3D Geo-data Management and Query Optimization in GeoRDBMS in Multi-User Access

Thesis submitted to the Andhra University, Visakhapatnam
in partial fulfillment of the requirement for the award of
Master of Technology in Remote Sensing and GIS

Submitted by

Vaibhav Kumar
M.tech. (Geoinformatics)

Supervised by

Dr.Harish Karnatak
Sci./Engr.-'SE'

Indian Institute of Remote Sensing, ISRO,
Dept. of Space, Govt. of India Dehradun – 248001
Uttarakhand, India

August, 2014

ABSTRACT

In the emerging era of Information and communication technologies, geotechnology is one of the fastest-growing fields. GeoRDBMS is a very important and evolving aspect for GIS, as it can manage a large volume of spatial data inside GeoRDBMS environment. The utilization of RDBMS for Geospatial data is one of the vital focus of GIS professionals in the last decades to store and manage 2D geo-data. However, the support for 3D geo-data inside RDBMS is still limited and is a challenging task for GeoRDBMS providers. The study combines the concept of augmented reality and GeoRDBMS to build new exciting applications in the field of urban development, disaster management etc. In this study, various file-based open standard 3D data models like CityGML, COLLADA and KML, these 3D data models are migrated to GeoRDBMS to bring the entire 3D geo-data in a common platform. Organization and management of 3D Geo-data are done in Geo-RDBMS using created 3D geo-data in various file formats and attribute tagging with 3D data is performed for spatial and nonspatial queries in a single and multiuser environment. These 3D data models contain data of a very large amount and high complexity so special attention is given to manually optimize the GeoRDBMS after migration of 3D geo-data. These 3D models are created in various 3D file formats like KML, CityGML, COLLADA. In this study, new spatial operations and software modules are created to perform attribute and location-based queries on 3D geo-data by developing front end applications. These spatial operations are further combined with newly developed software modules to bridge the gap between useful information extraction by the user e.g. segmentation of a 3D building and applying separate query over the segmented blocks or creation of actual 3D floors of a building based on attribute values of different floors. Performance study of various spatial operations over created 3D geo-data in GeoRDBMS is done in single and multiuser environment, and its results along with other important considerations are applied to create an automatic optimization environment for GeoRDBMS by using developed front end software to enhance query execution time by operating on various optimization parameters and spatial indexes. A plethora of software have been developed for processing and visualizing 3D geo-data, but its visualization on the web is still a challenging area. To overcome this limitation, this research also focuses on utilizing HTML5 and WebGL to develop virtual globes to visualize 3D geo-data. By applying such an approach, 3D capabilities can be realized directly in the browser without any need for an additional plug-in. Results obtained in this study can be very useful in the further study of the field of 3D GIS and more options can be provided to the user to extract important information from the 3D Geo-data by combining GeoRDBMS and augmented reality.

Keywords: GeoRDBMS, Spatial index, 3D modelling, 3D geo-data, 3D attribution, Querying, Virtual globes

Contents

List of Figures

List Of Tables

1. Introduction

DBMS is a computer based record keeping system whose overall responsibility is to store and manage data with various data access mechanism. Database management system, or DBMS, is a computer software program that is designed as a means of managing all databases that are currently installed on a system hard drive or network. Different types of database management systems exist, with some of them designed for the oversight and proper control of databases that are configured for specific purposes. In spatial database, the spatial data types are usually defined as Abstract Data Types (ADT), i.e. encapsulated types together with spatial operations(Chen, Abdul-Rahmana, and Zlatanova 2008a). GeoRDBMS extends a traditional business database system by incorporating spatial data types in its data model and query language. Furthermore, it supports spatial data types in its implementation, including spatial indexing(Breunig and Zlatanova 2011a). Concepts developed in the fields of geo-data bases or spatial databases may be applied to both 2D and 3D geo-applications. GeoRDBMS is very important and evolving aspect for 3D GIS, as large volume of spatial data can be managed inside GeoRDBMS environment and also it can store semantics similar to CityGML for 3d geo-data(Chen, Abdul-Rahmana, and Zlatanova 2008a)(Breunig and Zlatanova 2011a).

In this study various file based open standard 3D data models like CityGML, COLLADA and KML, these 3D data models are migrated to GeoRDBMS to bring entire 3D geo-data in a common platform. Organization and management of 3D geo-data is done in GeoRDBMS using created 3D geo-data in various file formats and attribute tagging with 3D data is performed for spatial and non spatial queries in single and multiuser environment. These 3D data models contains data of very large amount and high complexity so special attention is given to manually optimize the GeoRDBMS after migration of 3D geo-data. These 3D models are created in various 3D file formats like KML, CityGML, COLLADA.

CityGML is an open data model and XML-based format for the representation and exchange of virtual 3D city models(Kolbe 2009). As per open geospatial standards KML is an XML language focused on geographic visualization, including annotation of maps and images and exchange of 3D models. COLLADA is an XML-based schema to exchange 3D assets between applications enabling diverse 3D content processing tools to develop various exiting applications(Coumans and Victor 2007). Due to large volume and complex data structure the 3D geo-data needs special attention while storing in GeoRDBMS to achieve expected performance during data access and query. Spatial indexing techniques play an important role in handling large and complex data and for better performance of spatial queries, enhancing the performance of GeoRDBMS. Spatial indexes also play critical role in allowing effective access of these data sets in single and multi-user environment.

1.1 Background

1.1.1 Evolution of 3D

For many years data acquisition techniques and computational processes had their practical limitations, and evolved continually in the use of 3D information. In most of the cases and especially in urban contexts the evolution to real world 3D geo-objects is slow. This can be explained by a factor like the 2D way of thinking. The primary reflex when upgrading a 2D model, for example the cadastral model, may be to keep the 2D object's definition and add some 3D extensions. This approach leads to unsatisfactory results because of the incomplete and limitative approach. Working with 3D data allows us to consider the 3D world where many objects can significantly evolve. If an object has a new definition strongly related to 3D, the use of 3D model will be imperative by itself (Siyka Zlatanova 2003).

1.1.2 3D GIS and its Applications

A geographic information system (GIS) is a system designed to capture, store, manipulate, analyze, manage, and present all types of geographical data. Extended capabilities of GIS to build, visualize, and analyze data in 3D is 3D GIS.3D GIS applications are very useful in urban city planning, Disaster and emergency management, environmental monitoring, simulation and visualization in GIS etc.

1.1.3 GeoRDBMS

In the emerging era of technologies geo technology is one of the fastest growing fields. GeoRDBMS is very important and evolving aspect for GIS, as it can manage large volume Of spatial data(Breunig and Zlatanova 2011b)(Chen, Abdul-Rahmana, and Zlatanova 2008b).Currently geospatial is stored in RDBMS environment. GeoRDBMS are allowing various spatial functionalities which leads to various exciting applications. geo-databases serve as platform to integrate 2D maps, 3D models, and other geo-referenced data. However, current geo-databases do not provide sufficient 3D data modeling and data handling techniques other than visualization and simulation(Chen, Abdul-Rahmana, and Zlatanova2008b). New 3Dgeo-databases is needed to be developed for 3D models.

1.1.4 Virtual 3D Data Models

CityGML is an OGC standard to represent and exchange city models in an interoperable way(Kolbe 2009)(Kolbe, Nagel, and Herreruela 2013). CityGML is an open data model and XML-based format for the representation and exchange of virtual 3D city models. It is based on the Geography Markup Language version 3.1.1 (GML3) (Kolbe 2009)(Kolbe, Nagel, and Herreruela 2013). KML is an XML language focused on geographic visualization, including annotation of maps and images. Geographic visualization includes not only the presentation of graphical data on the globe, but also the control of the user's navigation in the sense of where to go and where to look. COLLADA defines an XML-based schema to make it easy to transport 3D assets between applications(Coumans and Victor 2007) .These standards address the way how to efficiently and nicely visualize 3D models and how to interact with them.

1.1.5 Spatial Indexing in GeoRDBMS

Spatial Indexes are what make using a spatial database for large data sets possible. Without indexing, any search for a feature would require a sequential scan of every record in the database which leads to slow performance of data access. Indexing speeds up searching by organizing the data into a search tree which can be quickly traversed to find a particular record(Sardadi et al. 2008a)(Vanichayobon and Gruenwald 1999). Various Spatial indexing techniques like B-tree for data can be sorted along one axis, R-tree considers every object having a Minimum Bounding Box(MBB) and further broken up into rectangles, sub-rectangles and subrectangles. Generalized Search Tree (GiST) index is template data structure for abstract data types .GiST is a template for implementing other indexing methods, such as B-tree and RTree, and is a balanced tree structure that contains <key,pointer> pairs.

The key is a member of a user-defined class that represents an attribute valid for all items that the pointer element can reach. A key in an R-tree like GiST refers to a bounding box (Schön et al. 2009). Due to large volume and complex data structure the 3D geo-data needs special attention while storing in GeoRDBMS to achieve expected performance during data access and query.

Spatial indexing techniques play an important role in handling large and complex data and for better performance of spatial queries, enhancing the performance of GeoRDBMS. Spatial indexes also play critical role in allowing effective access of these data sets in single and multi-user environment.

1.1.6 Client/server Computing for Database Systems

Client /server computing is a model in which geospatial data is accessible in shareable environment. Server acts as a service provider for the clients who request for certain resources over internet(Karnatak et al. 2007). The web GIS is an extension and application of client/server computing, where the geospatial data is accessible in a shareable environment. Client/server computing describes a model for computer networking that offers an efficient way to provide information and services to concurrent user(s) at the same time. Internet is a "connectionless" process, based on client/server architecture. In a client/server model, a client is defined as a requester of services and a server is defined as the provider. The client/server software architecture is a versatile, message-based and modular infrastructure that is more flexible, easier to use, interoperable and scalable than centralized, mainframe, time sharing computing (Karnatak et al. 2007).

1.1.7 3D geo-data Creation in GIS

3D geo-data in current scenario can be created various 3D GIS tools like *GoogleSketchup*, *EsriCityEngine*,*ArcScene*.3D geo-data is also generated by terrestrial photogrammetry techniques.

1.1.8 Spatial Queries

A spatial query is a special type of database query supported by spatial databases. The queries differ from SQL queries in several important ways. Two of the most important are that they allow for the use of geometry data types such as points, lines and polygons and that these queries consider the spatial relationship between these geometries. There are 3 types of Spatial Queries

Spatial Range Queries

- Find all cities within 50 miles of Delhi
- Query has associated region (location, boundary)
- Answer includes overlapping or contained data regions

Nearest-Neighbour Queries

- Find the 10 cities nearest to Delhi
- Results must be ordered by proximity

Spatial Join Queries

- Find all cities near a lake
- Expensive, join condition involves regions and proximity

1.1.9 Revolution of 3D over 2D

3D representations of the real world allow a more direct connection between information environments and their electronic representations(Tavanti and Lind 2001). 2D representations are still abstract in that they require the user to learn certain conventions, since they do not actually resemble the things they refer to (Tavanti and Lind 2001). Considering a situation where a mountain is represented as 2D data where it is very difficult to relate the model with the actual scenario as compared to a 3D model of the same mountain. Also,

a) 3D displays can be exploited to visualize large sets of hierarchical data.

b) The perspective nature of 3D representations makes it possible to show more objects in a single screen, objects shrink along the dimension of depth.

c) If more information is visible at the same time, users gain a global view of the data structure.

1.1.10 Virtual Globe

With the advancement and the demand of internet based applications virtual globe has become the new medium to visualize and interact with global geospatial data. It not only allows users to interact and extract content from the globe in the real time but also reduces the effort manually accessing archives of satellite imagery. To develop cross-platform, cross-browser applications, several *WebGL* based virtual globes have been developed. Some of the globes are mentioned below.

1.1.10.1 WebGL Earth

WebGL Earth is an open source software developed for visualization of maps, satellite imagery and aerial photography on top of a virtual terrain. It is based on *WebGL* standard specifications, which allows to build customized without use of plugins.

Features

- It allows camera-dependent functionalities such as rotation, zoom and tilt.
- It supports existing maps from multiple sources such as *OpenStreetMap* and *Bing*.
- It allows supports custom map tiles for the earth or other planets

1.1.10.2 OpenWebGlobe

OpenWebGlobe is a *WebGL* based 3D geobrowser, which allows to process and visualize very large volumes of geospatial data. It consists of a complete SDK to develop web based applications without the need of plugin. It supports various forms of image data, elevation data, point of interest and 3D models. It is mainly useful to process very large amount of data, in highly parallel and scalable computing environments.

1.1.10.3 Cesium

Cesium is an open-source JavaScript library to create 3D virtual globes as well as 2D maps on a web browser. It utilizes WebGL to provide hardware acceleration and plugin independence and provides cross-browser functionality.

Features

- It is most suitable for dynamic geospatial data visualization with the help of Cesium Language(CZML). CZML is a JSON based schema, which describes geospatial data along with properties.
- It includes extensive libraries which support 2D as well as 3D geometries.
- It supports data imports from KML, ESRI Shapefiles and JSON.
- It supports extensive materials to describe the surface appearance of the objects. It also supports custom materials for the objects.
- Support for various 3D GIS capabilities

In This study Cesium is used to create 3D virtual globe to display 3D geo-data and various other 3D results.

1.2 Problem Statement& Motivation

Attribution of real world phenomena in 3D GIS are useful in understanding and decision making environment. 3Dmodels can be developed for great use and can be used in urban planning, infrastructure management, disaster management, environment monitoring(F. Remondino et al. 2009)(Breunig and Zlatanova 2011b). Having 3D data stored in a GeoRDBMS, the user has the possibility to extract only a limited set of data and thus critically reduce the time for loading. locating, editing and examining a particular object also becomes quick, simple and convenient(Stoter and Zlatanova 2003a).

3D data is visualized and exchanged by different file formats like KML, CityGML and COLLADA, these file systems cannot handle large and complex amounts of data. GeoRDBMS could provide the framework to define the geometry and topology of complex natural and anthropogenic objects. There can be further simplification of those complex structures. Geometric primitives like point, line, line-segment, TEN, polygon, polyhedrons can be very helpful in modeling complex features(Chen, Abdul-Rahmana, and Zlatanova 2008b)(Breunig and Zlatanova 2011b) .Virtual reality environments like Google earth incorporated with GeoRDBMS can lead to very exciting applications and can fill the gap of creation and visualization of large 3D data. For effective manipulation of that geo-data to get exciting results, visualization tools based on incorporation with GeoRDBMS have better success than 3D models like VRML, X3D as they don't contain semantic values of features(Breunig and Zlatanova 2011b).

To date augmented reality(AR) development and GeoRDBMS has been treated differently , AR with GeoRDBMS is a promising field keeping in mind complexity of models(Breunig and Zlatanova 2011b). In the GeoRDBMS query interface, typical spatial database queries could be formulated, such as ''How many cubic meters of water will be in the building if the water height is one meter above ground level?"(Breunig and Zlatanova 2011b). Data types like TEN , polyhedrons have major disadvantages of creation of large amount of data, inability to model curved surfaces properly and lack of spatial operations and high level topology pose hindrance for many complex operations like finding amount of merging area or total area formed by combining different types of features etc(Breunig and Zlatanova 2011b)(Stoter and Zlatanova 2003b).

Hence, 3D models must be extended with new representations such as *freeform curves* and *surfaces*, which can also reduce the amount of data stored in GeoRDBMS and can lead to high performance combined with exciting operations(Breunig and Zlatanova 2011b).

Various Spatial indexing techniques are defined for GeoRDBMSs but still there are many aspects which can considered like partitioning of logical tables hybrid spatial indexing techniques for optimal performance of queries and variable page size support in POSTGIS(Schön et al. 2009). GeoRDBMS need to be tuned manually for optimal performance as there are many optimization parameters which need to be monitored("Postgis-2.0.pdf"). Various optimization techniques must be performed manually at regular intervals whenever a new index is created , recreated and after loading a bulk data or after a large number of UPDATEs, INSERTs or DELETEs are issued against a table("Postgis-2.0.pdf").

Performance study of 3D geo-data needed to be done in multiuser environment considering the various parameters like bandwidth, data security, concurrency etc and service standards to access 3D geo-data for multi-user . Currently there is no common platform for migrating various 3D models in GeoRDBMS, and there is a need to develop an optimal way to migrate 3D file formats to GeoRDBMS. Concepts of Spatial databases for 2D and 3D geo-data applications is supported by many Spatial Database Management Systems (SDBMS), while true three-dimensional (3D) support for spatial data is a recent addition in Spatial Information Systems (SISs). The examples includes, ESRI's ArcGIS Geodatabase with its support for two-and-a-half dimensions (2.5D) in its Digital Elevation Model (DEM) and Triangular Irregular Network (TIN) and the more recent development of a terrain feature class and support for 3D objects and buildings with its multipatch feature class (Schön, et. al, 2009). The SDBMSs like Oracle, PostgreSQL and MySql have extended their technologies to integrate true 3D features inside RDBMS.

The task of geo-database researchers is to find implementations for geo-models that provide efficient storage and retrieval of these models in spatial databases. Transformation is often necessary to convert a geo-model used for visualisation into a geo-model stored in the GeoRDBMS (Chen, et al., 2008). Besides GeoRDBMS, open data models like CityGML, KML, COLLADA , X3D are used to exchange 3D assets. But the use of XML and GML for modelling and sharing spatial data, however is not undisputed and inflates the amount of data to be handled. This is not surprising, because XML was originally designed for modelling semi-structured data and not for handling structured and complex geo-data(Breunig and Zlatanova 2011b)(Schön et al. 2009).

Next generation of GIS software and application highly depends on GeoRDBMS in both geometric and topological modelling and analysis. One of the desired components in such future software or system is the geometric and topological modelling that works with 3D spatial operations (Chen, et al., 2008). Creation of 3D data type as database objects in GeoRDBMS is very important to recognise 3D data model of complex real world features. In absence of 3D primitive to model 3D objects inside RDBMS the DBMS will not function properly which may lead to a problem like validation of 3D objects and the spatial functions work only on projection of these objects in database. In general, the 2D objects that bound a 3D object are stored in multiple record to maintain 1:1 relationship in database and to establish a clear connection between object in database and in object in reality(C. W. Q. C. A. Arens and J.E. Stoter 2003). The Oracle and PostgreSQl+PostGIS support 3D data type as polyhedron and TIN respectively.

The migration of file based 3D data models into GeoRDBMS require special tools and processes. GIS applications, which offer true 3D functionalities other than 3D visualizations, are still rare. Also GIS and DBMS vendors are not matured towards full support for 3D(Stoter and Zlatanova 2003b).

There is need to develop tools by which focus can be shifted form only visualization of the 3D to interaction with the 3D geo-data based on various attributes and characteristics. Imagining a situation where dynamically queries like "How will the structure of a building change if a side is lifted certain meters" can be performed and output can be viewed on the fly. In the current scenario there is a limitation on querying over actual 3D data and interacting with the 3D output like output of a floor based query returning a 3D objects as the output which can be further manipulated separately. Another very powerful mode of sharing 3D spatial information is *3DWebGIS*, The role of the Internet for the knowledge and sharing of multidimensional information is Fundamental (Pescarin et al. 2005).Internet, a client/server system, is a perfect means of GIS data accessing, analyzing and transmission. Using Internet for GIS makes it easy access to acquire GIS data from diverse data sources in the distributed environment. In the current scenario very less emphasis has been given in the field of*3DWebGIS*. Users require immediate data access, means for the interoperable integration of different 3D geoinformation in different levels of detail, tools for 3d analysis and further data processing (based on data storage using databases, general purpose 3d GIS with functionalities like visibility analyses etc.) as well as solutions for interactive visualization and presentation (Altmaier and Kolbe 2003).This is because still 3D Gis is still in a very early phase and the 3D data generated is very huge and complex to be shared on Web. There is need to combine GeoRDBMS with Web based technology as well so that revolutionary advancement in the field of 3DGIS can be achieved.

1.3 Research Identification

1.3.1 Research Objective

3D geo-data organization and management in GeoRDBMS using created 3D geo-data in various file formats and attribute tagging with 3D data to perform spatial and non spatial queries in single and multiuser environment.

1.3.2 Research Sub Objective

a) Study Best Spatial indexing techniques for 3D geo-data and optimization of spatial indexing by development of front end software application .

b) Monitoring and comparison of database performance for access of different 3D Geo-Data format in single and multiuser environment and service standard to access 3D geo-data for multiuser .

c) Front end software application development for 3D GIS operations.

d) Creation of new topological operations for 3D geo-data, creation of new spatial operations for 3D data types and attribute and location Based Spatial Query on 3D geo-data.

1.3.3 Research Questions

a) What are the challenges for organization and management of 3D Geo-Data inside RDBMS?
b) How can Location and attribute based query can be performed for 3D geo-Data for Topological relationship?
c) Is there any data loss and are all file formats having same data format in GeoRDBMS?
d) Do we need to create other data type for storing topology of 3D geo-data like 3D freeform curves?
e) How will the choice of Indexing technique affect the performance of 3D GeoRDBMS, an do we need to create a new indexing technique for optimal performance of GeoRDBMS?
f) Are open source software systems for GIS sufficient to handle 3D geo-Data inside GeoRDBMS?

1.3.4 Innovation aimed at

a) Creation of 3D geo-data in GeoRDBMS.
b) Creation of new Spatial operations and storing topology in 3D GeoRDBMS so that attribute and location based querying can be done.
c) Optimization of spatial indexing technique and other optimization parameters for GeoRDBMS.
d) Development of front end software application for selection of parameters for spatial index for optimization and for 3D GIS operations.
e) Showing 3D results on Web based viewer and earth explorer.

2. Study area and Materials/data used

Noida Sector 82 and its surrounding has been modelled for this project. It is situated in Noida short for the New Okhla Industrial Development Authority, is a city in Uttar Pradesh India under the management of the New Okhla Industrial Development Authority (also called NOIDA).It is situated around 20 km from New Delhi. Noida Sector 82 is divided in localities like KendriyaVihar, VivekVihar, SwarnimVihar, UdyogVihar etc. Study area is spread in around 4.2 km2 in area. Study area consists of in total 676 buildings blocks of 16 different types of corresponding structure, along with 359 in total line features like boundary walls and roads. Every building is of a certain type which have similar structure and color e.g. 124 buildings in kendriyavihar has same structure, material and color this factor makes it easier to model a large area to create a large dataset which is required in this study. The surrounding area is plain and has no structures that block the view of the buildings therefore it is relatively easy to take photographs of the buildings from various angles. The base length and standard height between the photographs can also be easily maintained as there are less occlusions situated in the near proximity of the area making it is easy to walk around it.

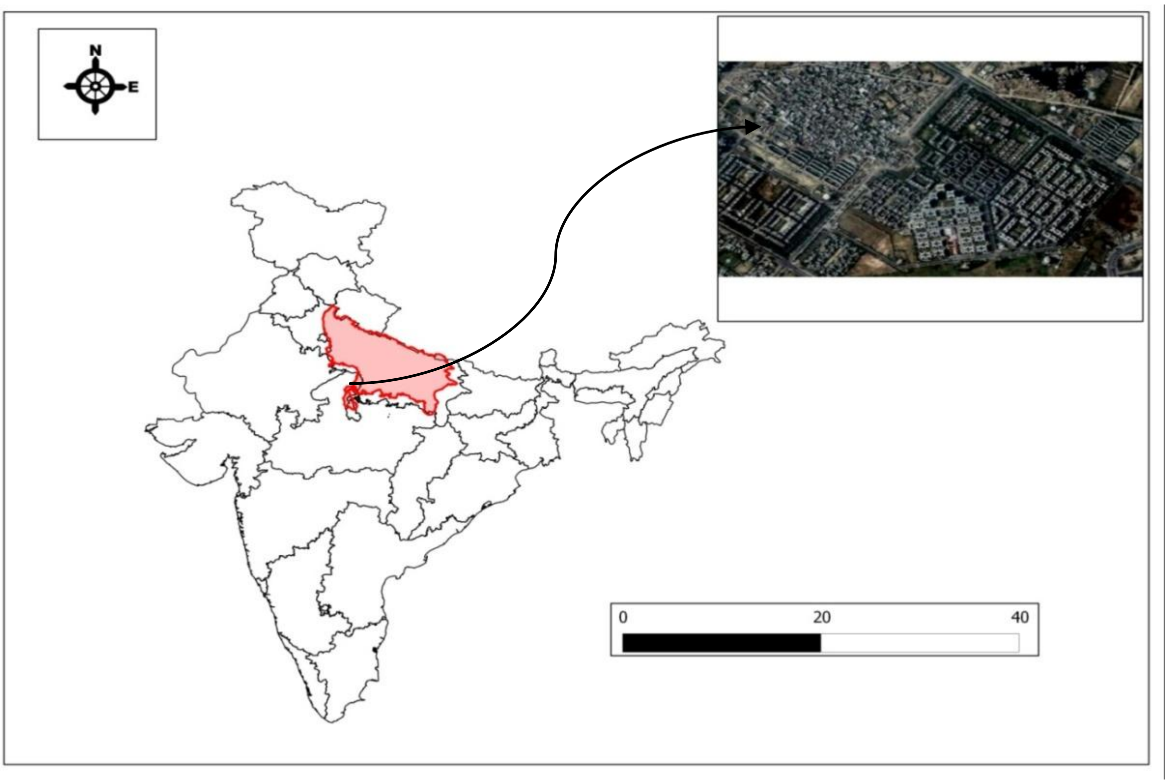


Figure 2.1 Study area Noida sector 82 and surrounding area.

2.1 Dataset and Software used

High resolution satellite data is accessed from various Geo-Web Services like Google , Bing and Bhuvan Geospatial. Photographs of the study area is used to carry out development of rule file and texturing using Nikon D-80 SLR camera. Latitude and longitude of the objects is collected using a GPS. Table1 below show the software and instruments used in this project.

2.1.1 Camera Settings

Camera Model: Nikon D80 Digital SLR Camera
Sensor: 10.2 million pixels and 23.6*15.8 mm CCD (DX Format)
Image Resolution: 3872*2597 (10.0MP)
Focal length: 18mm
Sensitivity: ISO 100-1600
Shutter Speed: 30 to 1/4000 sec
Image Format: JPEG and RAW

Table1- Software and instruments used in the project

S. No.	Item	Use
1	Postgresql+PostGIS, Oracle spatial11g	Database
2	ArcGis10.0	Extracting footprint and providing height values to each type of buildings
3	QGIS	Extracting footprint by performing digitization on data accessed from Geo-Web Servers
4	3D City DBimporter	Migration of CityGML file to database
5	OGR Library	Migration of KMl and Shape file to and from database
6	GoogleSketchup	For creating 3D model in CityGML format
7	Arcsceane	For creating 3D model at LOD1 by providing actual height information collected from field survey
8	CityEngine	For creating 3D model at LOD2 and LOD3 by writing rule files in CityEngine's procedural language
9	Leica GPS	GPS Points
10	FZKviewer	For visualizing CityGML data
11	FME	For making invalid schema to valid
12	Apache Tomcat	Server
13	Cesium	An open-source library to create 3D virtual globes
14	Camera	For taking photographs of study area

3. Literature Review

3.1 Introduction

3D modeling is a process in which information about an object is being derived from its actual existence and then further modeled to create a real world scenario. The 3D model can be used for various purposes like disaster management, urban planning etc. 3D modeling is becoming an essential tool in the planning of all above mentioned field. This research on '3D Geodata management and Query optimization in multiuser access', focuses on the utility of incorporation of 3D modeling with GeoRDBMS, various open source tools and web-based 3D services as a software package. The research constructs a 3D realistic model of study area utilizing 3D processing methods that employs both the free to use and open source and professional software approach. 3D model is migrated to GeoRDBMS and structure of data is being studied for various file formats like COLLADA, CityGML, KML. Manual tuning of GeoRDBMS is performed and performance assessment of various spatial queries is studied in single and multiuser environment. New software modules are written and are studied, along with spatial and non spatial queries attribute based queries are also performed on 3D Geodata. Front end spatial query tool is being developed to perform spatial and non spatial queries and resulting 3D output is being shown on developed web based platform.

3.2 GeoRDBMS and 3D GIS

(Chen, Abdul-Rahmana, and Zlatanova 2008b) defines Database management system, or DBMS, as a computer software program that is designed as a means of managing all databases that are currently installed on a system hard drive or network. The paper written by (Breunig and Zlatanova 2011b) explains GeoRDBMS as extension of traditional business database system by incorporating spatial data types in its data model and query language. Furthermore, it supports spatial data types in its implementation, including spatial indexing. This paper also discusses 3D geo-database research as a promising field to support challenging applications such as 3D urban planning, environmental monitoring, infrastructure management, and early warning or disaster management and response. GeoRDBMS is very important and evolving aspect for 3D GIS, as large volume of spatial data can be managed inside GeoRDBMS environment and also it can store semantics similar to CityGML for 3D geo-data this concept is similar to the what I have used in my study. To date, however, GeoRDBMS and AR(Augmented Reality) development have been treated separately.

(Breunig and Zlatanova 2011b) paper on 3D geo-database research retrospective and future directions discusses the inclusion of augmented reality (AR) methods, which overlay 3D models as virtual information with the physical environment in real time, is one of the possible new user interfaces for geo-database use also GeoRDBMS could provide the framework to define the geometry and topology of nature-formed and man-made objects in a unified way .This important point made in the paper is part of my present study. In a given GeoRDBMS, data may be modelled in tables (relational database approach) or as objects (object-relational or object-oriented database approach) as parts of a geo-data model. The separation of disciplines that model various real-world phenomena (topography, geology, atmosphere/climate, and ocean) has led to the definition of a variety of geo-objects, which are usually based on different representations, such as boundary representations and volumetric representations. Traditionally, topographical objects (on the surface of the Earth) are the oldest phenomena modelled and therefore exhibit the greatest variation. Applications relying on geo-models have distinct needs, some applications may require models only for visualisation, while others may require models for analysis and statistics.

(Chen, Abdul-Rahmana, and Zlatanova 2008b) paper on 3D spatial operations for GeoRDBMS geometry vs. topology discusses the role of GeoRDBMS in managing and handling, of large volume of spatial data. By developing 3D spatial database with appropriate operation tools such as 3D spatial operations would be very useful for of 3D GIS since the it would highly depend on the GeoRDBMS in both modeling and analysis. One of the desired components in such future software or system is geometric and topological modeling capability that works with 3D spatial operations. A 3D spatial system must support 3D data types, such as point, line, surface and volume in 3D Euclidean space. Such data types are based on a 3D geometric data model (i.e. vector and/or raster data with underlying geometry and topology).

In the paper Storage, manipulation, and visualization of LiDAR data by (Schön et al. 2009) suggests a very important point that 3D spatial system must also offer operations and functions embedded into its query language that can operate with its 3D data types. Spatial Database Management Systems (SDBMSs) vendors have extended their technologies to integrate true 3D features. This is largely driven by the increased availability of 3D data (e.g. from aerial and terrestrial LiDAR). In the papers (Stoter and Zlatanova 2003a)(Muki Haklay 2006) provide details about the different types of 3Ddata types(TIN, Polyhedron, MultiPolygon) that are used to model any natural or manmade features are also used in present study to create 3D geo-data.

3.3 3D geo-data handling for GIS applications

Concepts of Spatial databases for 2D and 3D geo-data applications is supported by many Spatial Database Management Systems (SDBMS), while true three-dimensional (3D) support for spatial data is a recent addition in Spatial Information Systems (SISs). The examples includes, ESRI's *ArcGIS* Geodatabase with its support for two-and-a-half dimensions (2.5D) in its Digital Elevation Model (DEM) and Triangular Irregular Network (TIN) and the more recent development of a Terrain feature class and support for 3D objects and buildings with its multipatch feature class(Schön et al. 2009)(Sardadi et al. 2008a).

The SDBMSs like *Oracle*, *PostgreSQL* and *MySql* have extended their technologies to integrate true 3D features inside RDBMS. The task of geo-database researchers is to find implementations for geo-models that provide efficient storage and retrieval of these models in spatial databases. Paper by (Chen, Abdul-Rahmana, and Zlatanova 2008b)(Tavanti and Lind 2001) explains transformation is often necessary to convert a geo-model used for visualization into a geo-model stored in the GeoRDBMS. Besides GeoRDBMS, open data models like CityGML, KML, COLLADA , X3D are used to exchange 3D assets. But the use of XML and GML for modelling and sharing spatial data, however is not undisputed and inflates the amount of data to be handled. This is not surprising, because XML was originally designed for modelling semi-structured data and not for handling structured and complex geo-data. Next generation of GIS software and application highly depends on DBMS in both geometric and topological modelling and analysis. One of the desired components in such future software or system is the geometric and topological modelling that works with 3D spatial operations. Creation of 3D data type as database objects in GeoRDBMS is very important to recognize 3D data model of complex real world features.

As per paper by (C. W. Q. C. A. Arens and J.E. Stoter 2003)in absence of 3D primitive to model 3D objects inside RDBMS the DBMS will not function properly which may lead to a problem like validation of 3D objects and the spatial functions work only on projection of these objects in database In the present study similar work is done to store data as 3D primitives. In general, the

2D objects that bound a 3D object are stored in multiple record to maintain 1:1 relationship in database and to establish a clear connection between object in database and in object in reality (Quak and Stoter, 2003). The Oracle and PostgreSQl+PostGIS support 3D data type as polyhedron and TIN respectively. The migration of file based 3D data models into GeoRDBMS require special tools and processes. In present study similar points and issues are being studied.

(Rizwan Khan and Smita Sengupta 2011) describes various file formats as

Keyhole Markup Language (KML):

Keyhole Markup Language (KML) is an XML-based language provided by Google for defining the graphic display of spatial data in applications such as Google Earth and GoogleMaps. KML enables these applications to support the open integration of custom data layers from many GIS users. KML files have either a *.kml* file extension or a .kmz file extension

(for compressed and zipped KML files). KML can be used to:

- Symbolize and display GIS data as elements within Google Earth and Google Maps using symbols, color, images and information tags.
- Stores attribute information about geographic features.
- Define the user's interaction with those features -- for example, to control fly-to and camera location settings in Google Earth.

Geography Markup Language (GML):

CityGML: CityGML is a common information model for the representation of 3D urban objects. It defines the classes and relations for the most relevant topographic objects and regional models with respect to their geometrical, topological and semantical properties. The thematic information of CityGML goes beyond graphic exchange formats and allowsdeveloping virtual 3D city models for sophisticated analysis. This can be used in different application domains like simulations, urban data mining, facility management, and thematic inquiries.

CityGML is realized as an open data model and XML-based format for the storage and exchange of virtual 3D city models. It is implemented as an application schema for the Geography Markup Language 3 (GML3), the extendible international standard for spatial data exchange issued by the Open Geospatial Consortium (OGC) and the ISO TC211.

CityGML is intended to become an open standard and therefore can be used free of charge.

CityGML defines five levels of detail numerated from zero to four:- (Figure1**)**

LOD 0:- 2.5 D digital terrain model

LOD 1:- Block model comprising buildings structure (Height Data)

LOD 2:- Differentiated roof structures and textures.

LOD3:-Architectural models with detailed wall and roof structures, detailed vegetation and

transportation objects.

LOD 4:- LOD3 model + adding interior structures for 3D objects. For example, buildings are

composed of rooms, interior doors, stairs, and furniture.

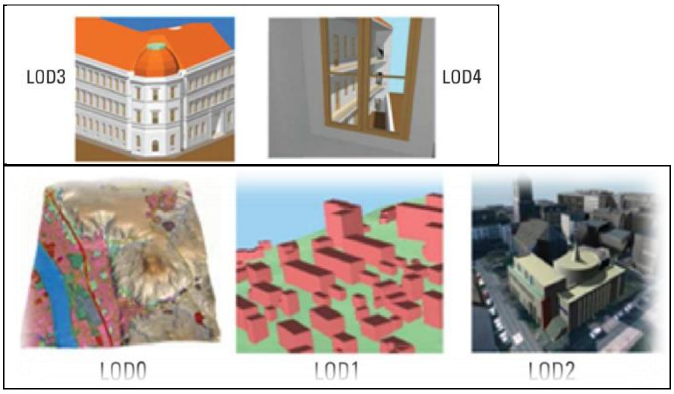


Figure3.1The five levels of detail (LOD) defined by CityGML

In the present study 3D model is generated in these file formats in LOD1, LOD2, LOD3, LOD4.

3.4 Close Range Photogrammetry and Point cloud Generation

Close range photogrammetry has proved to be quite useful in several fields.(Yang et al. 2013) describes its use in 'Image based 3D reconstruction' well in their paper. They detail about various techniques to reconstruct a 3D model by using photogrammetry techniques one of them is Structure from Motion (SFM), In which once the camera orientation parameters are available, the 3D coordinates of the cameras and the image-based point cloud can be generated for the scene. They also talk about SIFT algorithm which extracts feature points for fundamental attributes in experimental images, and records the corresponding images in the database for tracking process and camera continuous motion calculation. Because of highly significant and relatively easy to capture feature points, the SIFT algorithm is commonly used in computer vision. Based on the appearance of local points of interest, the SIFT algorithm can effectively resist the size and rotation difference of images. The tolerance of SIFT in light, noise, view and zoom-in/out is quite high, and can accurately identify object features in the large number of non-parametric database.

(Kaasalainen et al. 2009) describe their study in their paper 'Radiometric Calibration of Terrestrial Laser Scanners with External Reference Targets'. In their study they have presented a case study of radiometric calibration for two phase-shift continuous wave (CW) terrestrial scanners and discuss some major issues in correcting and applying the intensity data, and a practical calibration scheme based on external reference targets.

In the paper by they have presented a framework for 3D geometric shape segmentation for close-range scenes used in mobile manipulation and grasping, out of sensed point cloud data. In their study objects are segmented out from partial views and a reconstructed model is computed by fitting geometric primitive classes such as planes, spheres, cylinders, and cones. The geometric shape coefficients are then used to reconstruct missing data. Residual points are resampled and triangulated, to create smooth decoupled surfaces that can be manipulated. The resulted map is represented as a hybrid concept and is comprised of 3D shape coefficients and triangular meshes used for collision avoidance in manipulation routines.

3.5 3D modeling and 3D geo-objects

(De Cambray 1993) in the paper ‘three-dimensional (3d) modelling in a geographical database’ explains that spatial data is 3D data and most current geographical database systems are extended to 3D information. because, the third dimension appears to be more and more necessary for many application domains as geology, civil and military engineering, town planning, robotics, etc. These applications require not only the modelling and the visualization of 3D data but also the manipulation of these data. 3D data involved in a GIS may be subsoil data or relief data or 3D geographical data that can either be human made entities like bridges or naturally occurring entities like forests. A 3D entity is a volumetric object which can be convex or concave and which may contain holes. A cube and a sphere are examples of 3D entities.

Paper by (Fabio Remondino and El-Hakim 2006) defines “three-dimensional (3D) modelling of an object can be seen as the complete process that starts from data acquisition and ends with a 3D virtual model visually interactive on a computer”. As described by (Altmaier and Kolbe 2003)apart from developing a 3D model of desired a study area there are other very important aspects related to 3D modeling like Multiple use and sustainability, Interoperability, 3D visualization, Application fields for 3d geo-visualization, Interoperable 3d geovisualization over the www, Geo-visualization using web services, performance of spatial query while working on 3D geo-data inside GeoRDBMSetc these important points are being addressed in present study.

Paper by (Van Oosterom et al. 2002)explains the details of balance between topology and geometry of a 3D object and every 3D object must follow the defined topology rules like 4 intersection and 9 intersection model. 3D modeling as augmented reality creates the virtual environment of real world scenario. AS per paper by (Breunig and Zlatanova 2011b) the heterogeneity of current geo-spatial geometric and topo-logical data models shows the importance of standards for GeoRDBMS in 3Dmodelling.There is need to think beyond only 3D visualization and incorporate GeoRDBMS and AR development in 3D modeling process, this important point is part of my present study. As per paper by (Stoter and Zlatanova 2003a) 3D object used by most GeoRDBMS (Oracle, Postgres, IBM, Ingres, Informix) support the storage of 3D points, lines and polygons, multipolygon, 3D objects can also be represented as polyhedrons in 3Dmodelling.

Another aspect in 3Dmodelling is difference between 2.5D and 3D modelling As discussed in paper In Transition from 2.5D GIS to 3D by (Dieter Schmidt 1996) a single Z value is surfaces having similar z values ex DEM and TIN in 2.5D. To fill the gap between 2.5D and 3D an additional three dimensional data model is defined where every 3D geo-object 3D as well as 2D attributes. Paper by (De Cambray 1993) describes With usual 2.5D maps, each geographical feature (or part of) such as a forest, a field or a road becomes an individual entity represented by its 2D-boundary augmented with a z coordinate.

While linear entities are well approximated, inner surface elevation variations are not captured. A nonhomogeneous approximation of the underlying terrain is provided whose accuracy depends on the local entity density.

In the paper ‘ A 3-D city model data structure and its implementation in a relational database’ (Gruen and Wang 1998) explains that a 3-D objects usually can be classified into two types: surface objects (such as roads, waterways etc.) and body objects (such as building). Both types of objects can be completely represented with a facet model. On the other hand, a facet model is easily reconstructed with data from a photogrammetric system. CC-Modeler, developed in ETH Zürich, is a special photogrammetric tool to generate the facet model of 3-D objects from point cloud data obtained with a photogrammetric system.

In 3D modeling Interoperability of the developed model is very important point, There are various file formats which are used to carry designed 3D models and make 3D model interoperable, few of them are X3D, VRML, KML, CityGML, COLLADA. Not every file format can contain semantic details of the model except CityGML in the paper by (Kolbe 2009) It Introduces CityGML as an interoperable file format which not only represents the shape and graphical appearance of city models but specifically addresses the object semantics and the representation of the thematic properties, taxonomies and aggregations. It also explains that CityGML represents four different aspects of virtual 3D city models, i.e. semantics, geometry, topology, and appearance. Above, all objects can be represented in up to five different, well-defined levels-of-details.

In the present study CityGML is one of the file formats in which 3D model of study area is developed and further studied. These file formats are very large and are very complex. GeoRDBMS is the other options which can solve the problems being faced by these file formats where GeoRDBMS can effectively store semantic details of a 3D model as well. Topology of 3D geo-objects is another very important point which is being discussed in the paper (Muki Haklay 2006). Topological relationships can be broadly classified into adjacency, co-linearity and containment , it details 9-intersection relationships between two regular 3D object namely COVERS, COVERED_BY, CONTAINS, INSIDE, DISJOINT, MEET, OVERLAP and EQUAL. It also discusses issues related to these topological relationships between 3D objects and possible solutions for the issues and how it cuts down on computational processing required for Geographical Information Systems and its difference from 4-intersection relationship model. In the present study these topological operations are being performed on the study area.

3.6 Spatial Indexing and performance of GeoRDBMS

An efficient implementation of search trees is crucial for any spatial database system.(Schön et al. 2009) in his paper describes spatial Indexing in a database is used to accelerate operations performed on the dataset. A spatial index organizes the space and the objects within this space in a particular manner, so that a spatial query or a spatial operator does not have to traverse the complete table to retrieve specific data. There are various spatial indexing techniques, many SDBMS vendors typically offer spatial indexes like R-trees, GiST, Quadtree. In this paper detail of different hybrid spatial indexes which can be applied in future in various databases are also explained like V-reactive tree, VS tree, K-d tree etc but vendors that support true 3D data has so far been limited. In my present study spatial indexing is being studied and many new ideas is also deduced. In their study they also have done performance assessment of GeoRDBMS on point cloud data and how GeoRDBMS handles it. Crucial problem studied by them was how quickly a representative 3D set of point cloud data can be loaded into the database. As part of the on-going research in this area, the authors loaded a set of approximately 18 million points (approximately 700 MB) from a LiDAR point cloud. This required 20min13sec to load.In the paper by (Breunig and Zlatanova 2011b) detail of working of RTree, GiST is explained. Also it explains GiST unifies structures such as B-trees and R-trees in a single piece of code. GiST may behave like B-tree, R-tree, or any other customized spatial access method. Paper also discusses 3D spatial indexes and implementation of 3D spatial indexes over 2D spatial indexes when applied on 3D geo-data this important point is also studied in present work.

Spatial indexing technique is one of the most important factors in better performance of GeoRDBMS as 3D geo-data is a very large and complex data thus there is need of manual tuning and optimization of GeoRDBMS manually. In general, without indexing any search for a spatial feature requires a sequential scan of every record in the database ("Postgis-2.0.pdf"). The indexing techniques speeds up the searching by organizing data into a search tree which can be quickly traversed to find a particular record (Sardadi et al. 2008a).Optimization of spatial query also depends on choice of spatial indexing technique and optimization of spatial query on large 3D geo-data is one of the important objectives of the present study. SDBMS vendor like PostgreSQL supports three kinds of index i.e. B-Tree indexes, R-Tree indexes, and n dimensional GiST indexes. These spatial indexing techniques are applied on 3D data models created in the present study. In this paper working of 2D R-Tree and 3D R-Tree is being explained. Every object in 2D R-tree is being represented by a MBR(Minimum Bounding Rectangle) and in 3D every object is being represented by MBB(Minimum Bounding Box), this property of forming a MBB in 3D R-Tree becomes very difficult to apply on point cloud data this discussion is an important part of the present study.

Paper by (Raptopoulou, Vassilakopoulos, and Manolopoulos 2004) explains the concept of querying and indexing moving objects. For indexing such scenarios paper suggests XBR spatial indexing technique. As the XBR trees constitute a family of new secondary memory structures, which are suitable for storing and indexing multi-dimensional points and line segments. In 2 dimensions, the resulting structure is an External Balanced Quadtree, in 3 dimensions an External Balances Octtree, and in higher dimensions an external balanced hyper Quadtree. Paper by (Hellerstein, Naughton, and Pfeffer 1995) details about the GiST spatial indexing technique developed by them. Generalized Search Tree (GiST),which is easily extensible both in the data types it can index and in the queries it can support. Extensibility of queries is particularly important, since it allows new data types to be indexed in a manner that supports the queries natural to the types. In addition to providing extensibility for new data types, the GiST unifies previously disparate structures used for currently common data types. For example, both B+-trees and R-trees can be implemented as extensions of the GiST, resulting in a single code base for indexing multiple dissimilar applications. GiST spatial indexing technique is also an important part of my study.

4. Methodology

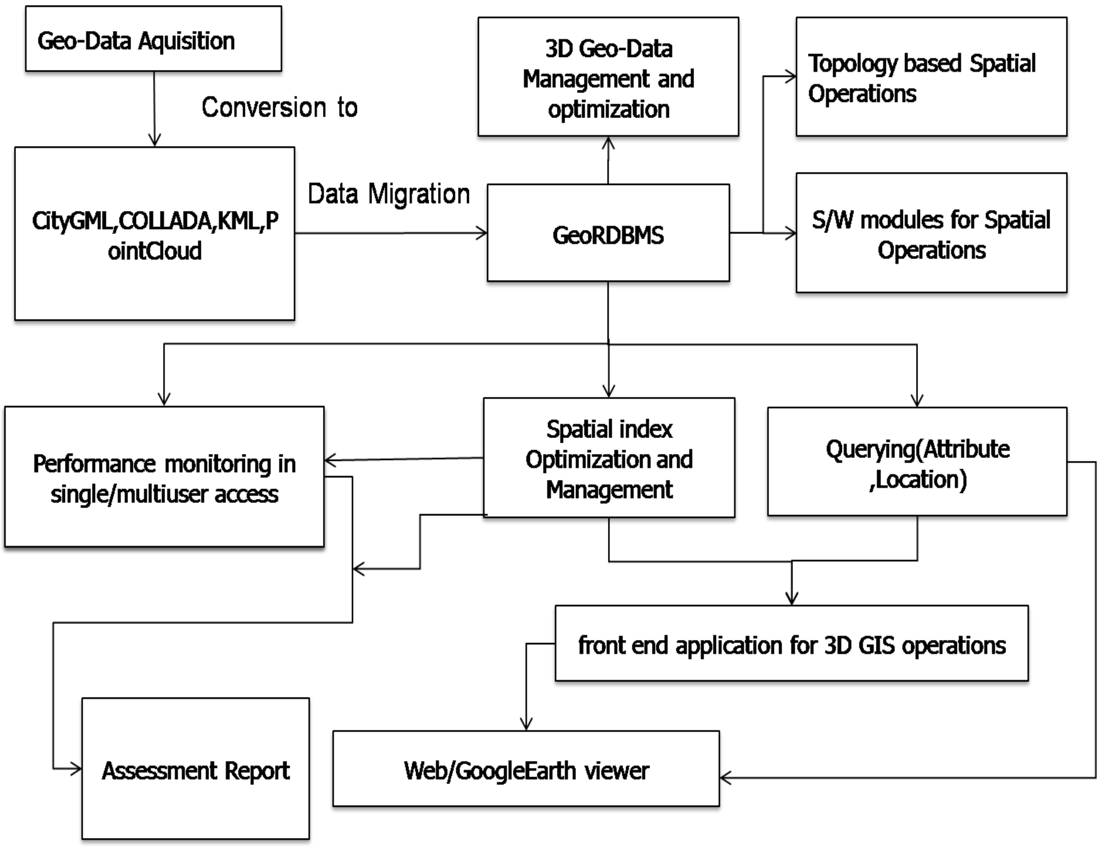


Figure 4.1 : Methodology Workflow Diagram

Methodology of this study mainly focuses on combining GeoRDBMS, augmented reality and studying the performance of GeoRDBMS by creating 3D geo-data in various file formats like CityGML, KML, COLLADA. The methodology also involves creation of new software modules for spatial operation on 3D geo-data by developed front end application. Figure 4.1 represents the overall flowchart of the methodology.

The methodology is mainly divided into following sections:

1. Acquiring the required datasets for the study area.

2. Conversion to File formats like KML, CityGML, COLLADA

3. Migration of file formats to GeoRDBMS

4. 3D geo-data organization, management and optimization

5. Development of software modules for spatial operations and topological operations

6. Performance study of GeoRDBMS and spatial indexing techniques for various file formats and generation of assessment report.

7. Attribute and location based querying on 3D geo-data using front end application.

8. Viewing the results on web browser and earth explorer.

4.1 Data Acquisition

Initial step of this study involves data planning and acquisition. The data that needs to be obtained consists of taking terrestrial images of the study area, consisting of in total 676 buildings blocks of 16 different types of corresponding structure, along with 359 in total line features like boundary walls and roads using a Nikon D-80 SLR camera. GPS points for these images and locations are recorded using a GPS device. This is done in order to geo-reference the images for better model accuracy.

The 3D geo-data model of the study area is created by using *QGIS*, *ESRI CityEngine* and *GoogleSketchup*. The QGIS is used to extract the footprint of 3D object by digitizing georeferenced images and is converted it to vector format. The generated shapefile is imported along with photographs of the study area for texturing into *CityEngine*. Rule files to develop the exact structure of the study area is written in *CityEngine*. Sample structure of the rule file is shown in Figure 4.2.3.

4.1.1 Satellite Data Acquired

High resolution satellite data is accessed from various Geo-Web Services like *Google* , *Bing* and *Bhuvan Geospatial*.

4.1.2 Software's Used

The software packages and programming languages used to complete this study are listed in the Table1 in section 2.1.

4.2 Creation and Conversion of 3D Models to File formats like KML, CityGML, COLLADA

CityEngine is used to create 3Dmodel of the study area methodology followed for creation of 3D model is in Figure 4.2.1.CityEngine is used to export 3Dmodel in KMl and COLLADA format. CityGML plugin in *GoogleSketchup* is used to prepare the 3D model in CityGML format. *GoogleSketchup* is used to create various *.obj* files for objects which is used in *CityEngine* for preparation of actual 3D model. In addition to this the 3D geo-data is also generated by terrestrial photogrammetry techniques in which overlapping photographs of the study area are taken.

Photomodeler software is used to generate the tie points. The actual height of few tie points is used to generate the 3D data model in various 3D file formats.

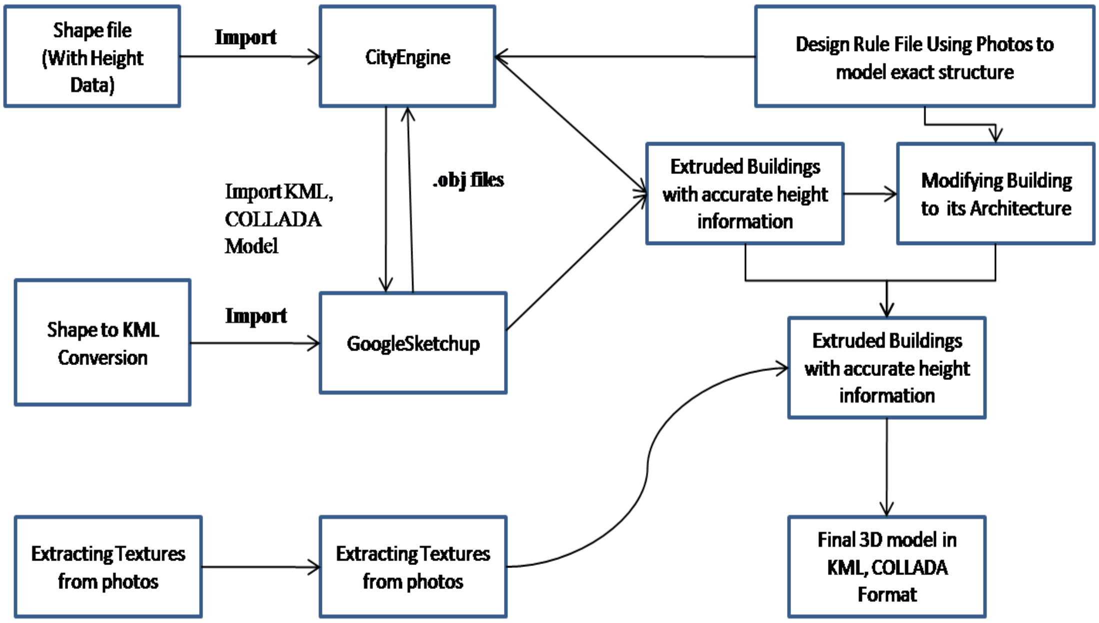


Figure 4.2.1 Methodology to create 3D Model of the study area

Extracted footprint of the study area is imported in *CityEngine*. Terrestrial photographs of the study area are used for texturing the developed 3D buildings. Terrestrial photographs are also used to know the exact structure of the buildings and based on the structures information extracted from the photograph, rule files for each type of building is written in CityGML procedural language basic structure of a rule file for a building is shown in Figure 4.1.2.3 . "*.obj*" files are used to design various entities like doors, windows etc to create exact real world scenario of the study area. Final 3D model is then exported in COLLADA, KML file formats.

GoogleSketchup is used to create CityGML file format of the 3D model. Process of creation of 3Dmodel in CityGML file format is shown in Figure 4.2.2

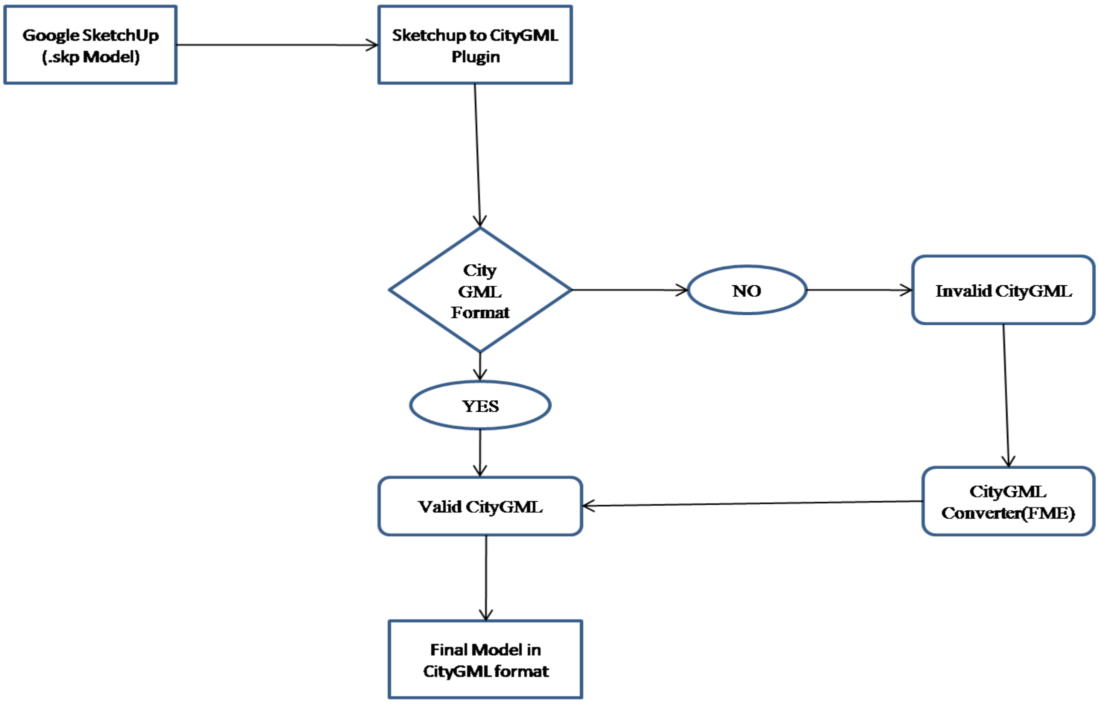


Figure 4.2.2 Process of creation of a 3D model into Valid CityGML format

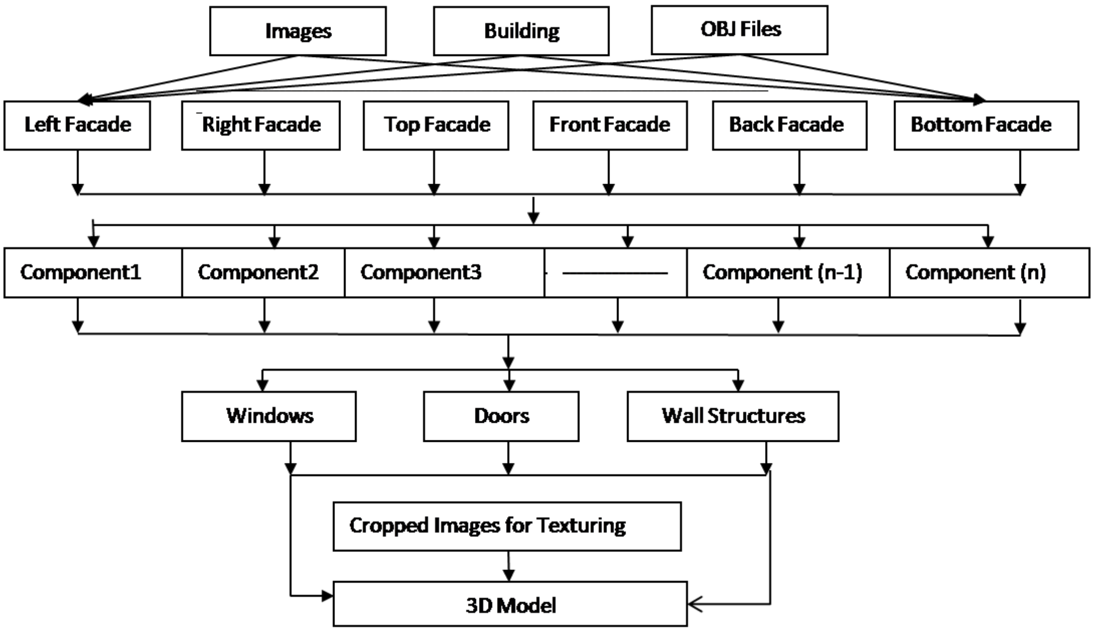


Figure 4.2.3 Basic structure of a Rule file of a building

3D model developed in *CityEngine* in KML and COLLADA format is imported in to *GoogleSketchup* and is saved as "*.skp*" model which is *GoogleSketchup* format model. This imported 3D model is exported as CityGML file format using *GoogleSketchup* to CityGML Plugin. This exported CityGML file format can be valid or invalid CityGML file, if the exported file is invalid CityGML than CityGML converter software like FME is used to redesign the schema of CityGML and make it valid CityGML file. Final 3Dmodel in KML, COLLADA, CityGML formats is migrated to GeoRDBMS for further study.

4.3 Migration of file formats to GeoRDBMS

The created 3D geo-data is migrated to PostGIS using PostGIS Shapefile and DBF importer exporter utility. Same operation is also performed by OGR command line utility. The original 3D geo-data is available in COLLADA, KML, and CityGML format. After migration 3Dgeo-data is further studied for its structure, format, size and complexity. CityGML file format is migrated using 3DCityDB Importer Exporter. CityGML is migrated to 3D City database which has a specific database structure to which CityGML data is migrated (Kolbe, Nagel, and Herreruela 2013). CityGML is migrated to GeoRDBMS structured in 46 different schemas like 'cityobject', furniture etc. Data in the CityGML file is matched by the tags which represents the type of object to the schema in the GeoRDBMS like <cityObjectMember> tag is matched to 'cityobject' schema in the GeoRDBMS. 3D geo-data from GeoRDBMS is exported as KML format using OGR utility library and in CityGML format using DBF importer exporter. 3D geo-data in COLLADA format is migrated to GeoRDBMS using DBF importer exporter to schema in GeoRDBMS. 3D geo-data in KML format is migrated to GeoRDBMS using OGR utility library to schema in GeoRDBMS. OGR utility library is used to export 3D geo-data from GeoRDBMS.

4.4 3D geo-data organization, management and optimization

3D geo-data created and is large and complex and file formats were designed for small semi structured data. The need for collaboration of GeoRDBMS with various file formats and augmented reality is discussed by various authors (Breunig and Zlatanova 2011b)(Schön et al. 2009). The same is also discussed in this study. These large and complex datasets are migrated to GeoRDBMS which provides the platform where 3D geo-data is organized, managed and optimized. Having 3D data stored in a GeoRDBMS, the user has the possibility to extract only a limited set of data and thus critically reduce the time for loading. locating, editing and examining a particular object also becomes quick, simple and convenient(Stoter and Zlatanova 2003c).

GeoRDBMS provides the framework to define the geometry and topology of 3D geo-data (Breunig and Zlatanova 2011b). According to the OpenGIS specifications, the geo-data is represented by two structures, i.e. geometrical(i.e. simple feature specifications) and topological (i.e. complex feature specifications).While the geometric structure provides direct access to the coordinates of individual objects, the topological structure encapsulates information about their spatial relationships (Stoter and Zlatanova 2003c). The geo-data features are stored in a geometrical model, without internal topology. Topological relationships between geometries can be retrieved by the use of spatial operators. Many SDBMS vendors provide support for actual 3D geo-data due to implementation of 3D primitives as data types. Various spatial data types supported by vendors like PostGIS and oracle are POINT, LINE, POLYGON, MULTIPOLYGON, POLYHEDRON, TIN. 3D geo-data migrated to GeoRDBMS is stored as 3D primitive inside the GeoRDBMS due to the support for 3D spatial data types like MULTIPOLYGON, POINT, LINE, POLYGON, TIN and POLYHEDRON (Khuan, Abdul-Rahman, and Zlatanova 2007). As per paper by (C. W. Q. C. A. Arens and J.E. Stoter 2003)in absence of 3D primitive to model 3D objects inside RDBMS the DBMS will not function properly which may lead to a problem like validation of 3D objects and the spatial functions work only on projection of these objects in database In the present study similar work is done to store data as 3D primitives.

In general, the 2D objects that bound a 3D object are stored in multiple record to maintain 1:1 relationship in database and to establish a clear connection between object in database and in object in reality (C. W. Q. C. A. Arens and J.E. Stoter 2003). An advantage of 3D multipolygons (compare to list of polygons) is that it is recognised as one object by front-end applications (GIS/CAD) that can access, visualise, and edit these data and post the changes back to the database. Another advantage of the 3D multipolygon approach is the one-to-one correspondence between a record and an object(Stoter and Zlatanova 2003c). An example of implementation of polyhedron inside GeoRDBMS is given below.

POLYHEDRON(PolygonInfo(6,24),SumVertexList(8),SumPolygon

List(4,4,4,4,4,4),VertexList(100.0,100.0,100.0,400.0,100.0,

100.0,400.0,400.0,100.0,100.0,400.0,100.0,100.0,100.0,400.0

,400.0,100.0,400.0,400.0,400.0,400.0,100.0,400.0,400.0),Pol

PolygonList(1,2,6,5,2,3,7,6,3,4,8,7,4,1,5,8,5,6,7,8,1,4,3,2))

1) PolygonInfo(6,24) denotes 6 polygons and 24 IDs in PolygonList,
2) SumVertexList(8) denotes the total vertices,
3) SumPolygonList(4,4,4,4,4,4) denotes total vertices for each of polygon (total
4) polygon is 6, referred to (1)),
5) VertexList() denotes the list of coordinate-values for all vertices (with no
6) redundant), and
7) PolygonList() denotes the information about each polygon from sets of ID.

Various topological operations which are applied on 3D primitive is shown in figure 4.4 below

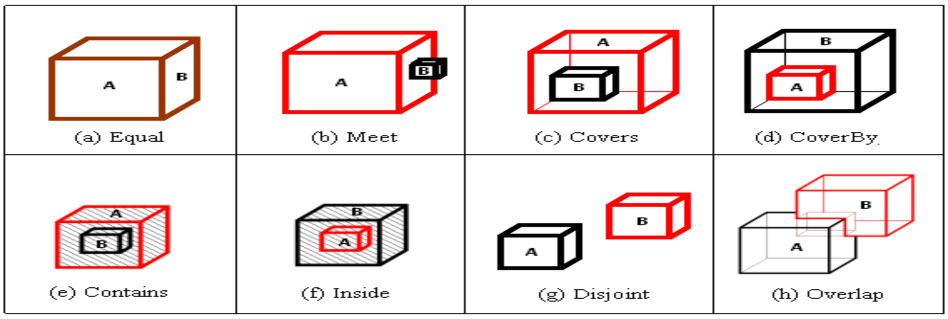


Figure 4.4 Topological operations for 3D primitive

Migrated 3D geo-data is stored as 3DPolygon, 3DMultipolygon in GeoRDBMS. Each polygon is stored as one row inside GeoRDBMS and each Multipolygon is stored in one row where its implementation has an issue of repetition of shared coordinates. This leads to inconsistency in the GeoRDBMS. Solution to this issue is by storing the 3D geo-data using polyhedrons, but use of TEN, polyhedron has major disadvantages of creation of large amount of data, inability to model curved surfaces properly and lack of spatial operations and high level topology pose hindrance for many complex operations like finding amount of merging area or total area formed by combining different types of features etc(Stoter and Zlatanova 2003c)(Breunig and Zlatanova 2011b).

Created 3D geo-data is very large and complex and needs special attention while migration and storage in GeoRDBMS. Although SDBMS vendors have functionality to optimize GeoRDBMS, but GeoRDBMS need to be tuned manually for optimal performance as there are many optimization parameters which needs to be monitored("Postgis-2.0.pdf")(Schön et al. 2009). In this study manually various configuration parameters are addressed and implemented to manually tune the GeoRDBMS for better performance of query execution time. which provide a crude method of influencing the query plans chosen by the query optimizer ex vaccum, auto vaccum, enable bitmapscan, setting catch size and working memory size etc.

Various Spatial indexing methods is studied and applied according to the structure of 3D geo-data so that GeoRDBMS performance is enhanced. Choice of spatial indexing is very important aspect of fast query processing as it depends on the type of the query and the 3D geo-data. Detail of performance of various spatial operations and their performance w.r.t to choice of spatial indexing techniques is studied and option is provided to manually select the best spatial indexing technique for different type of spatial operations for better query execution time.

4.5 Development of software modules for spatial operations and topological operations

Next generation of GIS software would highly depend on DBMS in both geometrical modeling and analysis. One of the desired components in such future software or system is the geometric modeling that works with 3Dspatial operations(Khuan, Abdul-Rahman, and Zlatanova 2007). True three-dimensional (3D) support for spatial data is a recent addition in Spatial Information Systems (SISs) (Schön et al. 2009) thus there is lack of spatial and topological operations to be applied on 3D geo-data. In the present study new spatial and topological operations are developed and applied on created 3D geo-data so that these spatial operations helps in building various exciting augmented reality applications. **New spatial and topological operations are:**

1.St_SideSegmentation- This software module segments a 3D model in to segments of LINESEGMENTS so that individually Query can be applied on the segmented model. Its utility can be well understood by an example of calculating distance between 2 buildings or finding width of a road between 2 buildings. In earlier scenario output was calculated based upon distance between centroid of the buildings which was not accurate, using this operation actual distance can be found between two sides of the building.

In other example where an operation has to be performed on one side of a building instead of the whole building like dynamically increasing the width of the side for visualization.

2.St_Rotatexyz- This software module is developed to rotate the 3D objects to an angle about x, y ,z coordinate axis. Angles are passed as parameters to the operation and it rotates the 3D object to the given angle in radians.

3.St_ExtrudePolygon- This module is developed for dynamically creating a 3D object at LOD2 from a 2D object by giving the height value. This operation is very important because it is alternative to various software tools e.g *ArcSceane*, *GoogleSketchup* etc, which are used create 3D model at LOD2 from the footprint.

New Python Modules developed are:

1) Python module for parsing the output KML files and customizing them by removing various tag characters, changing the tag structure and values.

2) Python module for parsing the output of attribute and location based spatial query and creating the actual 3D object.

3) Python module for generation of 3D object

New JavaScript Module developed is:

1) Java Script module to view 3D geo-data on web based viewer.

These software modules can be very beneficial for extracting information and for building exciting applications in the field of 3D modelling and augmented reality.

4.6 Performance study of GeoRDBMS and spatial indexing techniques for various file formats and generation of assessment report.

Due to large volume and complex data structure the 3D geo-data needs special attention while storing in GeoRDBMS to achieve expected performance during data access and query execution. Study of performance of GeoRDBMS becomes very important in deciding the choice of spatial indexing technique and various other optimization parameters that enhances the query execution performance. As spatial indexing techniques play an important role in handling large and complex data and for better performance of spatial queries, enhancing the performance of GeoRDBMS. Spatial indexes also play critical role in allowing effective access of these data sets in single and multi-user environment(Breunig and Zlatanova 2011b)(Sardadi et al. 2008b)(Hellerstein, Naughton, and Pfeffer 1995).

During this study the 18 major GIS operations are studied for 3D geo-data. GiST, RTree and BTree spatial indexing techniques are studied on the 3D geo-data. Performance study is carried out for file formats COLLADA, KML, CityGML in data access and query execution environment. Comparative study of performance is carried out for all the above mentioned file formats. Study of data loss is done during data migrating from file formats to GeoRDBMS. Study of spatial indexing techniques is done for all the formats on selected GIS operations. Impact of many factors like cardinality, frequency occurrence of values, bounding boxes, geometry type of 3Dgeo-data, working memory etc, on the performance of query execution is studied and a detailed assessment report is prepared. Execution time of spatial query is studied for various spatial indexing techniques for 3D file formats and the causes of the query execution time, impact of selection of spatial indexing technique, and other important factors on the performance of the query are explained in the assessment report. Study is also carried out to assess the impact and conditions of application of 3D spatial indexing technique, and 2D spatial indexing technique on 3D geo-data along with 2D and 3D variant of GIS operations.

Detailed study of query execution plan is done while doing performance study and based on the outcome of the report, optimization plan is applied on the 3Dgeo-data by developing front end application for optimization of GeoRDBMS. Operations like *ST_Force2D* and its implications on query execution time is studied when a 2D operation is applied on 3D geo-data.

A relation is deduced on why The 2D and 3D spatial operations show different performance with various indexing techniques and relation between data access time for the various file formats in GeoRDBMS and its causes is studied. In this study also carried out is the comparative study of selected four 2D and 3D spatial operations on the same data set. Advantages and disadvantages of application of various spatial indexing techniques on 3D geo-data is studied and scenarios are developed for application of correct spatial indexing techniques like RTree approach may be more efficient due to better maintenance of spatial immediacy but may be slow in updates or index creation and implements its own concurrency protocols (C. W. Q. C. A. Arens and J.E. Stoter 2003). A detailed report is generated as a whole for detailed performance analysis done on 3D geo-data inside GeoRDBMS.

4.7 Attribute and location based querying on 3D geo-data using front end application

In the current Scenario many vendors provide environment for storing 3D geo-data and also operations to perform spatial query on the data, but it is still in the early phases of performing attribute and location based 3D query. Many vendors have limitation of very few 3D spatial operations and also forcing the 3D data to 2D inside GeoRDBMS and then apply the spatial operation without considering the 3rd dimension.

In the present study attribute and location based querying is performed on actual 3D geo-data. For attribute and location based querying 2 methodologies are followed as shown in Figure 4.7.

1.Querying over 3D geo-data without dropping z index due to implication of 3D spatial operations in GeoRDBMS and getting 3D result of the query.

2. Querying over 3D geo-data by dropping the z value, due to implication of 2D spatial operations on 3D data, getting the query result and remodelling the 2D resultant data in to 3D geo-data with actual height values using pl/pgsql procedures and python scripts.

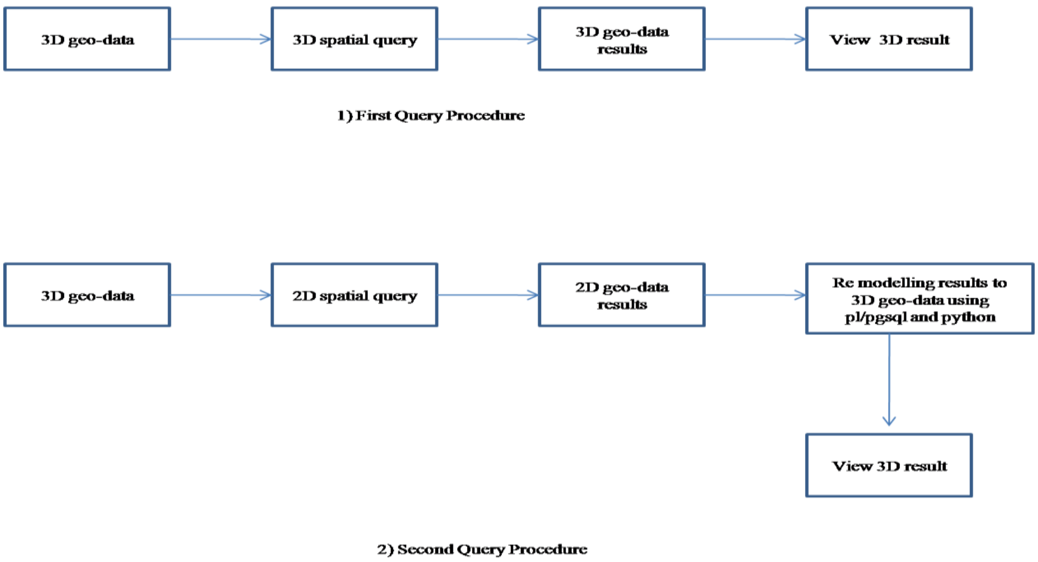


Figure 4.7 Methodologies followed for attribute and location based Query depending upon Spatial operations

A front end application is developed to query over the 3Dgeo-data in which various options are given for performing spatial and non spatial query. Along with query options application has options to select various optimization parameters which need to be applied on GeoRDBMS for tuning of GeoRDBMS and better query execution time.

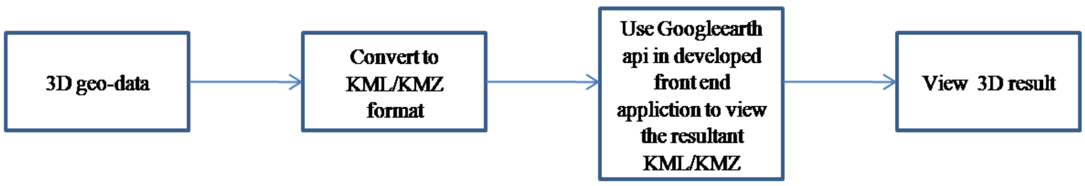
4.8 Viewing the results on web browser and Google earth

This is the last step of the methodology which is visualization part of the study, where 3D model and 3D results are shown on two viewers.

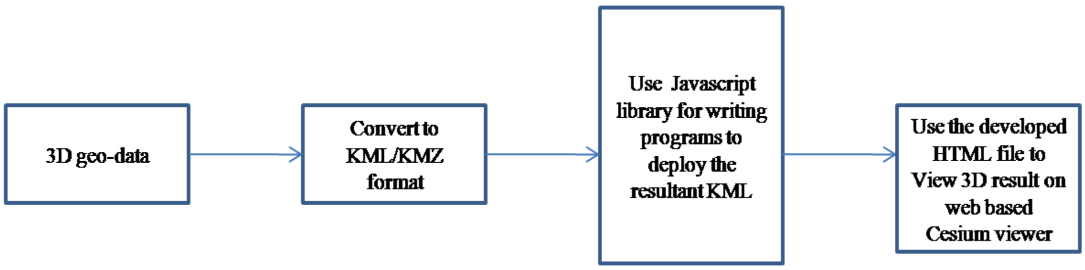
1) Earth explorer

2) On web using Virtual Globe

Different methodology are implemented to view the 3D data Figure 4.8 shows the methodologies followed to visualize the data on the web and on the Google earth.



1) Using Google earth API



2) Using Web based Cesium viewer

Figure 4.8 Methodologies followed for viewing the 3D results on cesium web viewer and Google earth API

3D geo-data results are converted to KML/KMZ format and Google earth API is used in the developed front end application to view 3D geo-data.

To view the 3D geo-data on web browser cesium viewer is used in the study. JavaScript library is used to develop resultant HTML file which is deployed on cesium globe using Apache Tomcat server.

5. Result and Discussions

The outcome of this study are described in this chapter with a discussion on different findings and their impact on 3D geo-data organization and management in GeoRDBMS.

5.1 Extracted footprint from high resolution satellite images

High resolution data access from various Geo-webservers *like Google, Bing and Bhuvangeospatial* is done as WFS in QGIS to allow digitization and extraction of building footprints as vector file format . Extracted footprint consists of in total 676 buildings blocks of 16 different types of corresponding structure, along with 359 in total line features like boundary walls and roads.The footprint is shown in Figure 5.1.

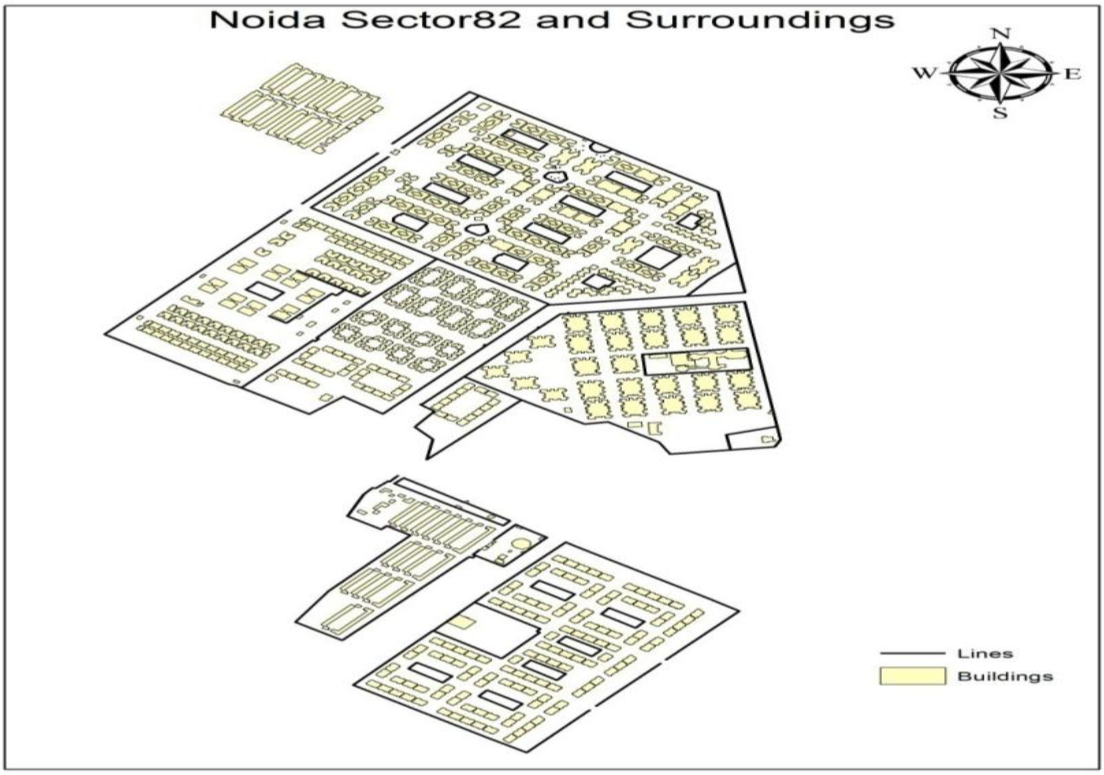


Figure 5.1 Extracted footprint of the study area

5.2 3D model of the study area

Development of 3D model at LOD0 and LOD4 is not in the scope of this study. However the 3D model at LOD1, LOD2, LOD3 is developed. In this process extracted footprint is migrated to *ArcSceane* and height and ground control point information collected during field survey shown in Figure 5.2.1 is used to create 3D model at LOD1. Developed 3D model at LOD1 is shown in Figure 5.2.2. Extracted footprint of the study area is migrated to *CityEngine* develop 3D model at LOD2 and LOD3.Developing a 3D model in *CityEngine* requires writing rules in CityEngine's procedural language structure of a rule file is shown in Figure4.1.2.2.Writing rules to develop a 3D model a block requires intensive effort and accuracy as rule file for a single building block consists of more than 2000 lines of code. Each building block has a separate rule file until the footprint are exactly the same in dimension. Images of the study area are captured during field survey is used to extract the information of structure of buildings, and texture the model.".*obj"* files areused for designing various 3D objects like windows and doors of the 3D model. Sample code of rule file is shown in Figure 5.2.3(a) & 5.2.3(b), developed 3D model at LOD2 is shown in Figure 5.2.4 and developed 3D model of some of the types of buildings at LOD3 is shown in Figure 5.2.5. A textured 3D 3D model of one of the building types is shown in Figure 5.2.6 and 3D geo-data in created in KML/KML, COLLADA format is shown in Figure5.2.7

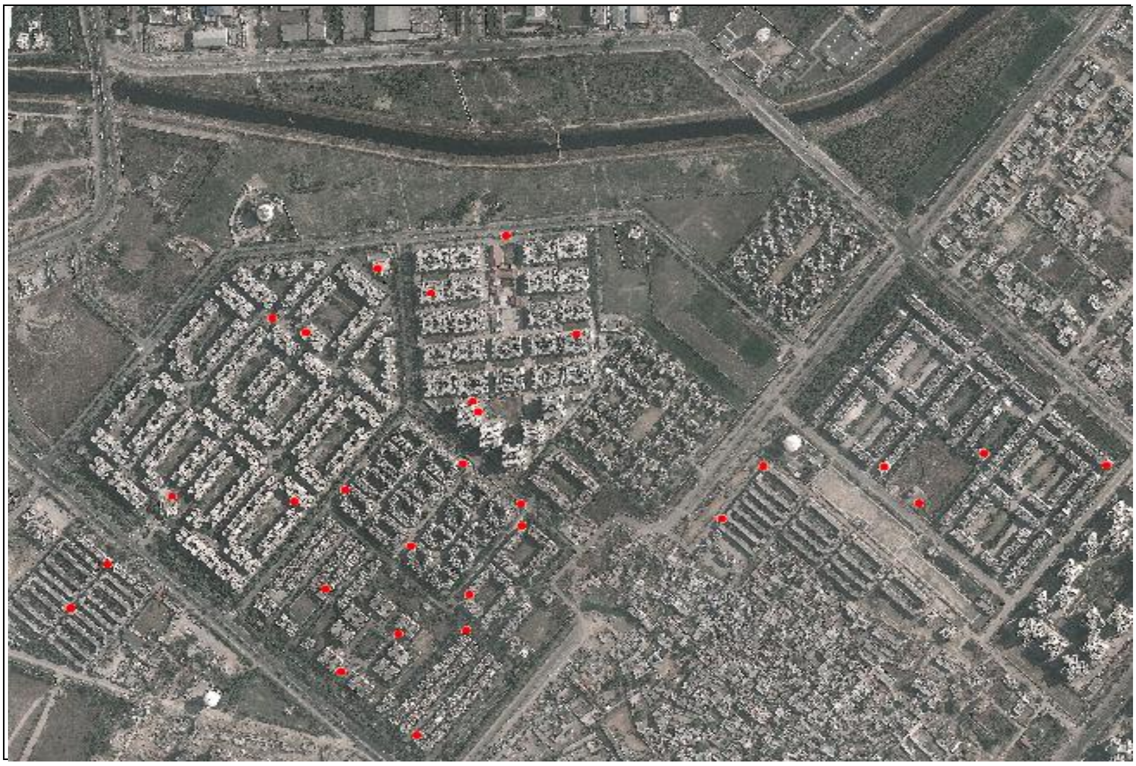


Figure 5.2.1 Collection of height, structure of the building Ground Control Point for validation of 3D data model during field validation

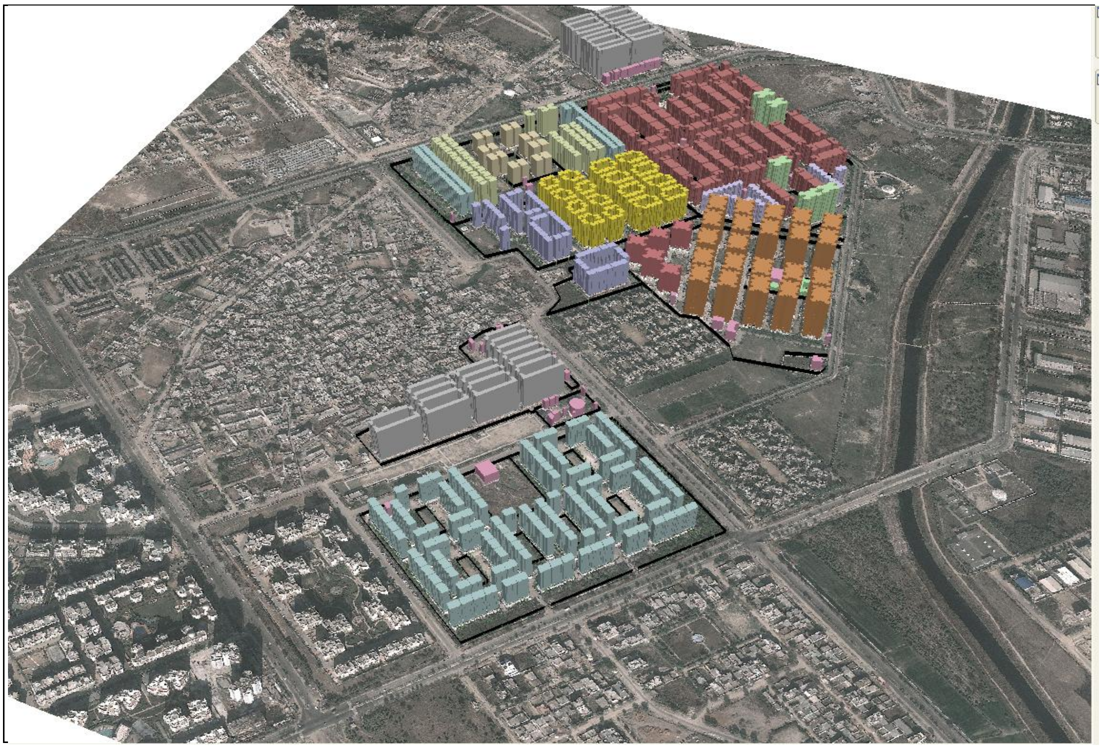
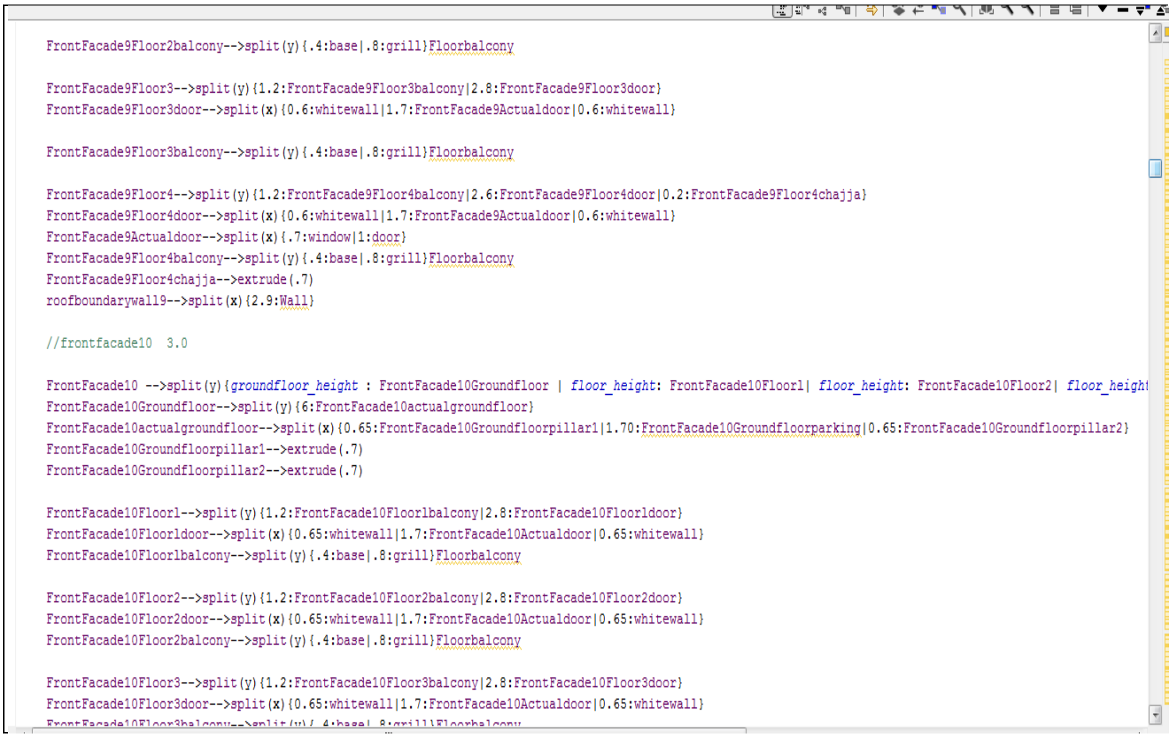


Figure5.2.2 3D model of the study area at LOD1



```
FrontFacade9Floor2balcony-->split(y){.4:base|.8:grill}Floorbalcony

FrontFacade9Floor3-->split(y){1.2:FrontFacade9Floor3balcony|2.8:FrontFacade9Floor3door}
FrontFacade9Floor3door-->split(x){0.6:whitewall|1.7:FrontFacade9Actualdoor|0.6:whitewall}

FrontFacade9Floor3balcony-->split(y){.4:base|.8:grill}Floorbalcony

FrontFacade9Floor4-->split(y){1.2:FrontFacade9Floor4balcony|2.6:FrontFacade9Floor4door|0.2:FrontFacade9Floor4chajja}
FrontFacade9Floor4door-->split(x){0.6:whitewall|1.7:FrontFacade9Actualdoor|0.6:whitewall}
FrontFacade9Actualdoor-->split(x){.7:window|1:door}
FrontFacade9Floor4balcony-->split(y){.4:base|.8:grill}Floorbalcony
FrontFacade9Floor4chajja-->extrude(.7)
roofboundarywall9-->split(x){2.9:Wall}

//frontfacade10  3.0

FrontFacade10 -->split(y){groundfloor_height : FrontFacade10Groundfloor | floor_height: FrontFacade10Floor1| floor_height: FrontFacade10Floor2| floor_height
FrontFacade10Groundfloor-->split(y){6:FrontFacade10actualgroundfloor}
FrontFacade10actualgroundfloor-->split(x){0.65:FrontFacade10Groundfloorpillar1|1.70:FrontFacade10Groundfloorparking|0.65:FrontFacade10Groundfloorpillar2}
FrontFacade10Groundfloorpillar1-->extrude(.7)
FrontFacade10Groundfloorpillar2-->extrude(.7)

FrontFacade10Floor1-->split(y){1.2:FrontFacade10Floor1balcony|2.8:FrontFacade10Floor1door}
FrontFacade10Floor1door-->split(x){0.65:whitewall|1.7:FrontFacade10Actualdoor|0.65:whitewall}
FrontFacade10Floor1balcony-->split(y){.4:base|.8:grill}Floorbalcony

FrontFacade10Floor2-->split(y){1.2:FrontFacade10Floor2balcony|2.8:FrontFacade10Floor2door}
FrontFacade10Floor2door-->split(x){0.65:whitewall|1.7:FrontFacade10Actualdoor|0.65:whitewall}
FrontFacade10Floor2balcony-->split(y){.4:base|.8:grill}Floorbalcony

FrontFacade10Floor3-->split(y){1.2:FrontFacade10Floor3balcony|2.8:FrontFacade10Floor3door}
FrontFacade10Floor3door-->split(x){0.65:whitewall|1.7:FrontFacade10Actualdoor|0.65:whitewall}
```

Figure5.2.3(a) Snippet code of a rule for a building block

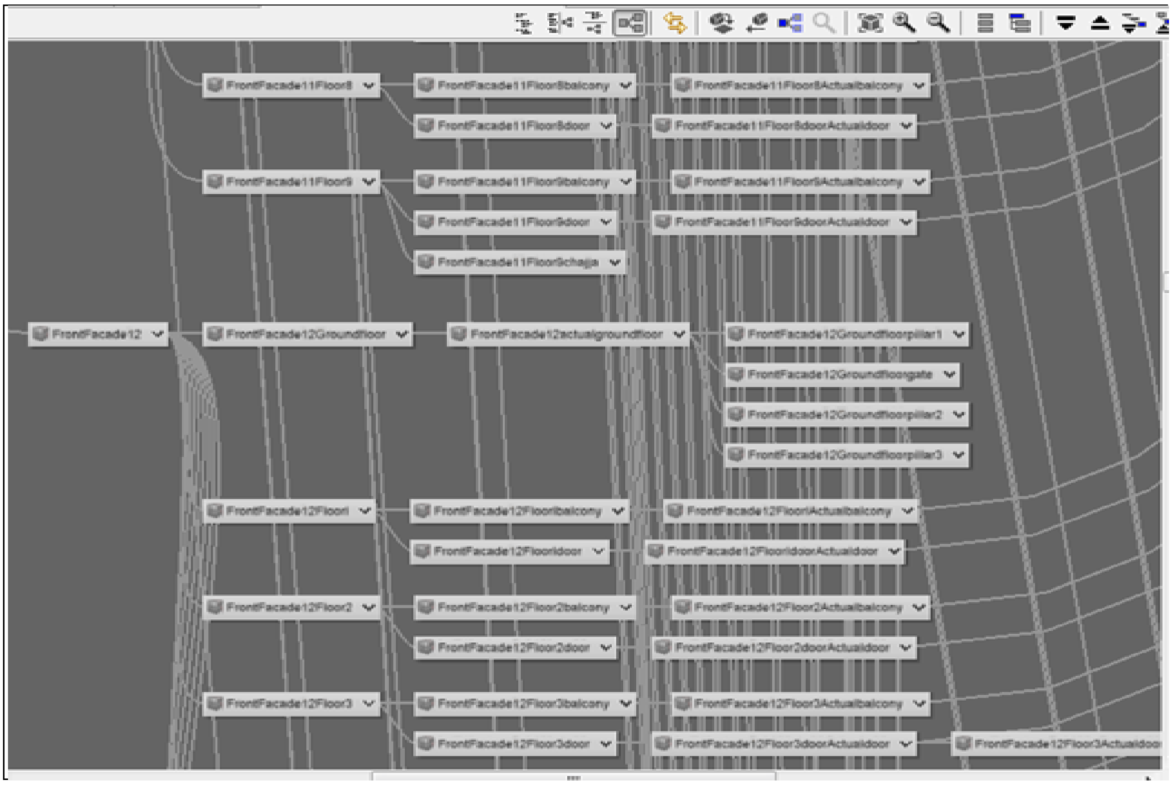


Figure 5.2.3(b) Structural design of a rule file

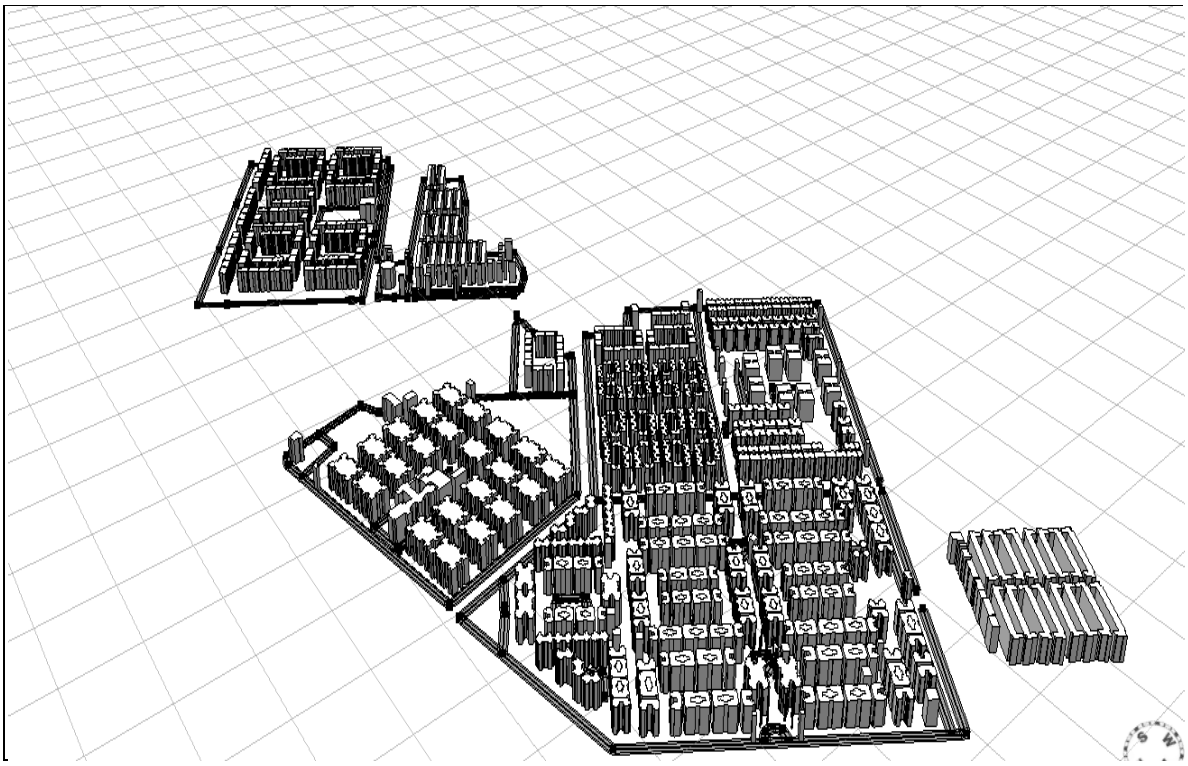


Figure 5.2.4 Developed 3D model in LOD2

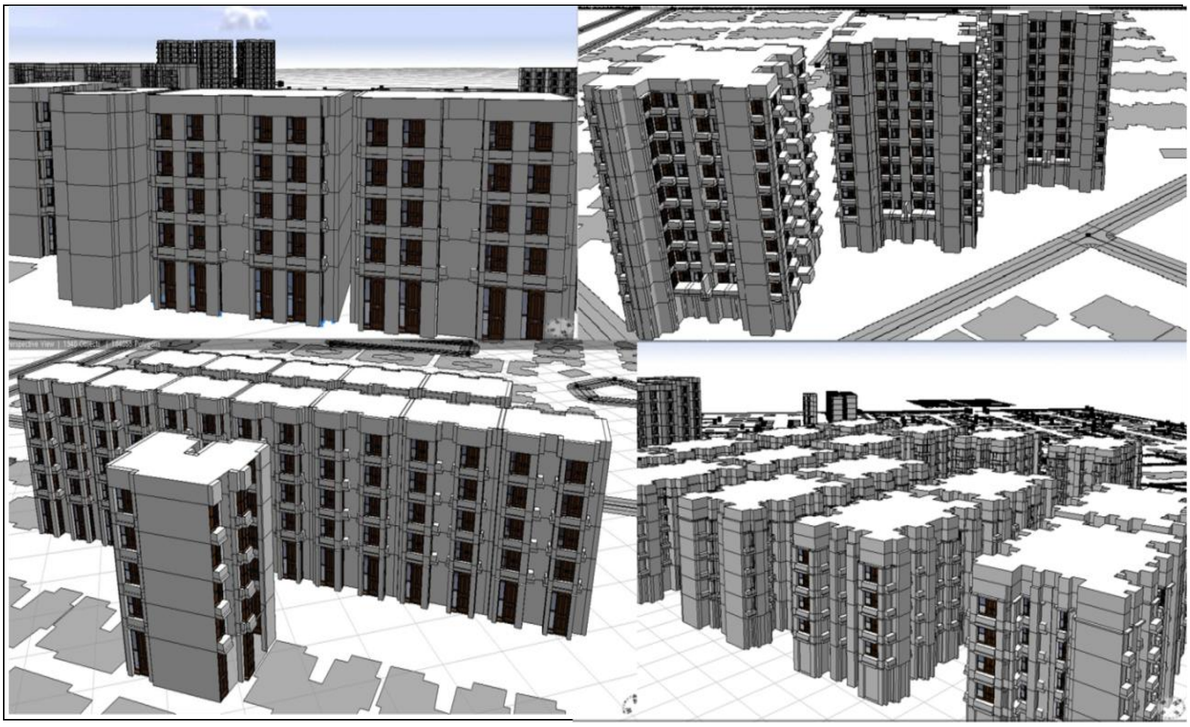


Figure 5.2.5 Developed 3D model of few of the types of buildings in LOD3

Figure 5.2.6 Textured 3D model of one of the type of 3D building

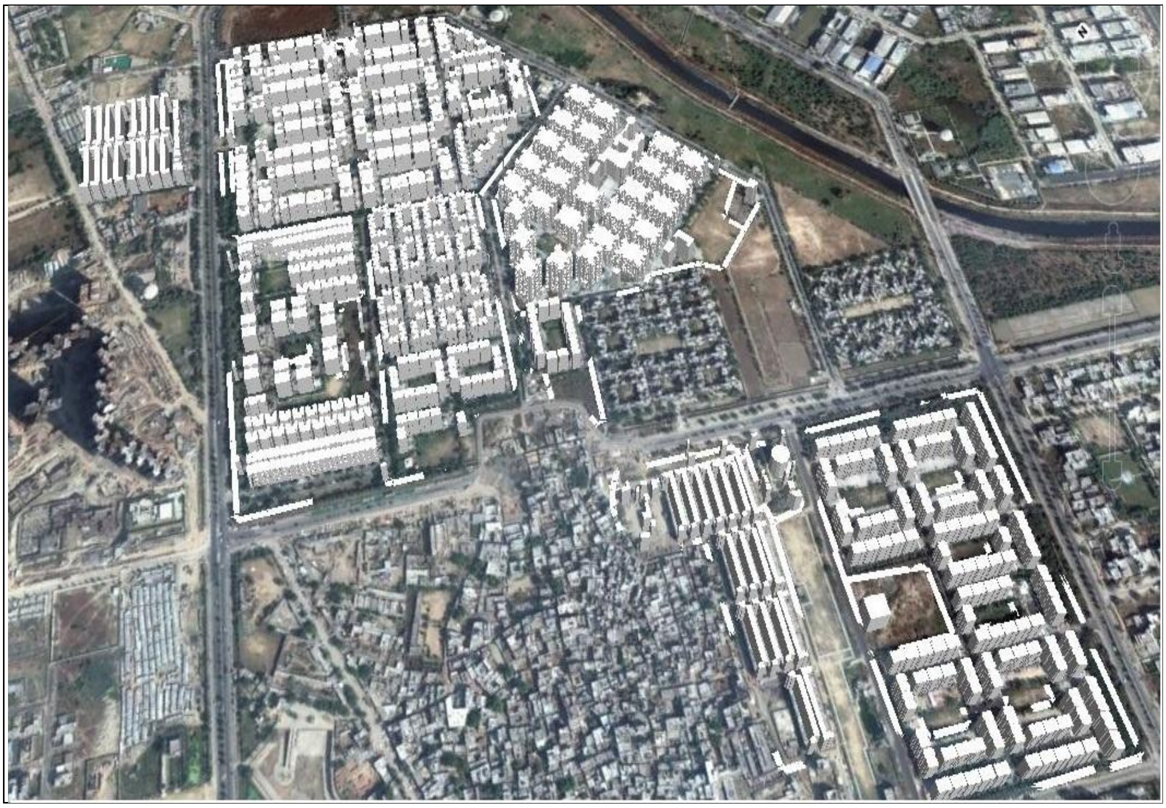


Figure5.2.7 Developed study area at LOD3 for file formats KML, CityGML, COLLADAformat

5.3 Performance Study of Various Spatial Indexes on 3Dgeo-data in GeoRDBMS

5.3.1 For Single User

The 3D geo-data created in CityGML, KML, COLLADA, are migrated to GeoRDBMS where various spatial indexing techniques are applied on these datasets. The performance assessment is presented below in Table2 to Table4 and Figure 5.3.1 to Figure 5.3.7.

During this study the 18 major GIS operations are studied for 3D geo-data. It is observed that all these formats are giving almost similar performance in data access and query execution (Table2, Table3, Table4, and Figure5.3.7). There is no data loss observed during data migrating from file formats to GeoRDBMS and also all file formats are having same data formatinsideGeoRDBMS.This also answers the research question "*Is there any data loss and are all file formats having same data format in GeoRDBMS*" mentioned in section 1.3.3(c). For execution of spatial queries only the bounding-box based operators like '&&' can take advantage of Gist spatial index. If minimum bounding box is not implemented for 3D geo-data while applying the spatial indexing than 3D operation will give very slow performance during data access e.g. *ST_3DDistance*, *ST_3DDWithin*.

In this study we have tested functions like *ST_3Ddistance, ST_3DClosestPoint* with indexing (GiST, R-tree, B-Tree) and without indexing. In this case the time taken for query execution on indexed data is more (Table1, Table2, Table3, and Figure 5.3.1 to Figure 5.3.7).

This delay in query executions by query can be reduced by using '*&&*' operator in the functions like *ST_DWithin*, *ST_MakeEnvelope*where Minimum Bounding Box is forcefully used for query execution. The '*&&*' operator use the index to quickly reduce the result set to only those geometries which have bounding boxes that overlap the "query box". If query box is smaller than the extent of the entire geometry then the time taken in the query execution will be less because this will reduce the number of calculations that need to be carried out. In some cases the functions which uses bounding box will not give expected performance in execution of query because query selection also has a big impact on performance e.g (*ST_3DIntersects).* In addition to this GROUPBY expression increases the total time. Cardinality has also impact on query execution time, because indexing technique may work efficiently only with either low cardinality or high cardinality as cardinality is used by the query engine to determine the optimal query plan to complete execution of a query. The query optimizer is cost-based, query plans that have the lowest estimated processing cost to execute. The query optimizer determines the cost of executing a query plan based factors like cardinality of the plan and cost model of the algorithm dictated by the operators used in the query. Distribution of a column is the frequency occurrence of each distinct value of the column, the column distribution guides us to determine which index type should be used. The range of values of indexed column also guides to select an appropriate indexing technique.

The 2D operations which can be applied on 3D data, internally calls *ST_Force2D* function which the projection of 3^{rd} dimension in X, Y plane and then applies the operation to supports bounding box for many operations. This process will also delay the query execution. In some cases GiST takes more time in PostgreSQL because query planner does not optimize the use of GiST indexes properly. In such cases the searches should use a spatial indexes properly rather than sequence scan of the whole table. It is also observed that PostGIS optimizer drops the applied spatial indexing technique for lesser number of pages, in those cases the query will not take any advantage of spatial indexing to enhance the performance of query operation. R-tree approach may be more efficient due to better maintenance of spatial immediacy but may be slow in updates or index creation and implements its own concurrency protocols. The advantage of GiST is saving only the 2D bounding box of large geometries which leads to a disadvantage for point geometries as an extensive overhead is produced. In this study we have also carried out the comparative study of selected four 2D and 3D spatial operations on the same data set. The results are shown in Table 5 to Table 13 and Figure5.3.8 to Figure5.3.17 .

The 2D and 3D spatial operations show different performance with various indexing techniques. The data access time is almost similar for different 3D file formats as shown in Table 5 to Table 13 and Figure5.3.8 to Figure5.3.17. The operations like calculation of length and area or ST_3DclosestPoint, ST_3Ddistance and ST_3Dlongestline in 3D geo-data takes more time in query execution because in this case the query also considers 3^{rd} dimension of geo-data for calculation. Also, these operations consider every point on geometry1 and geometry2 in the calculation which leads to higher processing time.

The 3D objects are stored as one multi-polygon or a set of polygons, where no relationship exists between the different 2D polygons which defines the object. In the process of inserting multipolygon for 3D geo-data in RDBMS, the data validation can't be performed for each insert operation which leads multiple entries of same coordinate values in the table. These multiple entries will reduce the performance of query execution and will also increase the risk of inconsistency in the database. This operation will also establish a 1: n relationship between the real world object and the number of records in the table. Therefore a more efficient administration of these large data sets are required to establish a 1:1 relationship between objects in reality and objects in the database.

In the case of *Maxdistance*operations as shown in Table 5 to Table 13, the 2D operation are taking more time than 3D operation, because the dimensions and variations of the values in the *x,y-plane* are larger than in the z-direction. The typical application of this can be a city plan which covers an area of 5x5 kilometers with buildings up to 50 meters heights are stored in database. When the query will be executed for this, it will try to find out all the objects in a specific x, y-region where 3D objects are on top of each other. Since the x and y coordinates are more selective than the z-coordinate. The selection of candidates by indexes for spatial query is a major factor for fetching the data from table and for the query execution time. The spatial indexes based on MBR will select more candidates than MBB based 3D indexing techniques. In such cases the MBB based 3D indexing techniques will not be more useful than MBR based 2D indexing techniques. In addition to this query optimizer of PostGIS also perform *Filter operation* for selection of candidates to enhance the performance of query execution. This study also answers the research question *"How will the choice of Indexing technique affect the performance of 3D GeoRDBMS, an do we need to create a new indexing technique for optimal performance of GeoRDBMS"* as mentioned in section 1.3.3(e).

5.3.1 Tables:

Table-2-Topological Operations and performance of spatial indexes in PostGIS(CityGML)

S. No.	**Operations**	**Btree Index** (sec)	**GiST Index** (sec)	**R-Tree Index** (sec)	**Without Index** (sec)
1	st_coveredby	44.355	26.908	31.586	57.97
2	st_covers	36.604	30.489	30.712	40.123
3	st_crosses	37.703	31.591	28.196	43.013
4	st_disjoint	30.579	29.462	34.554	33.344
5	st_contains	34.26	22.857	26.287	29.17
6	st_overlaps	30.47	29.16	33.938	28.816
7	st_equals	33.931	29.674	36.576	26.908
8	ST_3DClosestPoint	231.245	245.789	228.231	208.607
9	ST_3DIntersects	31.012	36.134	34.116	33.904
10	st_3ddfullywithin(2	33.68	21.174	20.692	35.798

S. No.	Operations	Btree Index (sec)	GiST Index (sec)	R-Tree Index (sec)	Without Index (sec)
	precision)				
11	st_3ddwithin(2 precision)	36.681	36.536	36.224	32.446
12	st_3ddistance	266.08	267.769	251.724	230.116
13	box3D	0.098	0.114	0.144	0.175
14	st_3dmakebox(only for points)	173.24	154.98	162.65	153.48
15	st_3dlongestline	301.23	273.47	282.34	354.99
16	ST_Boundary	0.384	0.321	0.334	0.356
17	ST_ExteriorRing	0.334	0.354	0.341	0.353
18	ST_3DMaxDistance	64.563	63.929	59.355	57.1

Table-3 Topological operations and performance of spatial indexes InPstGIS(KML)

S. No.	Operations	Btree Index (sec)	GiST Index (sec)	R-Tree Index (sec)	Without Index (sec)
1	st_coveredby	39.3	36.606	37.511	39.884
2	st_covers	35.99	29.741	31.032	33.991
3	st_crosses	32.703	31.591	26.915	30.196
4	st_disjoint	34.372	46.53	37.152	36.357
5	st_contains	34.47	27.857	28.432	29.287
6	st_overlaps	39.885	29.16	36.712	38.065
7	st_equals	34.571	31.674	32.196	29.164
8	ST_3DClosestPoint	250.87	234.567	268.82	206.432
9	ST_3DIntersects	30.425	28.134	29.237	27.948
10	st_3ddfullywithin(2 precision)	35.333	23.45	28.825	34.177
11	st_3ddwithin(2 precision)	36.681	39.516	35.749	30.446
12	st_3ddistance	261.283	257.607	258.724	248.116

S. No.	Operations	Btree Index (sec)	GiST Index (sec)	R-Tree Index (sec)	Without Index (sec)
13	box3D	0.091	0.104	0.133	0.158
14	st_3dmakebox(only for points)	152.24	147.98	164.65	183.69
15	st_3dlongestline	311.73	333.67	358.54	324.12
16	ST_Boundary	0.362	0.337	0.342	0.356
17	ST_ExteriorRing	0.343	0.377	0.345	0.343
18	ST_3DMaxDistance	60.21	59.129	56.581	56.309

Table-4 Topological operations and performance of spatial indexes In PostGIS(COLLADA)

S. No.	Operations	Btree Index (sec)	GiST Index (sec)	R-Tree Index (sec)	Without Index (sec)
1	st_coveredby	40.2	33.606	34.511	39.884
2	st_covers	38.129	29.341	32.132	33.991
3	st_crosses	31.613	30.531	26.915	33.296
4	st_disjoint	35.372	46.53	37.152	36.357
5	st_contains	33.47	27.857	28.432	29.287
6	st_overlaps	37.485	29.16	36.712	38.065
7	st_equals	34.571	31.674	32.196	29.164
8	ST_3DClosestPoint	250.87	244.767	269.82	206.432
9	ST_3DIntersects	26.425	28.134	23.237	22.848
10	st_3ddfullywithin(2 precision)	36.333	39.45	38.825	34.177
11	st_3ddwithin(2 precision)	37.681	39.516	35.749	30.446
12	st_3ddistance	254.243	257.607	258.724	251.32
13	box3D	0.093	0.109	0.129	0.147
14	st_3dmakebox(only for points)	154.14	146.98	166.55	183.69
15	st_3dlongestline	331.73	342.67	356.54	321.62
16	ST_Boundary	0.362	0.337	0.342	0.356

S. No.	Operations	Btree Index (sec)	GiST Index (sec)	R-Tree Index (sec)	Without Index (sec)
17	ST_ExteriorRing	0.343	0.377	0.345	0.343
18	ST_3DMaxDistance	63.21	61.324	64.381	55.201

Table5-Performance detail of 2D and 3D (CityGML) using GiST

S. No	Operations	GiST (msec) 2D	GiST (msec) 3D
1	Perimeter	25	30
2	Length	24	26
3	Maxdistance(small data)	470	85
4	shortestline	190	275
5	Maxdistance(large data)	45345	3337

Table6-Performance detail of 2D and 3D (CityGML) using R-Tree

S. No	Operations	R-Tree(msec) 2D	R-Tree(msec) 3D
1	Perimeter	30	40
2	Length	31	36
3	Maxdistance(small data)	470	65
4	shortestline	200	360
5	Maxdistance(largedata)	46733	3315

Table7-Performance detail of 2D and 3D(CityGML) without index

S. No	Operations	Without Index(msec)2D	Without Index(msec) 3D
1	Perimeter	35	45
2	Length	42	56
3	Maxdistance(small data)	480	75
4	shortestline	220	280

S. No	Operations	Without Index(msec)2D	Without Index(msec) 3D
5	Maxdistance(large data)	47651	3765

Table8-Performance detail of 2D and 3D (KML) using GiST

S. No	Operations	GiST (msec)2D	GiST (msec)3D
1	Perimeter	30	36
2	Length	24	28
3	Maxdistance(small data)	473	88
4	shortestline	156	283
5	Maxdistance(large data)	44245	3177

Table 9-Performance detail of 2D and 3D (KML) using R-Tree

S. No	Operations	R-Tree (msec)2D	R-Tree (msec)3D
1	Perimeter	33	40
2	Length	26	46
3	Maxdistance(small data)	502	67
4	shortestline	221	394
5	Maxdistance(large data)	46833	3214

Table10-Performance detail of 2D and 3D (KML) without index

S. No	Operations	Without Index (msec)2D	Without Index (msec)3D
1	Perimeter	35	45
2	Length	43	63
3	Maxdistance(small data)	496	78
4	shortestline	217	289
5	Maxdistance(large data)	49151	3371

Table11-Performance detail of 2D and 3D (COLLADA) using GiST

S. No	Operations	GiST(msec) 2D	GiST(msec) 3D
1	Perimeter	29	36
2	Length	24	28
3	Maxdistance(small data)	481	89
4	shortestline	226	395
5	Maxdistance(large data)	47245	3477

Table 12-Performance detail of 2D and 3D (COLLADA) using R-Tree

S. No	Operations	R-Tree (msec)2D	R-Tree (msec)3D
1	Perimeter	41	52
2	Length	36	53
3	Maxdistance(small data)	402	73
4	shortestline	221	394
5	Maxdistance(largedata)	46833	3214

Table 13-Performance detail of 2D and 3D (COLLADA) without index

S. No	Operations	WithoutIndex (msec)2D	WithoutIndex(msec)3D
1	Perimeter	32	45
2	Length	33	72
3	Maxdistance(small data)	436	79
4	shortestline	221	304
5	Maxdistance(large data)	41351	3671

5.3.1 Figures:

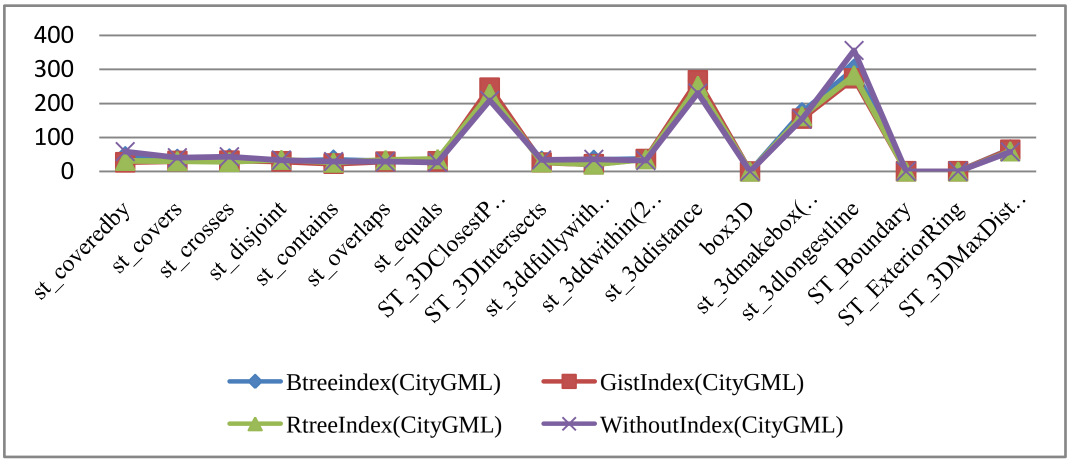


Figure 5.3.1 Line chart of operations vs. time taken by Spatial indexes for CityGML

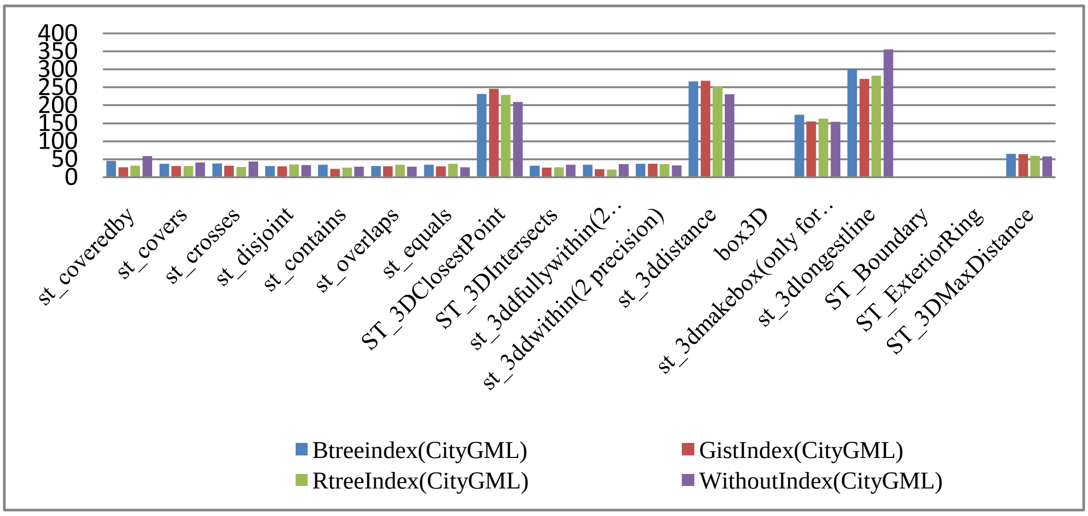


Figure 5.3.2 Bar Chart of operations vs. time taken by spatial indexes for CityGML

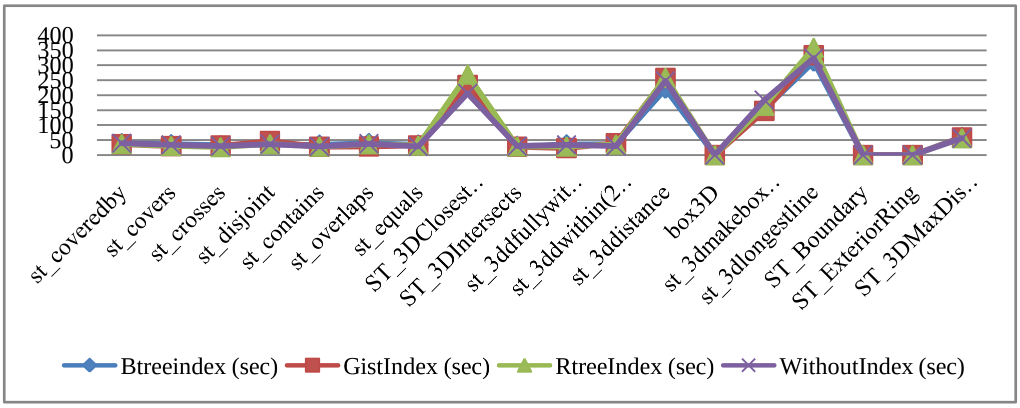


Figure 5.3.3 Line chart of operations vs. time taken by spatial indexes for KML

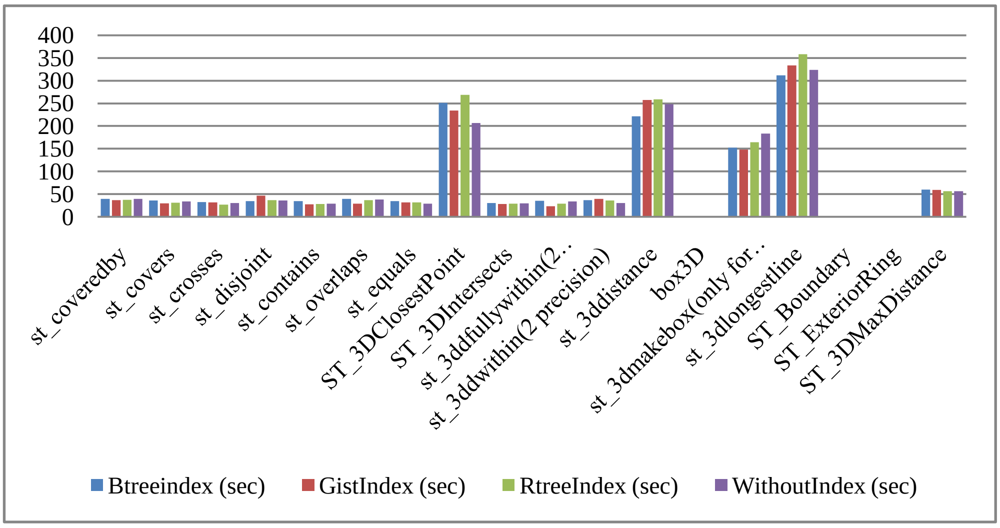


Figure 5.3.4 Bar Chart of operations vs. time taken by spatial indexes for KML

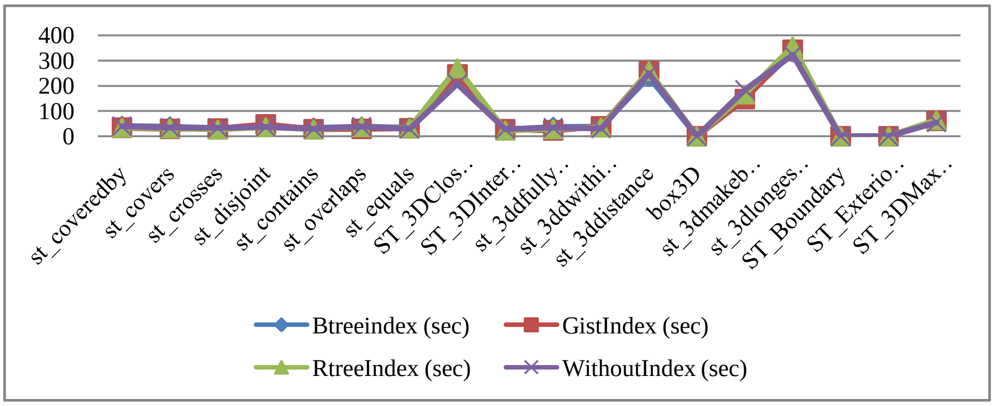


Figure 5.3.5 Line chart of operations vs. time taken by spatial indexes for COLLADA

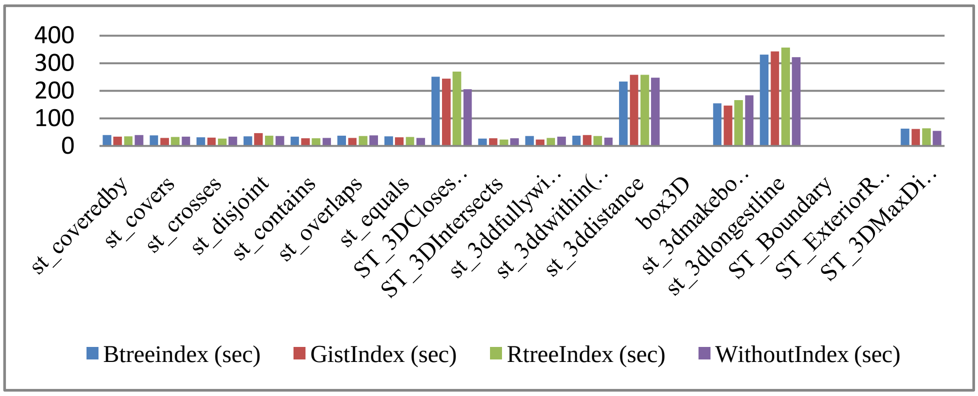


Figure 5.3.6 Bar chart of operations vs. time taken by spatial indexes for COLLADA

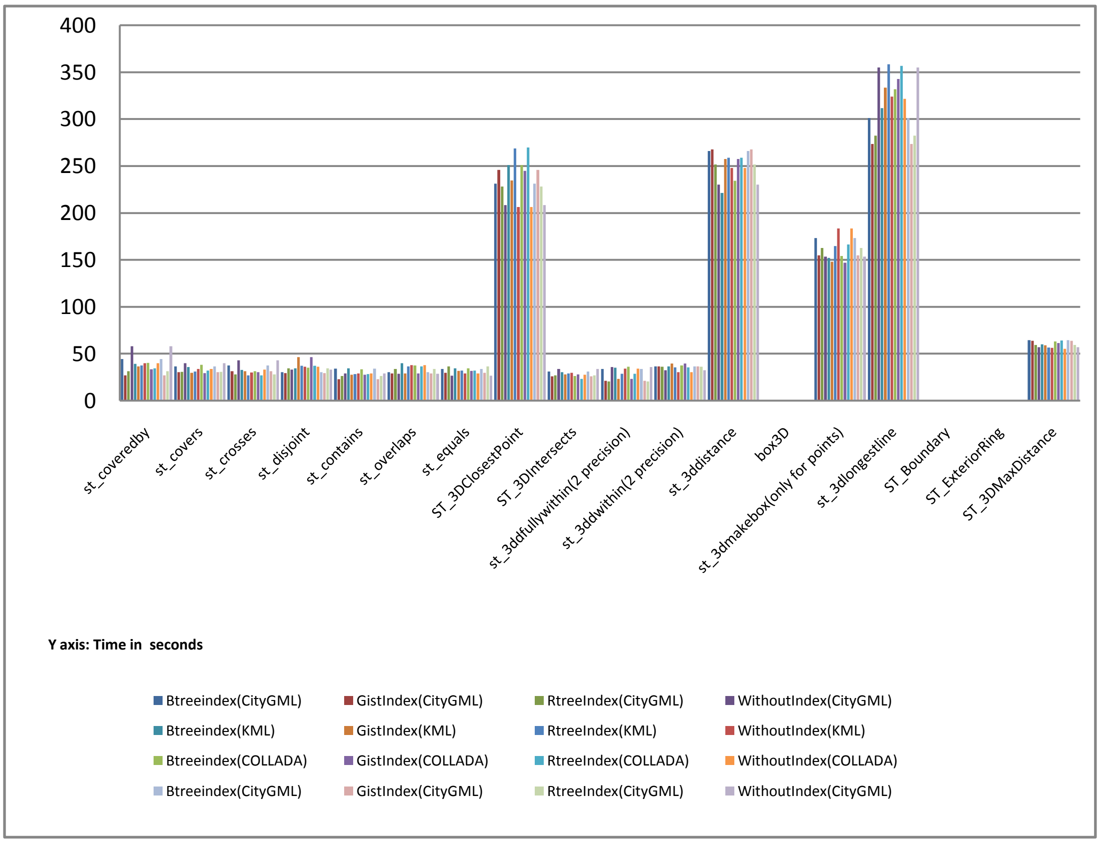


Figure 5.3.7 Bar chart of operations vs. time taken by spatial indexes for KML, COLLADA, CityGML

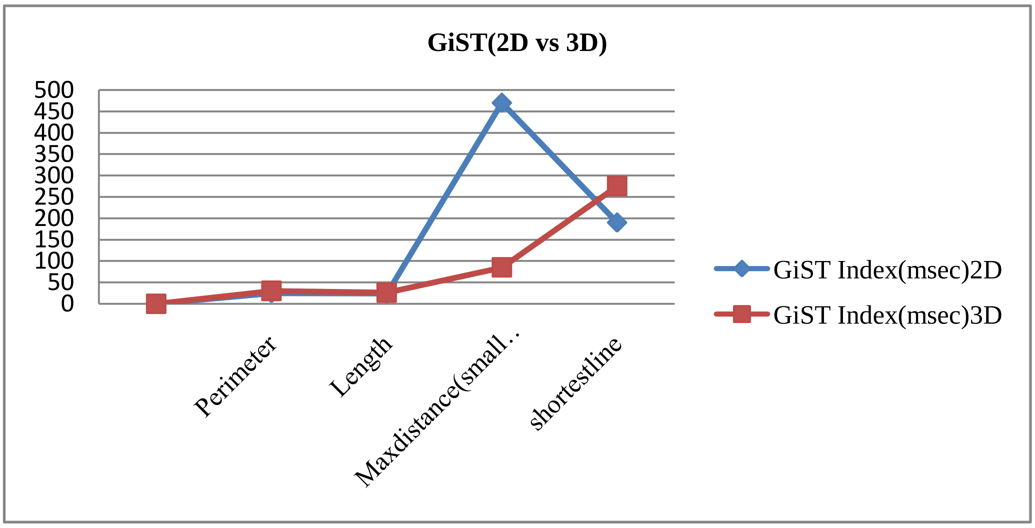


Figure 5.3.8 Line chart of Performance detail of 2D and

3D (CityGML) using GiST indexing

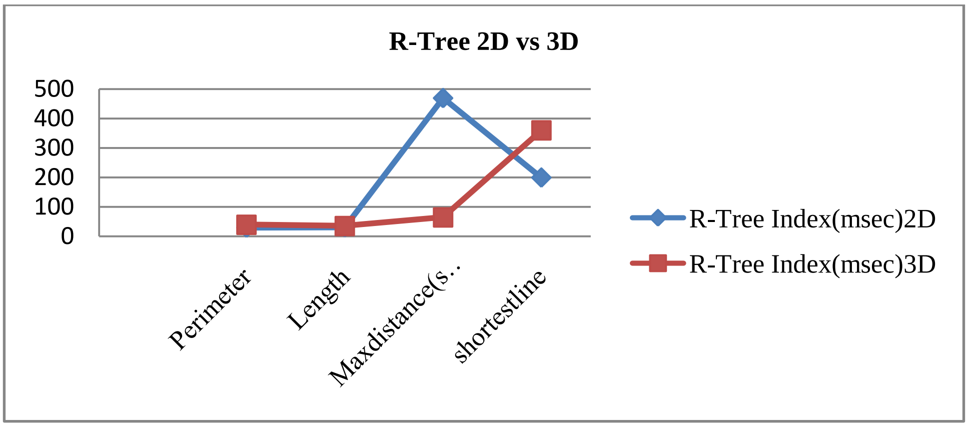


Figure 5.3.9 Line chart of Performance detail of 2D and3D(CityGML) using R-Tree indexing

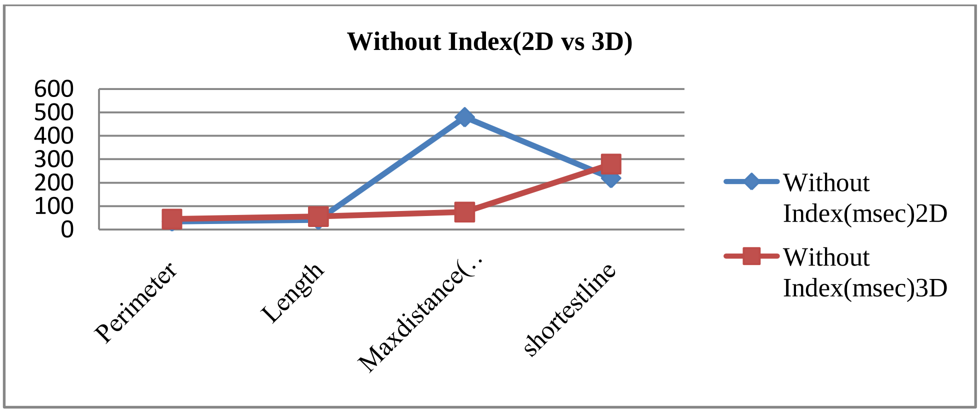


Figure 5.3.10 Line chart of Performance detail of 2D and3D (CityGML) without indexing

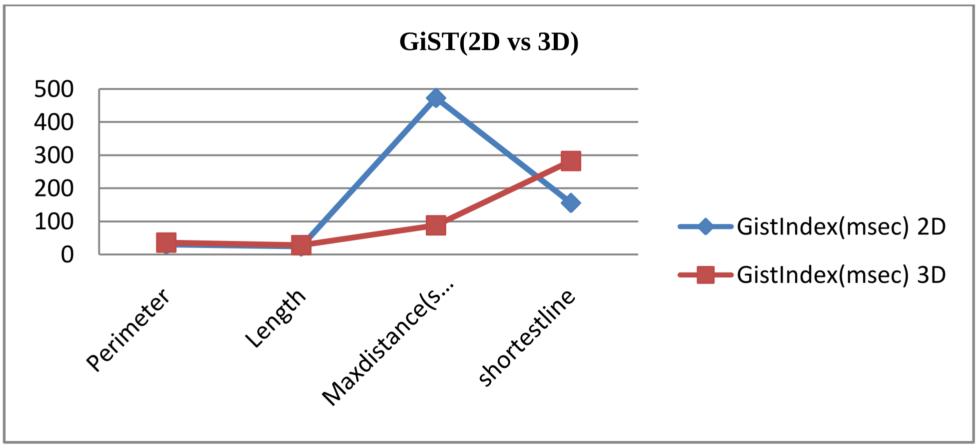


Figure 5.3.11 Line chart of Performance detail of 2D and3D (KML) using GiST indexing

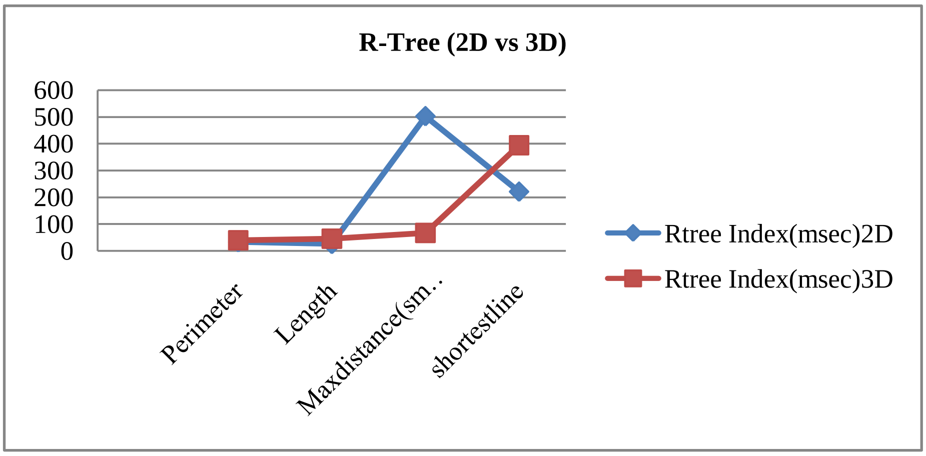


Figure 5.3.12 Line chart of Performance detail of 2D and 3D (KML) using R-Tree indexing

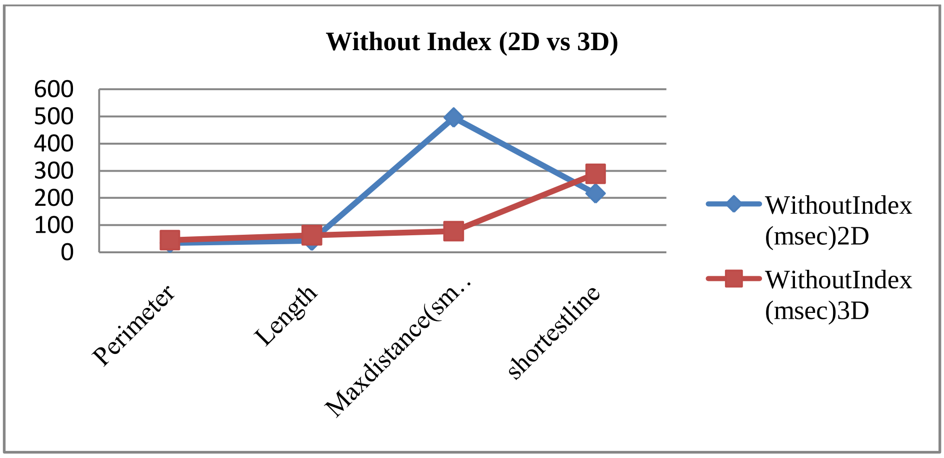


Figure 5.3.13 Line chart of Performance detail of 2D and 3D (KML) without indexing

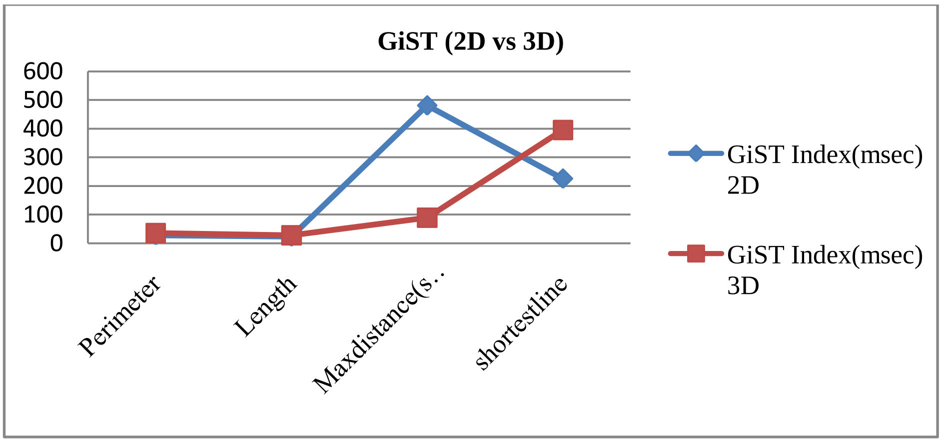


Figure 5.3.14 Line chart of Performance detail of 2D and3D

(COLLADA) using GiSTindexing

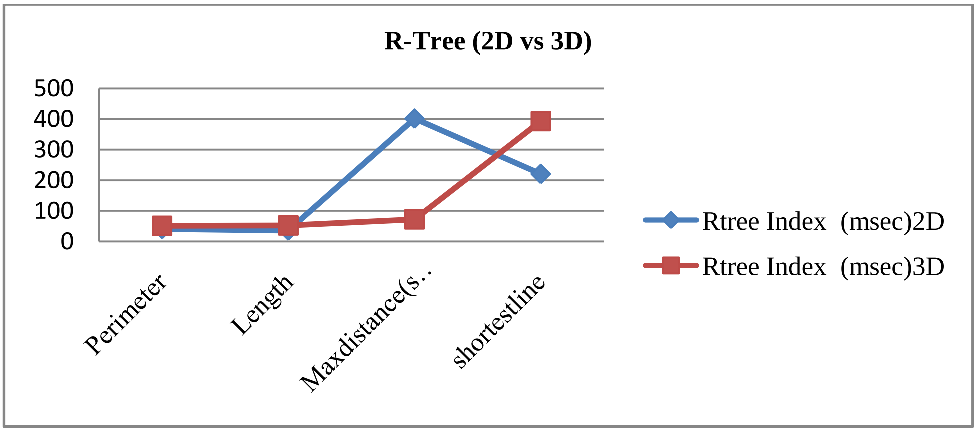


Figure 5.3.15 Line chart of performance detail of 2D and3D (COLLADA) using R-Tree indexing

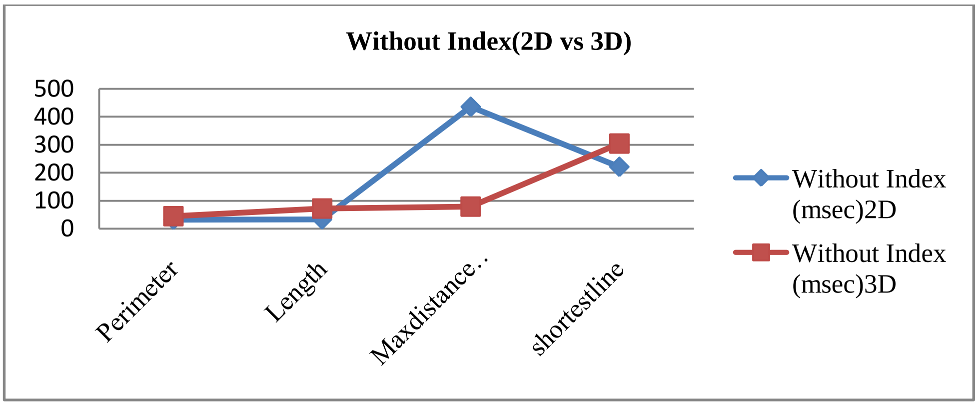


Figure 5.3.16 Line chart of performance detail of 2D and3D (COLLADA) without indexing

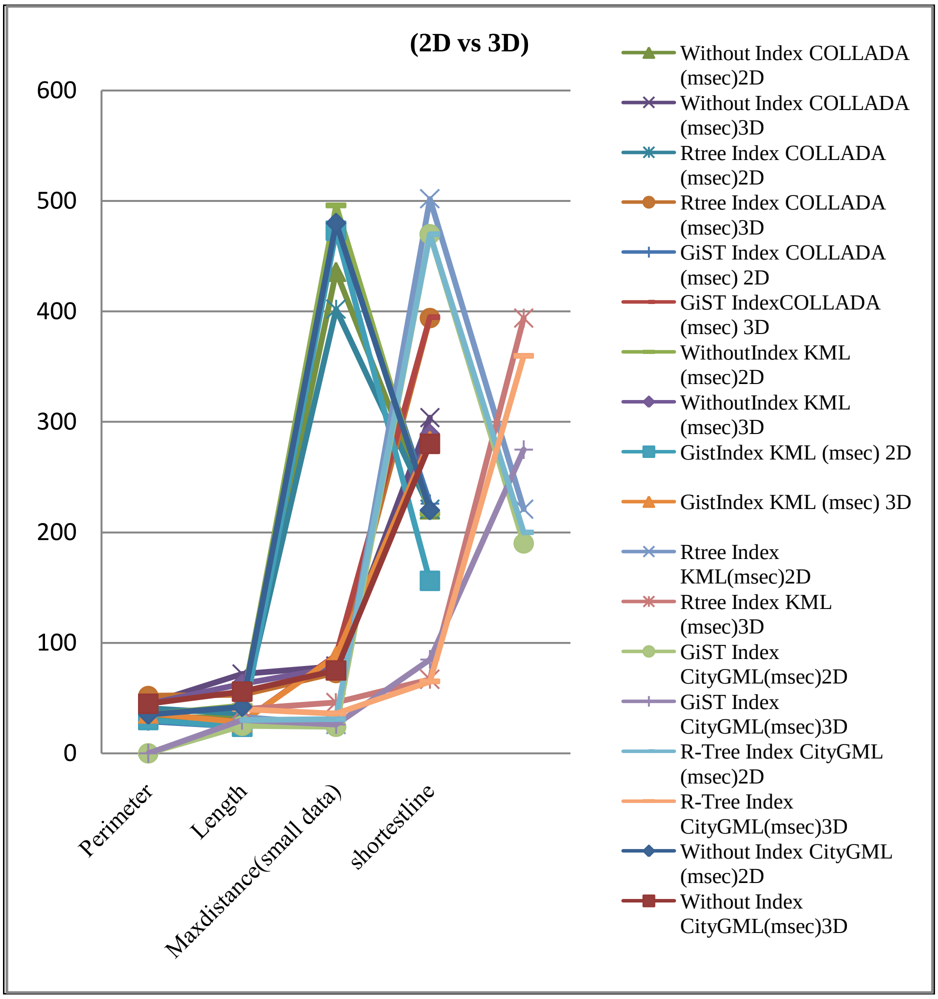


Figure 5.3.17 Line chart of performance detail of

2D and 3D for KML, COLLADA, CityGML

The outcome of the study presented above is accepted to publish in International Journal of Geographical Information Science Taylor& Francis Group , published from UK

5.3.2 For Multi-User

The 3D geo-data created in CityGML, KML, COLLADA, are migrated to GeoRDBMS where various spatial indexing techniques are applied on these datasets. The performance assessment is presented below in Table14 to Table16.

During this study the 4 major GIS operations are studied for 3D geo-data in multi-user environment, and results are compared to performance of spatial query and GeoRDBMS in single user. For this study developed front end software application is used to connect to database and execute the spatial queries. Snapshot of the front end application is shown in figure 5.4.1 to 5.4.4.Multiuser environment is created by running the tool in different systems as a client and execute the spatial query on database residing on the server side by connecting over a network. For this study 5 clients were created to run spatial queries over a centralized database. Same queries were executed on all the five clients with minimum delay and *average* time of query execution for all the clients are taken as time of execution of that query.

5.3.2 Tables:

Table 14-Topological Operations and performance of spatial indexes for Multi user access in PostGIS(CityGML)

S.No.	**Operations**	**GiST Index** (sec)	**R-Tree Index** (sec)	**Without Index** (sec)
1	ST_coveredby	207.32	224.65	421.77
2	ST_Covers	210.11	195.23	397.21
3	ST_3DClosestPoint	2162.39	2207.96	3082.59
4	ST_3DIntersects	309.87	316.11	309.87

Table 15- Topological Operations and performance of spatial indexes for Multi user access in PostGIS(KML)

S.No.	**Operations**	**GiST Index** (sec)	**R-Tree Index** (sec)	**Without Index** (sec)
1	ST_Coveredby	307.32	271.20	391.86
2	ST_Covers	315.78	324.55	406.72
3	ST_3DClosestPoint	1674.39	1841.81	2223.96
4	ST_3DIntersects	283.01	331.11	369.17

Table 16- Topological Operations and performance of spatial indexes for Multi user access in PostGIS(COLLADA)

S.No	**Operations**	**GiST Index**(sec)	**R-Tree Index** (sec)	**WithoutIndex** (sec)
1	ST_Coveredby	231.45	185.99	395.63
2	ST_Covers	210.73	219.33	307.41
3	ST_3DClosestPoint	2271.28	1953.83	3121.14
4	ST_3DIntersects	219.38	312.14	341.69

The results of above study shows that performance of the GeoRDBMS decreases considerably with the increased number of clients. When compared to query execution time of the same operation in single user access, multiuser access of the 3D geo-data takes more time which is 6 to 7 times compared to single user environment. This shows that the multiuser access of 3D geo-data in GeoRDBMS is a challenge in terms of performance enhancement. The GeoRDBMS researchers has to focus on development of new indexing techniques and data access methods for enhanced performance in multiuser access. However, the results presented in this study for multiuser access needs more work for benchmarking and performance assessment. The enhancement will require in data storage by using *Geometry and Topology* data type, the multiuser data access in a client server environment also depends on front end application, network bandwidth and other network resources.

However, the performance of database server is very critical to complete the execution of any query. The configuration of data servers in terms of virtual memory, caching, and other database server parameters will also play an important role in enhancing the performance of GeoRDBMS. The futuristic concept of high performance computing, new spatial indexes and new 3Ddata types like 3D free form curves to reduce the complexity and amount of data produced will also play an important role to process complex 3D geo-data in multiuser environment. This also answers the research question "*do we need to create other data type for storing topology of 3D geo-data like 3D freeform curves*" as mentioned in section *1.3.3(d).*

The distributed database systems is another important architecture which can handle large databases using declarative data centric programming language like ECL. Some of the interesting projects like *MapReduce from Google* and *ApacheHadoop* can also be explored to further process and analysis of complex 3D geo-data.

5.4 Front end software application development for 3D GIS operations and GeoRDBMS optimization

In this study one of the most important objective is to develop a front end tool which has modules to manually optimize the GeoRDBMS by selecting various optimization parameters, applying appropriate spatial indexing technique manually and automatically based on the type of query execution for performance enhancement of GeoRDBMS and improved query execution time. Application is developed using PyQt library in Python language. Developed application has the functionality to connect to GeoRDBMS as shown in Figure 5.4.1 and selecting the 18 most important optimization parameters including selection of spatial indexing technique and option of optimizing the query execution by clicking on the optimize button any time during the execution. Optimization parameters are selected based on *performance study of GeoRDBMS* as discussed in section 5.3 and based on study of research articles. Snapshot of the optimization window is shown in Figure 5.4.2 and snapshot of the part of the code developed is shown in Figure 5.4.3. Application is developed in such a way that according to the type of query, l *automatic selection of best spatial indexing* technique is done based on the results of *performance study of GeoRDBMS* as discussed in section 5.3. Application has option of performing various spatial queries by selection of the tables in the GeoRDBMS. It also has option of perform queries on developed modules like attribute and location based queries, segmentation of the 3D model and rotation of the 3D models. *As per the literature review, there is no such implementation of such modules done before, and these developed software modules can be of great importance in the field of 3D GIS.* Application has 2 options of viewing the resultant 3D geo-data 1) on Google earth API 2) Web based 3D virtual globe as shown in Figure 5.4.4.

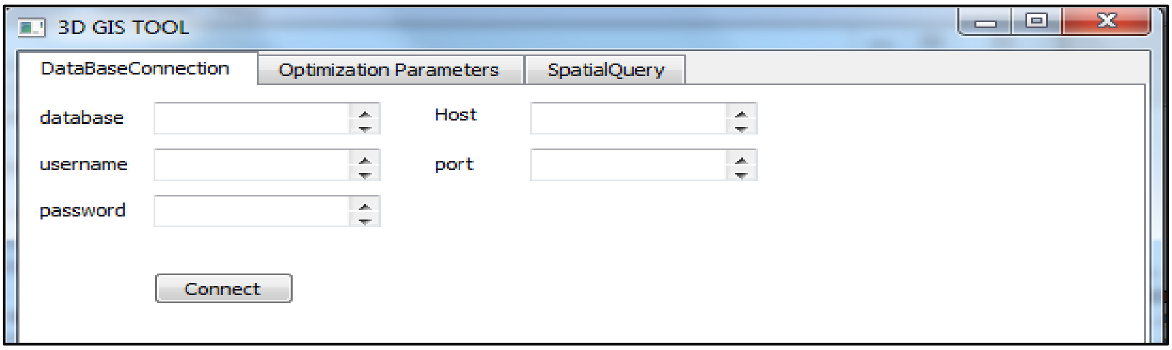


Figure 5.4.1 Interface for connection to GeoRDBMS

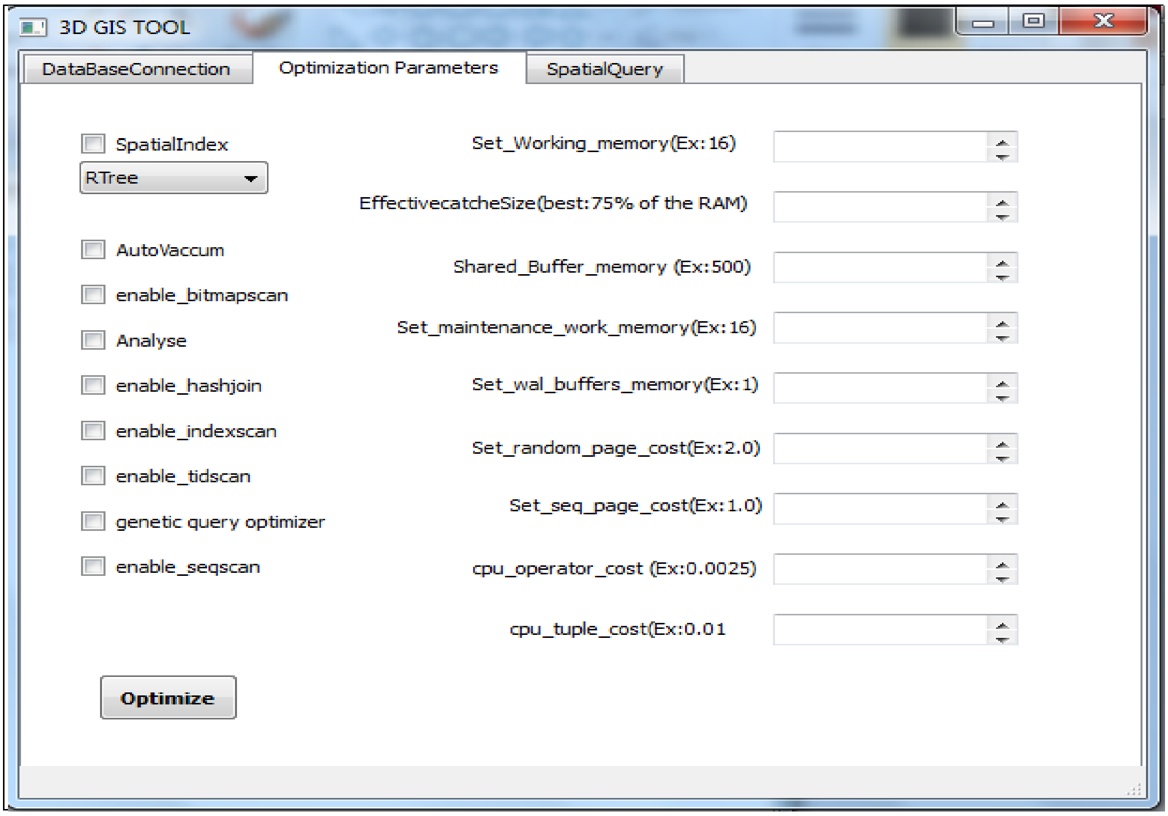


Figure 5.4.2 Interface to optimize GeoRDBMS

```python
    def Insertdata(self):
        try:
                if self.SpatialIndex.checkState() == QtCore.Qt.Checked and self.count==0:
##                      combocur.execute('CREATE INDEX spatindex ON cityobject USING rtree(envelope);')
##                      self.toappendstr=self.toappendstr+'Spatial Index Created Successfully\n'
##
                if self.Autovaccum.checkState() == QtCore.Qt.Checked:
                    combocur.execute("VACUUM (VERBOSE, ANALYZE) 3dtest;")
                    self.conn.set_isolation_level(old_isolation_level)
                    self.toappendstr=self.toappendstr+'Auto Vaccuming Intiated with VERBOSE AND ANALYZE\n'
                if self.enable_bitmapscan.checkState() == QtCore.Qt.Checked:
                    combocur.execute("set enable_bitmapscan=on;")
                    self.toappendstr=self.toappendstr+'Bitmapscan Enabled\n'
                if self.Analyse.checkState() == QtCore.Qt.Checked:
                    combocur.execute("ANALYZE VERBOSE 3dtest;")
                    self.toappendstr=self.toappendstr+'Analyzed Verbose Created\n'
                print '------------------------------------------------verbose initiatied----------------------'

                if self.enablehashjoin.checkState() == QtCore.Qt.Checked:
                    combocur.execute("set enable_hashjoin=on;")
                    self.toappendstr=self.toappendstr+'Hashjoin Enabled\n'
                print '------------------------------------------------hashjoin initiatied----------------------'

                if self.enabletidscan.checkState() == QtCore.Qt.Checked:
                    combocur.execute("set enable_tidscan=on;")
                    self.toappendstr=self.toappendstr+'Tids_Scan Enabled\n'
                print '------------------------------------------------tidscan initiatied----------------------'

                if self.enableindexscan.checkState() == QtCore.Qt.Checked:
                    combocur.execute("set enable_indexscan =on;")
                    self.toappendstr=self.toappendstr+'IndexScan Enabled\n'
                print '------------------------------------------------IndexScan initiatied----------------------'

                if self.geqop.checkState() == QtCore.Qt.Checked:
                    combocur.execute("set geqo=on;")
                    self.toappendstr=self.toappendstr+'genetic_query_optimizer\n'
                print '------------------------------------------------genetic query optimizer initiatied----------------------'

                if self.enablesqscan.checkState() == QtCore.Qt.Checked:
                    combocur.execute("set enable_seqscan =on;")
                    self.toappendstr=self.toappendstr+'enable_seqscan Enabled\n'
                print '------------------------------------------------Enable Seqscan initiatied----------------------'
```

Figure 5.4.3 Snippet of python code written to handle optimization parameters

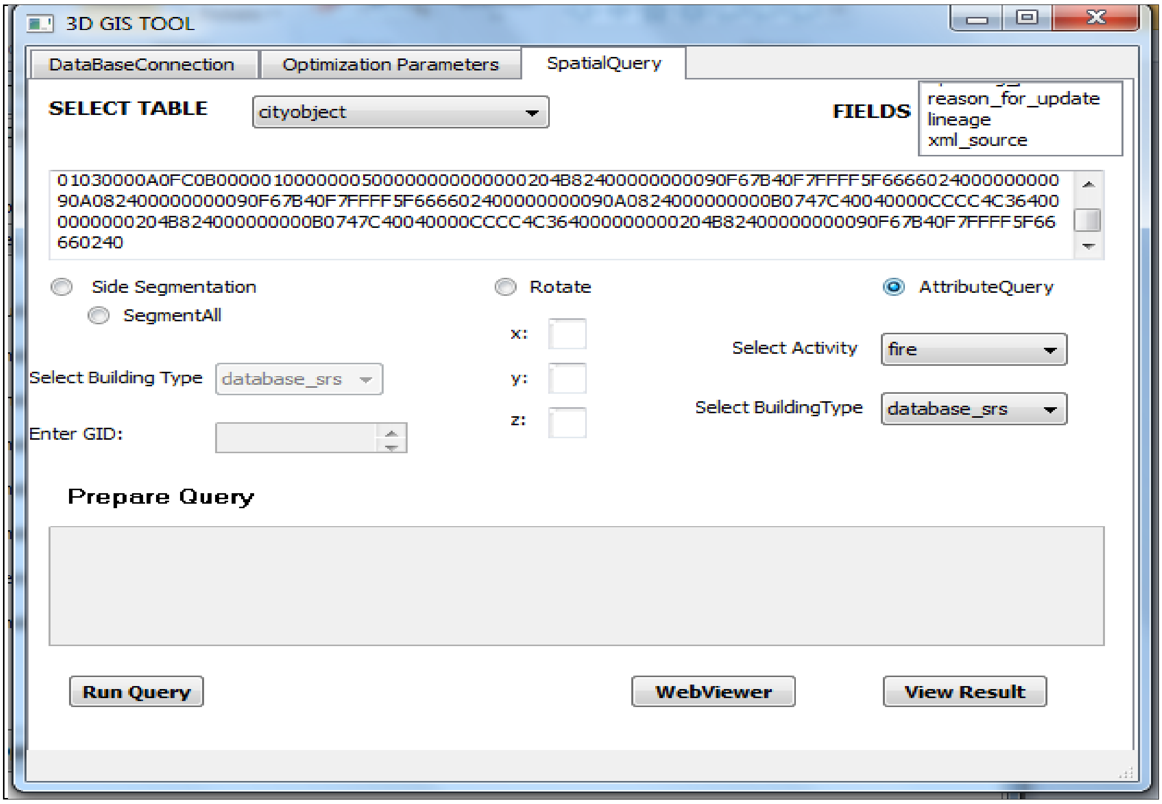


Figure 5.4.4 Interface to perform spatial query

5.5 Creation of spatial operations and software modules

In the present study new spatial and topological operations are developed and applied on created 3D geo-data so that these spatial operations helps in building various exciting augmented reality applications. **Developed spatial and topological operations are:**

1) St_SideSegmentation- This software module segments a 3D model in to segments of LineStringZ without losing the actual height information. Further individual query can be applied on the segmented model. Output of the operation is saved as KML/KMZ format Further python is used to develop software modules for parsing the output KML/KMZ files and customizing them by removing various tag characters, changing the tag structure and values to make it in a format to be viewed on *earth explorer* and web based viewer. Snapshot of the spatial operation and python module is shown in Figure 5.5.1to 5.5.2.

```
-- Function: "St_SideSegmentation"(character varying, character varying)

-- DROP FUNCTION "St_SideSegmentation"(character varying, character varying);

CREATE OR REPLACE FUNCTION "St_SideSegmentation"(tabname character varying, gid character varying)
  RETURNS void AS
$BODY$DECLARE
 v int;
BEGIN
Execute
'
insert into Segmentedtable(gid,geom)
SELECT ST_SetSRID(ST_AsText( ST_MakeLine(sp,ep) ),4326)
FROM
   -- extract the endpoints for every 2-point line segment for each linestring
   (SELECT
      ST_PointN(geom, generate_series(1, ST_NPoints(geom)-1)) as sp,
```

Figure5.5.1 Part of code for segmentation of 3D objects

```python
from lxml import etree, objectify
import os
metadata = 'setval.kml'
parser = etree.XMLParser(remove_blank_text=True)
tree = etree.parse(metadata, parser)
root = tree.getroot()

####
for elem in root.getiterator():

    i = elem.tag.find('}')
    if i >= 0:
        elem.tag = elem.tag[i+1:]
objectify.deannotate(root, cleanup_namespaces=True)
####
```

```python
kml_file=r'cityobject.kml'

with open(kml_file) as f:
    doc=parser.parse(f)

#for child in doc.iter():
#       print child.tag

for child in doc.iter():
        if(child.tag=="{http://www.opengis.net/kml/2.2}Polygon"):
                etree.SubElement(child, "{http://www.opengis.net/kml/2.2}extrude")
for child in doc.iter():
        if(child.tag=="{http://www.opengis.net/kml/2.2}Polygon"):
                etree.SubElement(child, "{http://www.opengis.net/kml/2.2}altitudeMode")

root=doc.getroot()


filename = "withextrude.kml"
FILE = open(filename,"w")
FILE.writelines(etree.tostring(root, pretty_print=True))
FILE.close()

tree = ET.parse('withextrude.kml')
root2 = tree.getroot()
for value in root2.getiterator('{http://www.opengis.net/kml/2.2}extrude'):
```

Figure5.5.2 Python modules for parsing the output file and generating the output

2) St_Rotatexyz- This software module is developed to rotate the 3D objects to an angle about x, y ,z coordinate axis. Angles are passed as parameters to the operation and it rotates the 3D object to the given angle in radians. Output of the operation is saved as KML/KMZ format Further python is used to develop software modules for parsing the output KML/KMZ files and customizing them by removing various tag characters, changing the tag structure and values to make it in a format to be viewed on *earth explorer* and web based viewer. Snapshot of the spatial operation a Figure 5.5.3.

```sql
-- Function: "St_Rotatexyz"()

-- DROP FUNCTION "St_Rotatexyz"();

CREATE OR REPLACE FUNCTION "St_Rotatexyz"()
  RETURNS void AS
$BODY$

BEGIN
Execute
'
create table testingexport(id integer, geom geometry(PolygonZ,3068));

INSERT INTO testingexport(id,geom)
SELECT  id,ST_RotateZ(geomtest,pi()/4)
FROM    vaibhav3dnew
where id=var;

UPDATE testingexport
SET geom=(SELECT ST_RotateX(test.geom,pi()/'||val1||')FROM  testingexport test where testingexport.id=test.id)
WHERE id=var2;

UPDATE testingexport
SET geom=(SELECT ST_RotateY(test.geom,pi()/'||val2||')FROM  testingexport test where testingexport.id=test.id)
WHERE id=var3;
```

Figure 5.5.3 Operation for rotation of 3D model about xyz axis

3) St_ExtrudePolygon- This module is developed for dynamically creating a 3D object at LOD2 from a 2D object by giving the height value. This operation is very important because it is alternative to various software tools e.g. *ArcSceane.*, *GoogleSketchup* etc, which are used to create 3D model at LOD2 from the footprint. Output of this spatial operation needs further study to produce better results Snapshot of the spatial operation is shown in Figure 5.5.4.

```
CREATE OR REPLACE FUNCTION public.extrude_polygon3d(wkb_geometry_param geometry, height integer, simplify boolean DEFAULT false)
 RETURNS geometry AS
 $BODY$
DECLARE
f int;
ret_geom geometry;
wkb_geometry geometry;
BEGIN

--convert polygon to linestring
IF ST_GeometryType(wkb_geometry_param) != 'ST_Polygon' THEN
    RETURN NULL;
END IF;

IF simplify THEN
    wkb_geometry = ST_Simplify(ST_Transform(ST_Exteriorring(wkb_geometry_param), 3068), 0.5);
ELSE
wkb_geometry = ST_Transform(ST_Exteriorring(wkb_geometry_param), 3068);
END IF;

--initialise output geometry
ret_geom =ST_MakeLine(ST_PointN(wkb_geometry,1),ST_PointN(wkb_geometry,1));

--Move first point to up

SELECT ST_AddPoint(ret_geom,
      ST_MakePoint(ST_X(ST_PointN(wkb_geometry, 1)),

                    ST_X(ST_PointN(wkb_geometry, 1)),+ height)
) into ret_geom;
```

Figure 5.5.4 Operation for creating a 3D model based on height

New Python Modules developed are:

1)**Python module** for parsing the output KML files as output of spatial queries and customizing them by removing various tag characters, changing the tag structure and values which is shown in Figure 5.5.2.

2)**Python module** for parsing the output of attribute and location based spatial query and creating the actual 3D object which is shown in Figure 5.5.5.

3)**Python module** for creation of 3D object generator which is shown in Figure 5.6.9.

New JavaScript Module developed is:

1) **Java Script** module to view 3D geo-data on web based viewer shown in Figure 5.6.2.

```python
print '---------------------------------------------------------------------------------'

def makepolygon(autolist,floorheight,messagelist):
     autofinallist=[1,2,3,4,5]
     vertlist=[]
     #autofinallist=autolist[:]
     one=autolist[0][:]
     two=autolist[1][:]
     two2=autolist[1][:]
     one2=autolist[0][:]
     one3=autolist[0][:]
     autofinallist[0]=one
     autofinallist[1]=two
     autofinallist[2]=two2
     autofinallist[3]=one2
     autofinallist[4]=one3
     newinterchanged=newpolygoncreation(autofinallist,floorheight,messagelist)
     finalverticalstring=stringmanupulation(newinterchanged)
     vertlist.append(finalverticalstring)
     verticalpolygon(vertlist,messagelist)

def mainfunctionforattributetagging():
    with open(kml_file) as f:
      doc=parser.parse(f)

    root=doc.getroot()

    messagelist=[]
    floorheight=4
    topflag=4
    for child in doc.iter():


            if child.attrib=={'name': 'gid'}:
                messagelist.append(child.text)
            if child.attrib=={'name': 'height'}:
                messagelist.append(child.text)
            if child.attrib=={'name': 'floor'}:
                messagelist.append(child.text)
            if child.attrib=={'name': 'activity'}:
                messagelist.append(child.text)
```

Figure 5.5.5 Python module for attribute based query

5.6 Output of various spatial operations viewed on web based viewer and earth explorer

5.6.1 Output of a spatial query

3D geo-data when stored in GeoRDBMS various attribute and location based queries can be performed on the 3D geo-data. In one of the shown example output of a location based query is shown on earth explores and on web based viewer as well. Semantic information about buildings can also be viewed on web based viewer, which is not in the scope of this study. Figure 5.6.1 shows the output on the earth explorer and Figure 5.6.2 shows the output on web based viewer.

Figure 5.6.1 Output of location based spatial query of selection of building type on earth explorer

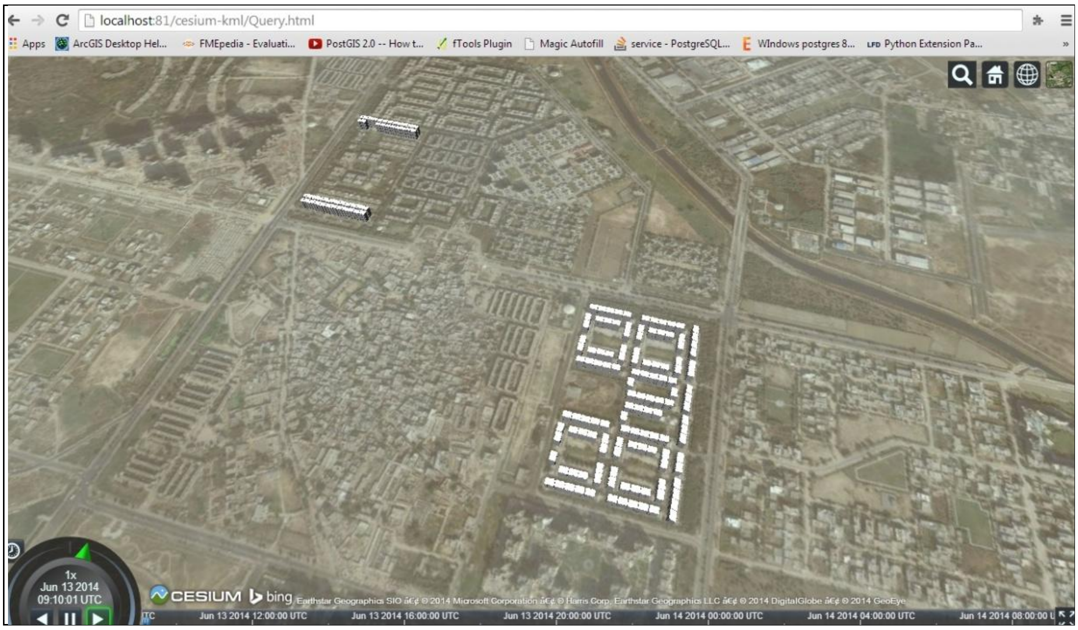


Figure 5.6.2 Output of one the location based spatial query of selection of building type on Web based viewer

5.6.2 Output of 3D segmentation based on spatial query

This is a new development and contribution of this study retrieve side-wise information of a 3D building using segmentation of the building , the application of this module maybe to work on individual sides of a 3D building to find distance between two sides of a building . Developed spatial operation *St_SideSegmentation* is used to segment a 3D object in GeoRDBMS. Developed font end application shown in Figure5.4.4 is used to select the desired 3D geo-data. By executing a query at the back end and 3D segmented output is produced . Output of the executed query is also stored in a different Table so that further queries can be performed on the segmented 3D geo-data. Developed Python modules are used to parse the generated KML file and earth explorer is used to view the output. 3D segmented output of a query is shown below in Figure 5.6.2

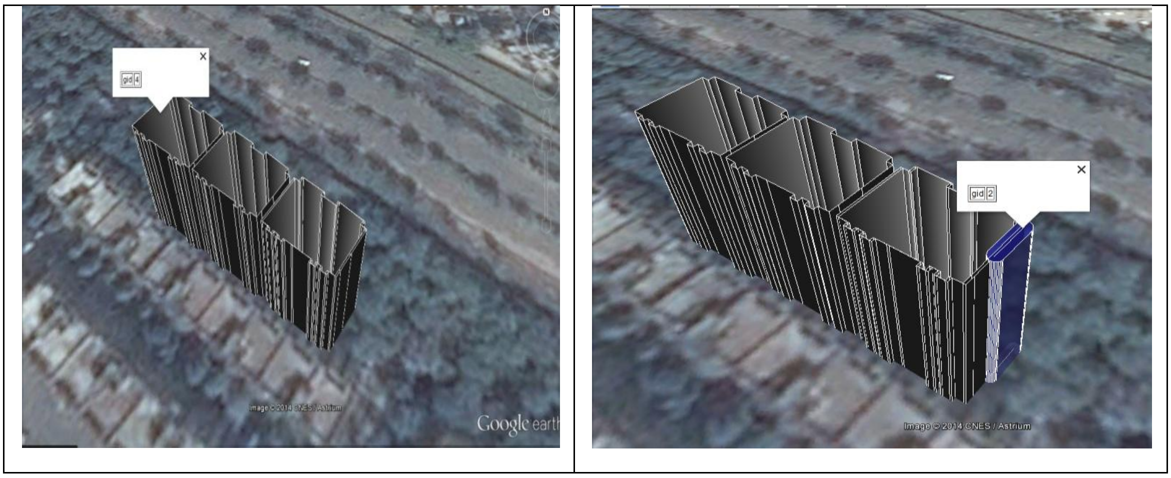


Figure 5.6.2 Segmented output of 3D building using developed spatial query

5.6.3 Location and attribute based spatial query to retrieve floor-wise information in 3D buildings

This is a new development and contribution of this study retrieve floor-wise information of a 3D building using segmentation of the building where each floor is viewed as separate 3D object which has its own semantic information. The application of this module maybe to generate a scenario using spatial and attribute based query on particular floor of a building. In the scenarios like to extract and visualize the exact floors where some activity like fire or any commercial etc is going or floor wise query based on attribute based information can be done by the developed module. Attribute based spatial queries can be performed from the developed front end application as shown in the section 5.4.4. Result from the query execution is KML/KMZ file which is further parsed using developed python module to be viewed on earth explorer and web based 3Dviewer. Query is performed to get the different buildings to get a scenario like to get the floor where fire has been caught, output of the query is 3D buildings and separate 3D objects as floors which has caught file are shown in Figure 5.6.3, Figure 5.6.4 and floors as segmented 3D object is shown in Figure 5.6.5 on earth explorer. Same result is shown on web based viewer in Figure 5.6.6. This implementation also answers the research question *"How can location and attribute based query can be performed for 3D Geo-Data for Topological relationship"* as mentioned in section 1.3.3(b).

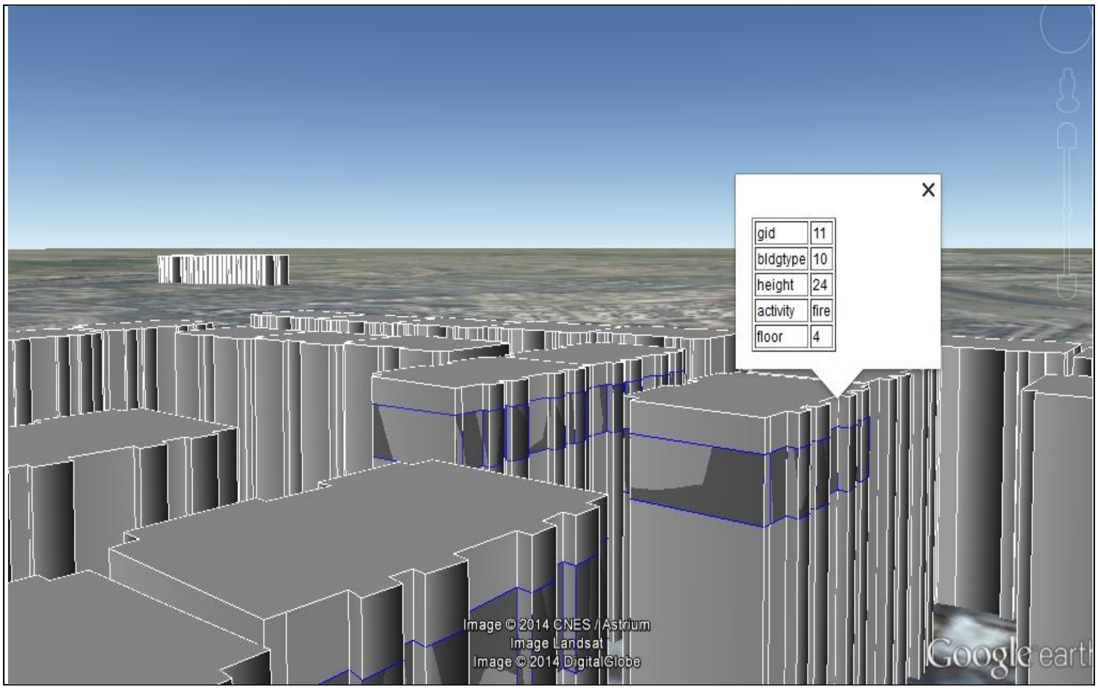


Figure 5.6.3 3D buildings along with the 3D floor wise attribute based query output

As Shown in Figure 5.6.3 Blue colored 3D object is the segmented floor which is displayed as separate 3D object in Figure 5.6.5 and Figure 5.6.6

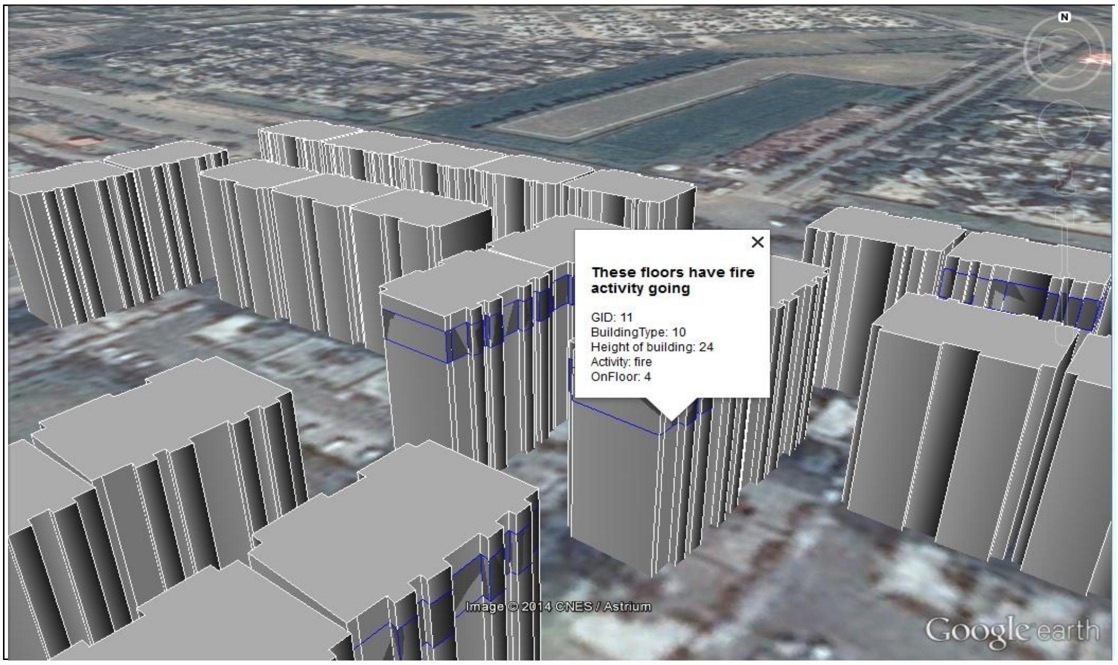


Figure 5.6.4 Fire Information shown on clicking the floor in blue colored 3D object

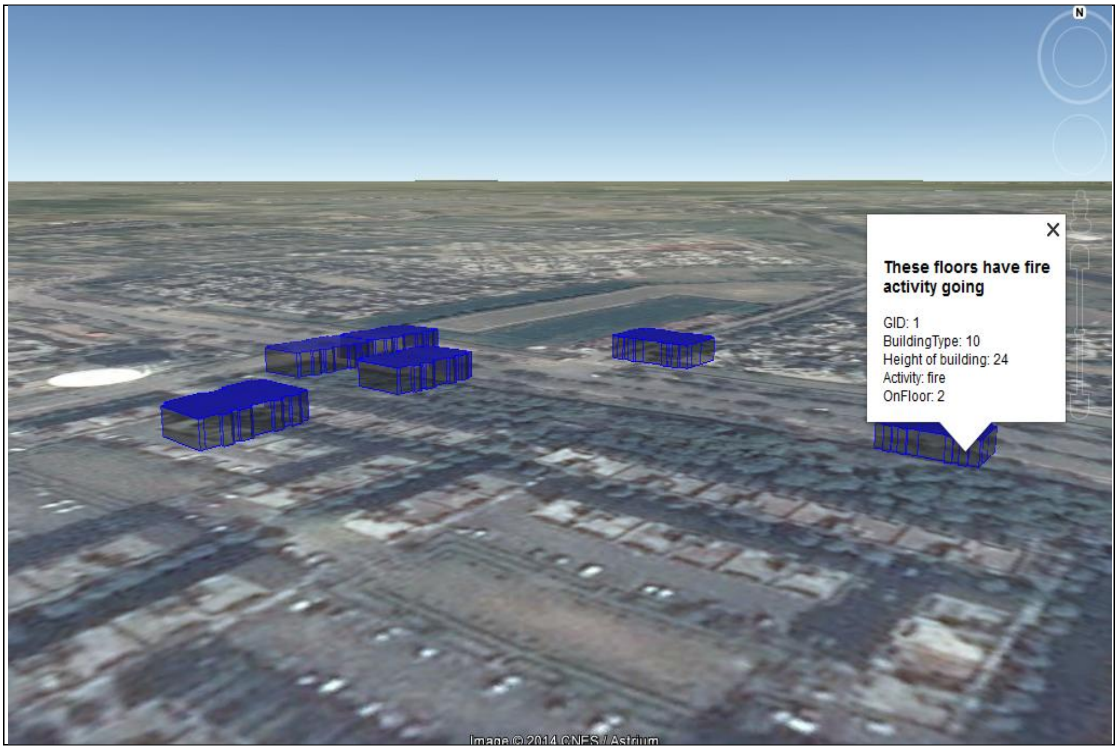


Figure 5.6.5 Different floors output as 3D object having caught fire viewed on earth explorer

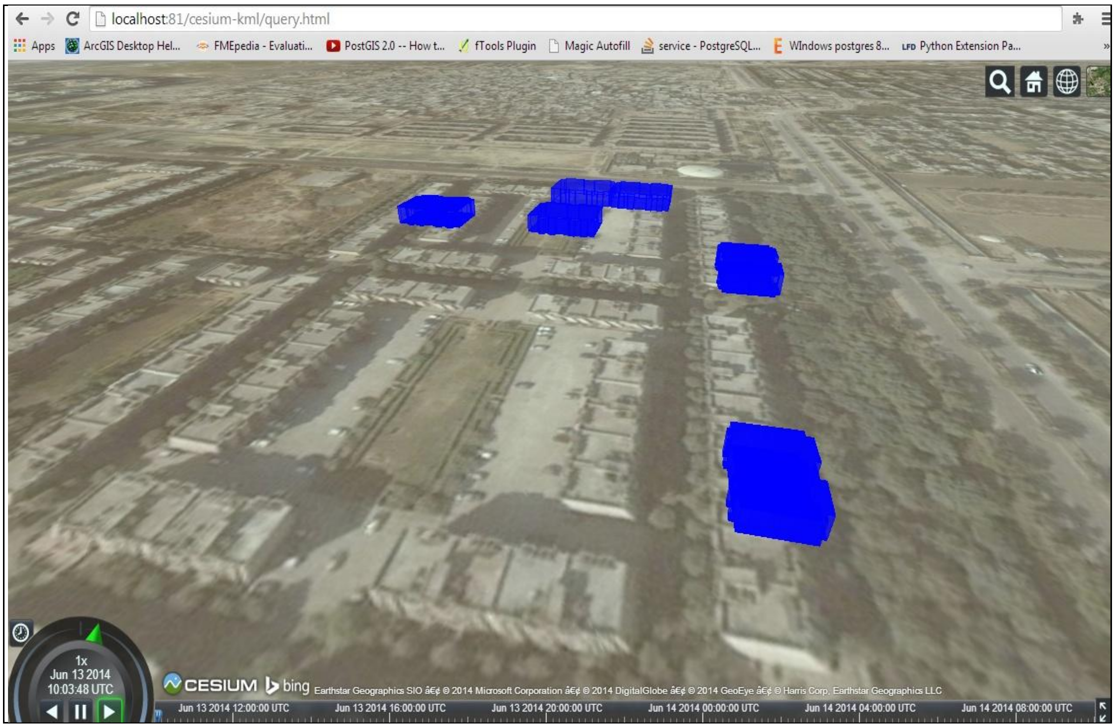


Figure 5.6.6 Different floors output as 3D object having caught fire viewed on web based viewer

5.6.4 Rotation of 3D geo-data in xyz coordinate axis of earth based on spatial query

There are many scenarios where user requires to see the 3D object by positioning it in x, y and z coordinate axis. One such example can be use in urban planning where a user wants to view the best suited position of an proposed construction of a building for better decision making. *This is a new development and contribution of this study wherein developed application as shown in Figure 5.4.4 is used to perform rotation based queries by providing angle of rotation in x, y and z axis.* In the back end developed *St_Rotatexyz* spatial operation generates the 3D output of the query in separate table which is further exported in KML/KMZ format. Developed Python modules are used to perform tag parsing , setting values and other processes to produce the output to be viewed on earth explorer and web based viewer. Figure 5.6.7 and 5.6.8 shows the output of the rotation based query.

(a) (b)

Figure 5.6.7 (a) 3D objects without rotation (b) 3D objects after rotating about X4Z2Y8

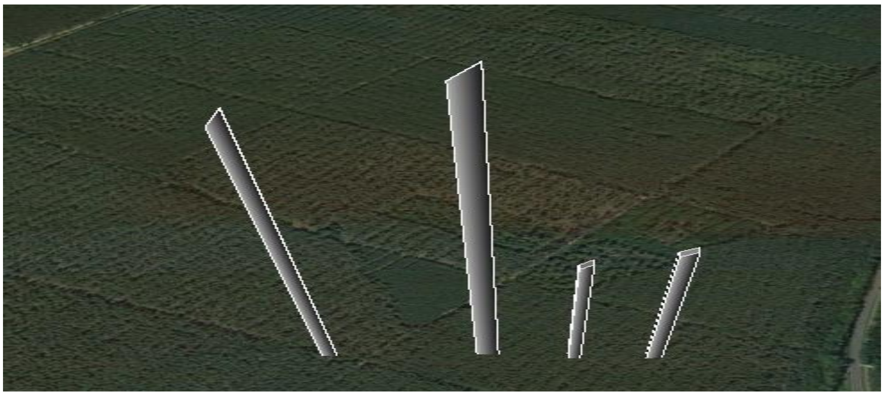


Figure 5.6.8 Shows difference in position after rotation of 3D objects

5.6.5 3D Building Generator

This is a new development and contribution of this study where 3D Buildings of desired height is generated without using any 3D modelling tools like GoogleSketchupandCityEngine. This module can be used to build fast and exciting applications in the field of augmented reality. A developed front end application as shown in Figure 5.6.9 is used to select the footprint as shown in Figure 5.6.10 with no z coordinate user is required to provide the height of the building. The developed module generates the result in KML/KMZ format which can be viewed on earth explorer and web based viewer as shown in Figure5.6.11 and 5.6.12.

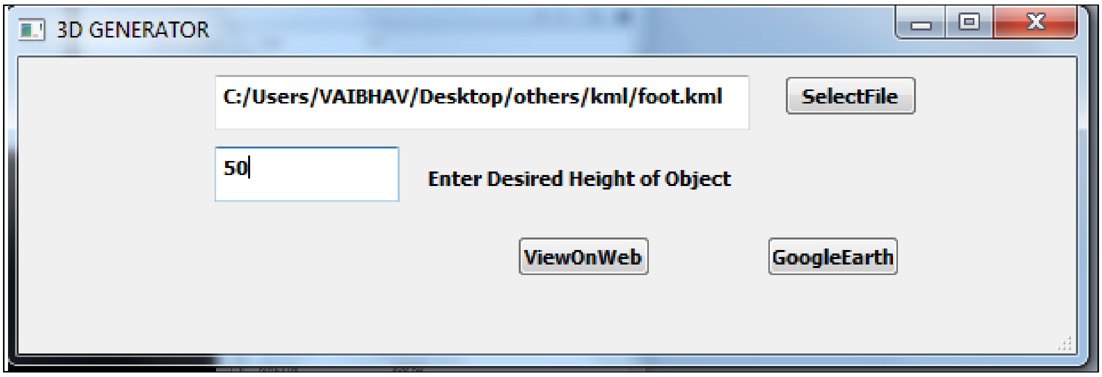


Figure 5.6.9 Automatic 3D Generator

Figure 5.6.10 Footprint of the Buildings

Figure 5.6.11 Generated 3D Building as viewed on earth explorer

Figure 5.6.12 Generated 3D Buildings as viewed on Web

6. Conclusions and Recommendations

6.1 Conclusions

This research has successfully proposed and implemented the study of 3D geo-data organization and management in GeoRDBMS using created 3D geo-data. Methodology has been developed to create 3D geo-data in various file formats KML/KMZ, COLLADA, CityGML. The study of various file formats and 3D geo-data in GeoRDBMS has been done to understand the complex 3D citymodels. The methodology has been developed to visualize the CityGML in the form of KML/KMZ. It enables the 3D buildings and semantics to be visualized on global scale on a virtual globe and also on earth explorer. This study successfully implements attribute tagging with 3D geo-data to perform spatial and non spatial queries in single and multiuser environment. Performance study of GeoRDBMS and query execution is done successfully in single and multiuser environment for developed 3D geo-data in various file formats. New spatial operations like segmentation of 3D buildings, attribute based floor based query and rotation of buildings along with Python and JavaScript software modules are developed. In this study creation of a front end application is completed successfully to perform query in single and multiuser environment and also for optimization of GeoRDBMS. Software modules are written in Python and JavaScript to visualize the 3D geo-data on virtual globe and earth explorer. The development of such functionality can be helpful for decision makers or urban planners in various scenarios.

6.1.1 Answer of research questions

1) What are the challenges for organization and management of 3D geo-data inside GeoRDBMS?

3D geo-data developed in file formats like KML, CityGMLand COLLADA, generate very large and complex amounts of data. GeoRDBMS could provide the framework to define the geometry and topology of complex natural and anthropogenic objects. As per the recent studies GeoRDBMSis unable to provide the exact structure of the geo-data. Geometric primitives like point, line, line-segment, TEN, polygon, polyhedrons are used in modelling complex features, but there is a challenge of handling large amount of data being generated by these primitives. Data types like TEN, polyhedrons have major disadvantages of creation of large amount of data, inability to model curved surfaces properly and lack of spatial operations and high level topology pose hindrance for many complex operations. Topology of 3D data type is one of the important aspects which still needs lot of focus, lack of high level topology leads to many constraints while performing topology based operations. Redundancy of data is also an important challenge while organization of 3D geo-data inside GeoRDBMS. Data types like M*ultiPolygonZ*has problem of redundancy while storing 3D geo-data which leads to increase in the size of geo-data and thus leads to degradation in performance of GeoRDBMS. There is need for development of advanced GeoRDBMS along with advanced techniques like hybrid spatial indexing techniques for better organization and management of 3D geo-data inside GeoRDBMS.

2) How can Location and attribute based query can be performed for 3D geo-data for Topological relationship?

Answer to this research question is discussed in section 5.6.3.

3) Is there any data loss and are all file formats having same data format in GeoRDBMS?

Answer to this research question is discussed in section 5.3.1

4) Do we need to create other data type for storing topology of 3D geo-data like 3D freeform curves?

Answer to this research question is discussed in section 5.3.2.

5) How will the choice of Indexing technique affect the performance of 3D GeoRDBMS, and do we need to create a new indexing technique for optimal performance of GeoRDBMS

Answer to this research question is discussed in section 5.3.1 and section 5.3.2

6) Are open source software systems for GIS sufficient to handle 3D geo-data inside GeoRDBMS?

Concepts of Spatial databases for 2D and 3D geo-data applications is supported by many Spatial Database Management Systems (SDBMS), while true three-dimensional (3D) support for spatial data is a recent addition in Spatial Information Systems (SISs). Database vendors like PostGIS have extended their support to true 3D data types like *Polyhedron* and also store various file formats like KML/KMZ, COLLADA, CityGML.PostGIS also offersvarious 3D operations which are used to query over the stored geo-data inside GeoRDBMS. PostGIS also offers functionality to create own spatial operations which is also an important part of this study. Although open source software systems like PostGIS are sufficient to handle 3D geo-data inside GeoRDBMS, but there is still lot of study is needed to be done in the field of performance enhancement of the GeoRDBMS. The MBB based spatial indexing techniques are difficult to apply on complex data like point cloud. In the recent release of PostGIS there is no native support for point cloud data inside GeoRDBMS. The support for variable page size say 2k,4k,8k,16k etc., needs to be developed in PostGIS as it allows fixed 8k page size for R-Tree indexing due to which there is problem of storing big images, complex geometries in geo-RDBMS. The approaches like combining R-Tree with octree in the geo-RDBMS and other hybrid indexing techniques like V-reactive tree need to be considered.

In addition to this the performance optimization can be done by partitioning of logical tables. In this case the spatial indexes can be performed in parallel. Also the creation of topology structure for 3D objects can reduce the redundancy in the storage of coordinates, which will also enhance the performance of 3D query operation.

6.2 Recommendations

1. CityGML parsing: Current research involves KML parsing for visualization of CityGML contents on a virtual globe. The recommendation is to develop the functionality to parse theCityGML directly for global scale. That will allow the user to directly visualize CityGML contents without any need of pre-processing.
2. Development of new advanced GeoRDBMS: Current research involves vendor like PostGIS as to organize and manage 3D geo-data inside GeoRDBMS . Limitations of PostGIS while handling 3D geo-data is explained in section 6.1.1 point number 6. The recommendation is to develop new advanced GeoRDBMS that allow high performance, more powerful spatial indexing techniques and other optimization parameters. Advanced GeoRDBMS must support new 3D data types that can reduce the complexity size of stored the 3D geo-data.
3. Spatial and Topological operations: In the current scenario lack of spatial operations and high level topology pose hindrance for many complex operations like finding amount of merging area or total area formed by combining different types of features etc. The recommendation is to develop new spatial and topological operations and to develop high level topology structure of 3D data type inside GeoRDBMS so that it can be incorporated with augmented reality scenarios.
4. 3D modelling tools: Current research involves creation of 3D model of the study area, This 3D model is created using tools like *CityEngine* and *GoogleSketchup*.The recommendation is to develop new open source tools which can be easily used in 3D modelling.

7. References

Altmaier, Angela, and Thomas H. Kolbe. 2003. "Applications and Solutions for Interoperable 3D Geo-visualization." In *Proceedings of the Photogrammetric Week*, 251–67. http://www.ikg.uni-bonn.de/fileadmin/publication/media/Altmaier_und_Kolbe_PhoWo2003.pdf.

Arens, C.W. Quak C.A., and P.J.M. van Oosterom J.E. Stoter. 2003. "MAINTAINING REALITY." *Department of Geodesy Faculty of Civil Engineering and Geosciences*. Accessed July 13. http://repository.tudelft.nl/assets/uuid:fe3d7255-4adb-4103-99f3-a6353cf350dd/ceg_arens_2003.pdf.

Arens, Călin, Jantien Stoter, and Peter van Oosterom. 2005. "Modelling 3D Spatial Objects in a geo-DBMS Using a 3D Primitive." *Computers & Geosciences* 31 (2): 165–77. doi:10.1016/j.cageo.2004.05.013.

Breunig, Martin, and Sisi Zlatanova. 2011a. "3D geo-data base Research: Retrospective and Future Directions." *Computers & Geosciences* 37 (7): 791–803. doi:10.1016/j.cageo.2010.04.016.

———. 2011b. "3D geo-data base Research: Retrospective and Future Directions." *Computers & Geosciences* 37 (7): 791–803. doi:10.1016/j.cageo.2010.04.016.

Chen, T. K., A. Abdul-Rahmana, and S. Zlatanova. 2008a. "3D Spatial Operations for geo-DBMS: Geometry Vs. Topology." In *International Archives of XXIth Congress of the ISPRS*. http://www.isprs.org/proceedings/XXXVII/congress/2_pdf/4_WG-II-4/08.pdf.

———. 2008b. "3D Spatial Operations for geo-DBMS: Geometry Vs. Topology." In *International Archives of XXIth Congress of the ISPRS*. http://www.isprs.org/proceedings/XXXVII/congress/2_pdf/4_WG-II-4/08.pdf.

Coumans, Erwin, and Keith Victor. 2007. "COLLADA Physics." In *Web3D*, 101–4. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.93.5429&rep=rep1&type=pdf.

De Cambray, Béatrix. 1993. "Three-dimensional (3D) Modelling in a Geographical Database." In *AUTOCARTO-CONFERENCE-*, 338–338. Citeseer. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.52.2698&rep=rep1&type=pdf.

Dieter Schmidt, Dieter Fritsch. 1996. "In Transition from 2.5D GIS to 3D." http://www.isprs.org/proceedings/xxxi/congress/part4/748_XXXI-part4.pdf.

Gruen, Xinhua Wang Armin, and X. Wang. 1998. "A 3-D City Model Data Structure and Its Implementation in a Relational Database." In *Proceedings of the International Conference of Spatial Information Science and Technology, Wuhan China*, 429–36. http://www.researchgate.net/publication/228428662_A_3-D_city_model_data_structure_and_its_implementation_in_a_relational_database/file/60b7d528dcf9036d64.pdf.

Hellerstein, Joseph M., Jeffrey F. Naughton, and Avi Pfeffer. 1995. *Generalized Search Trees for Database Systems*. September. http://books.google.com/books?hl=en&lr=&id=7a48qSMuVcUC&oi=fnd&pg=PA467&dq=%22that+allow+it+to+behave+like+a+B%2B-tree,+an+R-tree,%22+%22ef%EF%AC%81cient+implementation+of+search+trees+is+crucial+for%22+%22tree+technology.+We+introduce+a+new+data%22+&ots=taha1uPo1i&sig=UTSUUwwMKiKxPCcyxsgWem9Kdec.

Höfle, BERNHARD, Martin Rutzinger, Thomas Geist, and Johann Stötter. 2006. "Using Airborne Laser Scanning Data in Urban Data Management-set up of a Flexible Information System with Open Source Components." In *Proceedings of UDMS*, 2006:25th. http://wl27www658.webland.ch/bh/docs/papers/2006/Hoefle_et_al_UDMS_2006.pdf.

Kaasalainen, Sanna, Anssi Krooks, Antero Kukko, and Harri Kaartinen. 2009. "Radiometric Calibration of Terrestrial Laser Scanners with External Reference Targets." *Remote Sensing* 1 (3): 144–58. doi:10.3390/rs1030144.

Karnatak, Harish Chandra, Sameer Saran, Karamjit Bhatia, and P. S. Roy. 2007. "Multicriteria Spatial Decision Analysis in Web GIS Environment." *GeoInformatica* 11 (4): 407–29. doi:10.1007/s10707-006-0014-8.

Khuan, Chen Tet, Alias Abdul-Rahman, and Sisi Zlatanova. 2007. "New 3D Data Type and Topological Operations for geo-DBMS [M]." *Urban and Regional Data Management: UDMS Annual*. http://books.google.com/books?hl=en&lr=&id=BRDKoVdsPF4C&oi=fnd&pg=PA211&dq=%22polyhedron+could+be+defined+as+a+bounded+subset+of+3D+space+enclosed+by+a+finite+set+of%22+%22and+topology+model+(Penninga+et+al,%22+%223D+CityGML+(Kolbe+et+al.+2006).+It+is+an+open+data+model+and+XML-based%22+&ots=9WXEf-sP34&sig=pccLgQ_i75V9C7uAuNBv6ajDlRU.

Kolbe, Thomas H. 2009. "Representing and Exchanging 3D City Models with CityGML." In *3D Geo-information Sciences*, 15–31. Springer. http://link.springer.com/chapter/10.1007/978-3-540-87395-2_2.

Kolbe, Thomas H., Claus Nagel, and Javier Herreruela. 2013. "3D City Database for CityGML." *Addendum to the 3D City Database Documentation Version* 2 (1). http://opportunity.bv.tu-berlin.de/svn/3dcitydb-imp-exp/tags/1.5.0-postgis/resources/manual/3DCityDB-Documentation-Addendum-v2_0_6.pdf.

Muki Haklay, Claire Ellula. 2006. "Beyond_9-intersections_-_Modifications_to_a_B-Rep_Topological_Structure_to_Identify_Additional_Relationships_in_3D.pdf". Dept. of Geomatic Engineering, University College London,. http://www.ge.ucl.ac.uk/.

Pescarin, Sofia, Maurizio Forte, Luigi Calori, Carlo Camporesi, Antontella GUIDAZZOLI, and Silvano Imboden. 2005. "Open Heritage: An Open Source Approach to 3d Real-time and Web-based Landscape Reconstruction." In *Proceedings of XI International Conference on Virtual Systems and Multimedia: Virtual Reality at Work in the 21st Century, Ghent Belgium*, 313–20. http://www.cineca.it/sites/default/files/paperpescarinfortecalori.pdf.

"Postgis-2.0.pdf."

Raptopoulou, Katerina, Michael Vassilakopoulos, and Yannis Manolopoulos. 2004. "Towards Quadtree-based Moving Objects Databases." In *Advances in Databases and Information Systems*, 230–45. http://link.springer.com/chapter/10.1007/978-3-540-30204-9_16.

Remondino, F., S. El-Hakim, S. Girardi, A. Rizzi, S. Benedetti, and L. Gonzo. 2009. "3D Virtual Reconstruction and Visualization of Complex Architectures-The' 3D-ARCH' Project." In *Proceedings of the ISPRS Working Group V/4 Workshop 3D-ARCH" Virtual Reconstruction and Visualization of Complex Architectures", 2009.* http://nparc.cisti-icist.nrc-cnrc.gc.ca/npsi/ctrl?action=rtdoc&an=15188886.

Remondino, Fabio, and Sabry El-Hakim. 2006. "Image-based 3D Modelling: A Review." *The Photogrammetric Record* 21 (115): 269–91.

Rizwan Khan, Hardik Panchal, and N.L. Sarda Smita Sengupta. 2011. "GIS-based Smart Campus System Using 3D Modeling." In *GIS-based Smart Campus System Using 3D Modeling*. Hyderabad,India.

Sardadi, Maruto Masserie, M. Rahim, Zahabidin Jupri, and D. Baman. 2008a. "Choosing R-tree or Quadtree Spatial Data Indexing in One Oracle Spatial Database System to Make Faster Showing Geographical Map in Mobile Geographical Information System Technology." *World Academy of Science Engineering and Technology* 46: 249–57.

———. 2008b. "Choosing R-tree or Quadtree Spatial Data Indexing in One Oracle Spatial Database System to Make Faster Showing Geographical Map in Mobile Geographical

Information System Technology." *World Academy of Science Engineering and Technology* 46: 249–57.

Schön, Bianca, Michela Bertolotto, Debra F. Laefer, and Sean Morrish. 2009. "Storage, Manipulation, and Visualization of LiDAR Data." In *Fabio Remondino, Sabry El-Hakim, Lorenzo Gonzo (eds.). Proceedings of the 3rd ISPRS International Workshop 3D-ARCH 2009:" 3D Virtual Reconstruction and Visualization of Complex Architectures" Trento, Italy, 25-28 February 2009: International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences: Volume XXXVIII-5/W1*. http://researchrepository.ucd.ie/handle/10197/2380.

Siyka Zlatanova, Roland Billen. 2003. "3D Spatial Relationships Model: a Useful Concept for 3D Cadastre?" *Computers Environment and Urban Systems* 27: 411–25.

Stoter, Jantien, and Siyka Zlatanova. 2003a. "Visualisation and Editing of 3D Objects Organised in a DBMS." In *Proceedings of the EuroSDR Com V. Workshop on Visualisation and Rendering*, 22–24. http://www.gdmc.nl/zlatanova/thesis/html/refer/ps/st_zl03.pdf.

———. 2003b. "3D GIS, Where Are We Standing." In *Proceedings ISPRS Workshop on Spatial, Temporal and Multi-dimensional Data Modelling and Analysis, Quebec, Canada*. Citeseer. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.106.5529&rep=rep1&type=pdf.

———. 2003c. "Visualisation and Editing of 3D Objects Organised in a DBMS." In *Proceedings of the EuroSDR Com V. Workshop on Visualisation and Rendering*, 22–24. http://www.gdmc.nl/zlatanova/thesis/html/refer/ps/st_zl03.pdf.

Tavanti, Monica, and Mats Lind. 2001. "2D Vs 3D, Implications on Spatial Memory." In *Information Visualization, 2001. INFOVIS 2001. IEEE Symposium On*, 139–45. IEEE. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=963291.

Van Oosterom, Peter, Jantien Stoter, Wilko Quak, and Sisi Zlatanova. 2002. "The Balance Between Geometry and Topology." In *Advances* in *Spatial Data Handling*, 209–24. Springer. http://link.springer.com/chapter/10.1007/978-3-642-56094-1_16.

Vanichayobon, Sirirut, and Le Gruenwald. 1999. "Indexing Techniques for Data Warehouses' Queries." *The University of Oklahoma, School of Computer Science, Technical Report*. http://www.cs.ou.edu/~database/documents/vg99.pdf.

Yang, Ming-Der, Chih-Fan Chao, Kai-Siang Huang, Liang-You Lu, and Yi-Ping Chen. 2013. "Image-based 3D Scene Reconstruction and Exploration in Augmented Reality." *Automation in Construction* 33 (August): 48–60. doi:10.1016/j.autcon.2012.09.017.