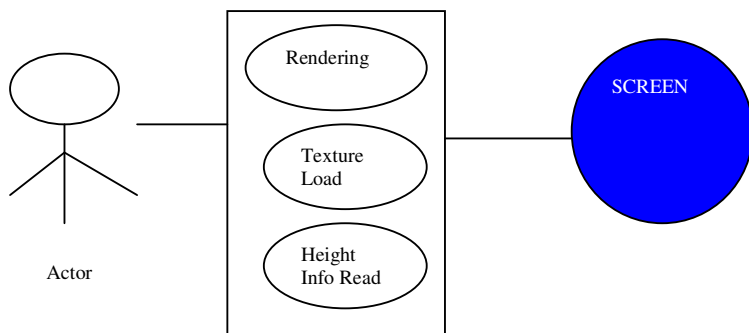


Fig: 5- Draping the Texture image on the Elevation

4.3 Process Diagram

The diagram explains the step by step of each sequence taking place in the system. The sequence diagram for the work implementation is shown below

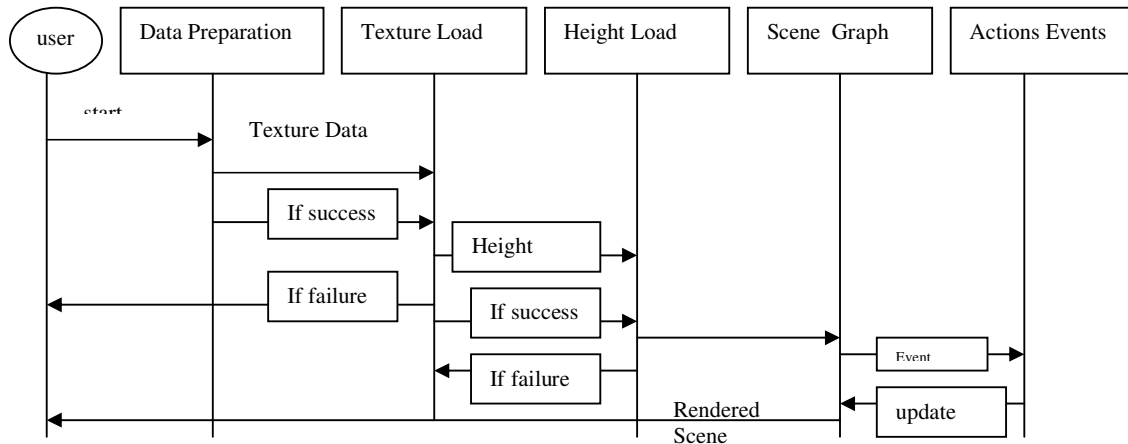


Fig: 6- Process Diagram

The 3D Visualization tool follows a sequence of steps for data representation starting with the Data Preparation where in the data sets are collected then the collected data sets are Texture Data and DEM (Digital Elevation Model) Data. The texture Data will give the image information and the Height information is given by the elevation data. This elevation information forms the mesh. The data after preparation will be given to the scene Graph. The Scene Graph is the Open Graphics Library function that calls the method to render the graph on to the screen. The various events are attached to the scene Graph. The events are of two types: Keyboard events and Mouse Events. Based on the selection of events the scene graph gets updated on to the screen. The scene graph is update continuously based on the selection of events.

5. DISCUSSION

The result of this application created is 3D visualization of satellite image. For generating a 3D satellite image, first the data is prepared where in the data sets are collected. The collected data sets are Texture data and DEM (Digital Elevation Model) data. Texture data will give the image information and DEM data will give the height information. Now we render these two information to generate a scene graph and thus the 3D visualization of a satellite image is generated. This scene graph is generated using the concept of Java 3D. In this application work we have implemented some functionality like forward, backward, left, right, etc.

By changing the YAW i.e. the z value in the co-ordinate system user changes the height information. By reducing the height we are coming near to the object. In application we can forward the satellite image by pressing 'W' Key. By increasing the height we are going away from the object. In application we can Backward the satellite image by pressing 'S' Key. By changing the ROLL i.e. the x value in the co-ordinate system users changes the x direction. Value Decrease in x direction moves the image in the left direction. In application we can move satellite image in left direction by pressing 'a' key. Value increase in x direction moves the image in the right direction. In application we can move satellite image in right direction by pressing 'd' key.

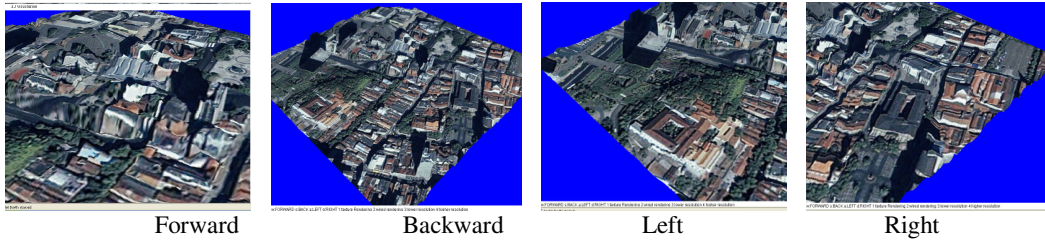


Fig: 7 -3D Visualization of satellite image moving in various directions

Texture Rendering is a process of displaying the satellite image from the frame buffer to the display. Frame buffer is the memory area assigned in the graphic card which controls the display information on the screen. Each pixel on the screen occupied is memory location in the frame buffer. In application we can see texture rendering process by pressing the numeric key '1'.



Fig: 8– Texture rendering of a Satellite image

To represent any feature properly we need to create a TIN (triangular irregular node) there we create a mesh by joining a tin. To represent a surface we need minimum 3 points because 4 points cannot represent a surface. DEM data is available in image at some intervals; let's say DEM data is available in image after every 100 pixels (100x100). We take a point (x, y) at a particular interval and then take DEM information. Now we take these 3 points i.e. coordinates of x direction, y direction and height information of each point to form a triangle. Same process is repeated for the complete image. These triangles jointly describe as wired mesh. Wire mesh is used to locate and identify any specific location which is difficult to pinpoint in texture rendered satellite image display. In application we can see texture rendering process by pressing the numeric key '2'.

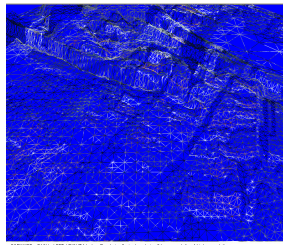


Fig: 9 – Wire rendering of satellite image

In high resolution terrain/object is seen from a near distance we see a high resolution data. From higher z value to lower z value we see more detail of data. In this application we can see high resolution by pressing the numeric key '3'. In low resolution terrain/object is seen from a farther distance we see a low resolution data. From lower z value to higher z value we see less detail of data. In this application we can see high resolution by pressing the numeric key '4'.

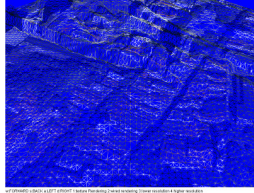


Fig: 10- High resolution of a Satellite image

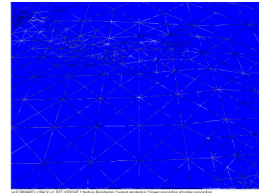


Fig: 11- Low resolution of a Satellite image

6. CONCLUSION

We have presented geometrical and topological models for visualization of terrains and man-made features in satellite and aerial imagery. With an idea of visualization we have developed a robust and optimised engine for the three dimensional view of the image. It is based on the Java 3D technology and uses open graphics libraries for data view and visualization. The advantage of this work is it can be used for a selected portion of data for optimization. This work is based on the java Technology hence no portability issues are involved. Generation of 3D site models is being promoted commercially with main use in the private sector. However, research activities on 3D site models which support the commercial activities are immature at present, though several interesting research activities have been and are still performed.

7. FUTURE PROSPECTS

This model still have a lot of future to be accomplished i.e. Mouse Events Screen is to be added to applet, mouse listeners are to be designed extensively and have to be tested, the option should be given in the application to add different resolution of satellite images in the respective layer to enhance the level of details, Performance Improvements are to be tested, the application is to be tested for various data sets collected from Google earth, look and feel of the application is to be improved, testing of the application is to be rigorously made.

REFERENCES

- [1] DONALD, A.M., 2003, *Mathematical Methods for Scientist and Engineers* (University Science Books).
- [2] FONESKA, L.M.G. and MANJUNATH, B.S., 1996, Registration techniques for multi-sensor remotely sensed imagery. *Photogrammetric Engineering and Remote Sensing*, 62, pp. 1049–1056.
- [3] SLATER, P.N., 1992, Optical remote sensing systems. In *Space Oceanography*, Arthur P. Cracknell (Ed.) (World Scientific), pp. 13–33.
- [4] SABINS, F.F., 1987, *Remote Sensing: Principles and Interpretation*, 2nd edition (W.H. Freeman and Company).
- [5] SCHOTT, J.R., 1997, *Remote Sensing: The Image Chain Approach* (Oxford University Press).
- [6] JOSEPH, G., 2003, *Fundamentals of Remote Sensing* (Hyderabad, India: University Press).
- [7] LILLESAND, T.M. and KIEFER, R.W., 1995, *Remote Sensing and Image Interpretation*, 4th edition (New York: John Wiley and Sons).
- [8] Li, H. 1993. *Multi Sensor Image Registration and Fusion*, PH.D thesis, University of California, Santa Barbara.
- [9] Li, H., B.S. Manjunath, and S.K. Mitra, 1995. An approach to multi sensor image registration, *IEEE transactions and Image Processing*, 4(3):320-334.
- [10] Brown, L.G., 1992. A survey of image registration technique, *computing surveys*, 24(4):325-376.
- [11] Takeuchi, S., 1993. Image registration between SAR and TM data using DEM and slant range information, *Proceedings of IGARSS - IEEE International Geoscience and Remote Sensing Symposium*, Tokyo, Japan, 3:1351-1353.
- [12] Goshtasby, A., 1986. Piecewise linear mapping functions for image registration, *Pattern Recognition*, 19(6):459-466

- [13] Zheng, Q., and R. Chellappa, 1993. A computational vision approach to image registration, IEEE Transactions on Image Processing, 2(3):311-326.
- [14] Djamdji, J.P., A. Bijaoui, and R. Maniere, 1993. Geometrical registration of images: the multiresolution approach, Photogrammetric Engineering b Remote Sensing, 59(5):645-653
- [15] Ton, J., and A.K. Jain, 1989. Registering Landsat images by point matching, IEEE Trans. on Geoscience and Remote Sensing, 27(5):642-651.

Authors

Surbhi Rautji is a Software Engineer in Oracle Technologies. She has completed her Bachelor of Technology in Computer Science and Engineering from Amity University, Noida, India. Her fields of interest include Image Processing, Image security, Artificial Intelligence, Cognitive Studies, Genetic Algorithms and Computer Network Security.



Deepak Gaur had received Master of Engineering in Computer Science & Engineering from Punjab Engineering college, University of Technology, Chandigarh. He has completed his B.Tech in Computer Science & Engineering from Himachal Pradesh University, Shimla(H.P). Presently he is working as Assistant Professor in CSE Department, Amity University, Noida, Uttar Pradesh, India. Mr. Deepak Gaur, research Area is Image Processing, Image Compression, Security Systems, Image Security and pattern reorganization.



Karan Khare is a web developer and security specialist in Gradestack Pvt. Ltd. He has completed his education in Bachelor of Technology in Computer Science and engineering from Amity University, Noida. He holds interest in Artificial Intelligence, Genetic Algorithms, Cognitive Studies, Computer Network Security and Image Processing.

