2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT)
Bangalore, India. Oct 7-9, 2022

An Application for Road Network Data Management and Querying

Sudha Chaturvedi
Fair Isaac Corporation(FICO),
Bengaluru, Karnataka, India 560017
sudhachaturvedi18@gmail.com

Dr. Tapsi Nagpal
Department of Computer Science and Information Technology,
Lingayas University, India 121002
dr.tapsi@lingayasvidyapeeth.edu.in

Abstract **- Information Communication Technology based location services and data management technology has huge impact on route prediction and management. This work deals with spatial road network data management effectively for developing applications for querying and visualization of road networks. Underlying data is huge in size and its spatial attributes brings complexity in storage. A detailed technique is presented to solve storage complexity. Application architecture is presented which leverages spatial and non-spatial attributes of road network data for use in querying and analyzing the road network data. Proposed application enables users for route planning and can also be useful for civic authorities like municipality, security agencies for querying Road Network Databases for resource management and Planning. Framework is built using open source technologies for data storage and visualization.**

Index Terms – Spatial Data, Road Network, OSM, PostGis.

I. Introduction

Road network data has wide use in location base service like route planning, route prediction, resource management, traffic congestion prediction, smart city planning etc. This data is huge in size and involves handling of both spatial and non-spatial attributes makes data management complex [1, 14]. This Paper is presents an application whose aim is to cater the need for application which can assist in trip planning and road resource management. In simple scenarios, users can use the application for normal queries like finding shortest path, route planning etc. The civic organizations can use the application to query road network databases to fetch related information for planning and resource management. The application uses the spatial enabled databases to store the spatial features related to road like centre line of roads, road Junctions etc and also non spatial data related to road like width of road, turn restrictions, name of the road etc. [15]. The Road network is stored in the form of graph network so that queries to road network database can be done and some special queries like shortest path can be executed. Proposed application provides the web based interface to user for querying the road network database. The result of the queries can be non-spatial like "*Find the width of Road NH-4*" as well as it can be spatial also like "*Show the route between Mumbai to Delhi"*. The application has the capability to handle both kinds of these queries. This paper is the proof of concept of an application which can be used for route planning and resource management.

Data source for road network is the graph representation of road network. Schema representation of road network graph contains edges of road network which is 2-dimensional multi-line geometries in nature [2, 16]. Another data set is the set of road network vertices representing road junctions or important locations on road network edges [17, 18]. Popular commercial vendors like Google maps provides the interface which exposes the data via web and set of APIs which like location search, paths from one location to another etc. But applications which are meant to implement customized queries require the complete data to be sourced into appropriate storage [1, 14]. Alternative is Open Street Map (OSM) which is open source and provides facilities like location search, direction search. Additionally data can also be sourced and stored in spatial databases on which custom queries can be executed.

Data is sourced from Open Street Map(OSM) using web interface of OSM or alternatively from servers of OSM. For experimentation purpose data is sourced from Open Street Map. Data cleaning is applied using custom utility and described in section II. Processed data is imported into spatial layer enabled PostGis database using data processing utility Osm2pgsql. Once data is available in database custom algorithms like shortest path, customized queries and be applied. Architecture and step of application is as discussed in section III. Data visualization for output of query results is a required module for analysis. A customized web based interface is implemented which integrates open layer in web page using Java Script. Open layer implementation used is Geoserver which is open source and provides interfacing to spatial data sources like in this case tables having spatial data in PostGis. Geoserver renders open layers placed over each other. In this implementation two open layers are defined. First layer is image representation of geographic reason [5]. This can be done by using any suitable source like Google maps or as in this case digitized image sourced online from Open Street Map (OSM). Second layer is road network data sourced from PostGis database and is super imposed over first layer of image representation geographic region.

II. Road network data Management

A. Road Network Data

A road network is a directed graph G(V, E), where V is a set of vertices representing road intersections and terminal points, and E is a set of edges representing road segments each connecting two vertices [3, 4, 19]. Set of Edges (E) is 2-

2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT) | 978-1-6654-6855-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/GCAT55367.2022.9972076

dimensional multi-line geometries and set of vertices (V) road network vertices representing road junctions or important locations on road network edges [20]. Sample road network is as shown in Fig. 1.

- *Nodes:* contains point features – road junctions, important places, vertices of the polygons (of water resources, buildings etc.)

- *Lines:* contains arcs representing the centre lines of the roads.

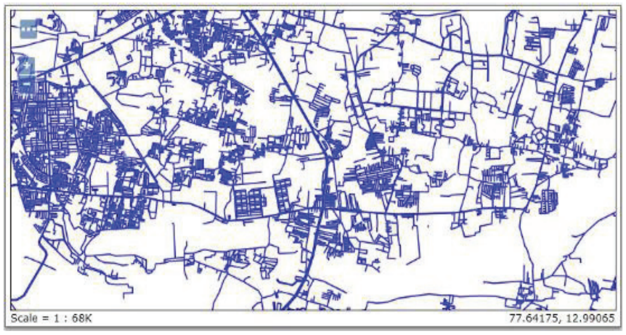


Fig 1. Road Network

B. Road Network Data Source

Road network can be sourced from commercial vendors but they are of limited usage. For example Road network data is available with Good Maps but the data is tiled images representing the geographical entities like road networks, polygons representing buildings, water bodies etc. [22]. On top of it APIs are available for example location search, path search, national and international political boundaries [9, 11]. But limitation is you cannot source the data in database to build application for implementation of customized queries for research and development.

Alternative is to source the data from Open Street Map (OSM) where data is available under open source license. Open Street maps creates and provides free geographic data such as street maps and is published under an open content license, with the intention of promoting free use and re-distribution of the data (both commercial and non commercial) (en.wikipedia.org) [2, 12]. Data can be downloaded from OSM interface at http://www.openstreetmap.org/. Snapshot of OSM interface is as shown in Fig. 2.

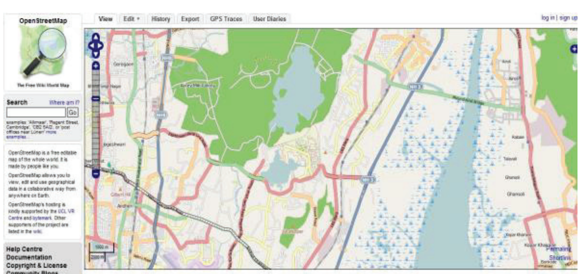


Fig. 2 Interface of Open Street Maps

From this interface an area can be zoomed in and can be selected to export to the local client as image file (GIF, PNG or JPEG) or as .osm file in XML format. If the selected area is small then it works. For larger area files, ww.cloudmade.org is the official site through which the OSM data (Centerline of Roads, Polygons of Buildings, State and international boundaries as lines, Water bodies polygons etc.,) can be downloaded. The data is available at various levels. Data of whole world, country wise, state wise are available to download [4, 10].

C. Road Network Data Cleaning

Road network data sourced from any source is digitized form of geometries representing the centre lines of the roads and coded 2-dimensional arcs along with nodes representing the intersection of the roads [18, 19, 20]. There can be some inaccuracies in digitization process. Data quality plays vital role in accuracy of the query results [5, 6]. Major of the data quality issues and their resolution is as discussed below.

- Removal of undershoots: Find all segments whose endpoints do not coincide with the endpoints of some segment. For each such segment, find all vertices of any segment within a tolerance of the unconnected endpoint. Snap the unconnected endpoint to the nearest such vertex. Example of undershoot is as in Fig. 3.

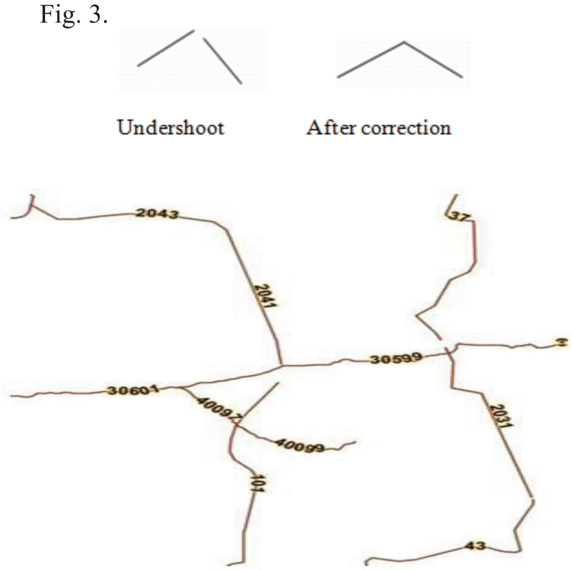
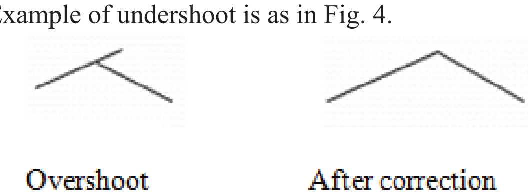


Fig. 3 Example of undershoot

- Removal of Overshoots: Find intersections between all pairs of segments. Many possibilities can be eliminated by first comparing the bounding rectangles of the segments. At each intersection, split each of the intersecting segments into two segments. Example of undershoot is as in Fig. 4.



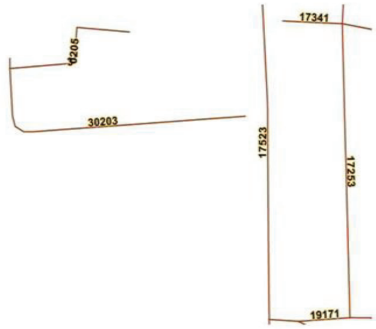


Fig. 4 Example of overshoot

➢ Some places may seem located off side the roads and they seem to be stray nodes but actual they may be actually on the road. These should be corrected. An example of Important places but not lying on the road is as in Fig. 5.

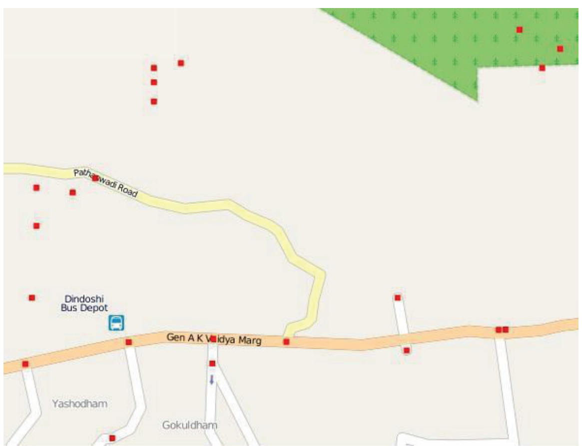


Fig. 5 Important places but not lying on the road

➢ All the important places like hospital, schools etc., may not be located at only road intersections. But nodes must be placed on the road network and the arcs must be split at these positions as well. An example of Important places lying actually on road but not in digitized map is as in Fig. 6.

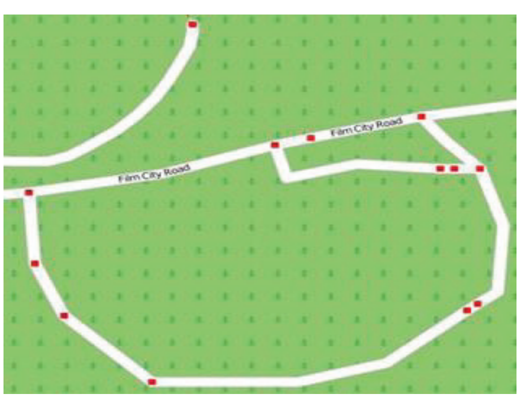


Fig. 6 Important places lying actually on road but not in digitized map

There may not be possible to transit from one arc to another arc even if there may be an intersection. For example- at flyovers, at the intersection of road and rail networks. Fig. 7 represents road intersection but there is no transition possible

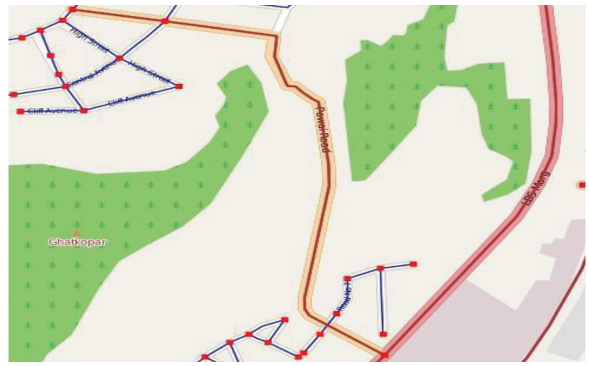


Fig. 7 Road intersection but no transition possible

➢ For each segment all the nodes should found (junctions and places) which lie on this segment. The segment should be then divided into sub-segments between each pair of these vertices. Fig. 8 shows arcs with densely populated vertices.

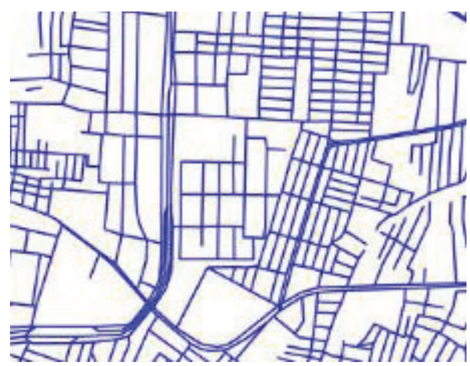


Fig. 8 Edges densely populated with vertices

D. Road Network Data Storage

The roads may be viewed as collection of spatial objects and relationships of these objects are needed to be established [6, 21]. Network and Topological models are two common representations of spatial objects and hence for roads in vector form. They mainly differ in the expression of topological relationships among the component objects. Topological relationships among spatial objects are those relations that are invariant under topological transformations. In other words, they are preserved when the spatial objects are, for instance, translated, rotated, or scaled in the Euclidean plane. Topological relationships include adjacency, overlapping, disjointness, and inclusion, and constitute therefore an important class of spatial relationships. The explicit representation of such relationships in the spatial data model provides more knowledge and is helpful during the evaluation of queries [7, 8].

The road network files from OSM contains two files first, contains the road network edges with edge geometry and other non spatial attributes like name of the road, length of the attributes, vertex number representing the two end points of the edge and important location like mall, schools etc. [12, 13]. Second data file contains the vertices representing end points of arcs and has spatial attributes representing the 1-demesional point object represented by latitude and longitude pair. Sample data file is as represented in Fig. 9.


```xml
<?xml version="1.0" encoding="UTF-8"?>
- <osm license="http://opendatacommons.org/licenses/odbl/1-0/" attribution="http://www.openstreetmap.org/copyright" copyright="OpenStreetMap and contributors"
generator="CGImap 0.6.0 (23926 thorn-02.openstreetmap.org)" version="0.6">
    <bounds maxlon="72.9002400" maxlat="19.0856300" minlon="72.8947900" minlat="19.0829200"/>
    <node version="1" lon="72.8919209" lat="19.0813645" uid="1306" user="PlaneMad" timestamp="2011-07-18T21:16:46Z" changeset="8762480" visible="true"
       id="1366096061"/>
    <node version="2" lon="72.8898527" lat="19.0818274" uid="1306" user="PlaneMad" timestamp="2016-06-25T18:28:59Z" changeset="40286730" visible="true"
       id="1366096063"/>
    <node version="1" lon="72.8914596" lat="19.0825633" uid="1306" user="PlaneMad" timestamp="2011-07-18T21:16:46Z" changeset="8762480" visible="true"
       id="1366096082"/>
    <node version="1" lon="72.8982092" lat="19.0833624" uid="1306" user="PlaneMad" timestamp="2011-07-18T21:16:46Z" changeset="8762480" visible="true"
       id="1366096086"/>
    <node version="1" lon="72.8980017" lat="19.0840744" uid="1306" user="PlaneMad" timestamp="2011-07-18T21:16:47Z" changeset="8762480" visible="true"
       id="1366096107"/>
    <node version="2" lon="72.8904256" lat="19.0843989" uid="1306" user="PlaneMad" timestamp="2016-06-25T18:28:59Z" changeset="40286730" visible="true"
       id="1366096112"/>
    <node version="2" lon="72.8994362" lat="19.0844052" uid="1306" user="PlaneMad" timestamp="2016-06-25T18:28:59Z" changeset="40286730" visible="true"
       id="1366096120"/>
    <node version="2" lon="72.8917734" lat="19.0845535" uid="1306" user="PlaneMad" timestamp="2016-06-25T18:28:59Z" changeset="40286730" visible="true"
       id="1366096125"/>
    <node version="2" lon="72.8940917" lat="19.0853955" uid="1306" user="PlaneMad" timestamp="2016-06-25T18:28:59Z" changeset="40286730" visible="true"
       id="1366096142"/>
    <node version="1" lon="72.8988012" lat="19.0866026" uid="1306" user="PlaneMad" timestamp="2011-07-18T21:16:47Z" changeset="8762480" visible="true"
       id="1366096158"/>
    <node version="1" lon="72.8983948" lat="19.0836654" uid="1293487" user="Giyavudeen" timestamp="2013-04-02T14:29:09Z" changeset="15582315" visible="true"
       id="2244961382"/>
    <node version="1" lon="72.8995798" lat="19.0840778" uid="1293487" user="Giyavudeen" timestamp="2013-04-02T14:29:09Z" changeset="15582315" visible="true"
       id="2244961385"/>
```


Fig. 9 Road network data file

Data file is converted to SQL statement file using utility known as osm2pgsql. Utility takes as input road network files and converts it into PostGis compatible SQL files which can be directly imported to database table. Utility encodes the multiline geometries for edges and point objects for vertices in binary encoded format which is compatible with geometry data type of PostGis database. Data stored in PostGis database table is as shown in fig 10. Schema for storing the road network in database is shown in Fig. 11.

File Edit View Tools Help
100 rows

	id [PK] integer	osm_id bigint	osm_name character var	osm_meta character var	osm_source_ bigint	osm_target_ bigint	clazz integer	flags integer	source integer	target integer	km double precis	kmh integer	cost double precis	reverse_cost double precis	x1 double precis	y1 double precis	x2 do
1	1	4354935			3597890513	431610693	51	3	787	30883	0.6710551	5	0.13421102	0.13421102	77.638871	12.9902453	77
2	2	4354935			431610693	431610695	51	3	30883	788	0.20598954	5	0.041197907	0.041197907	77.6328052	12.9891204	77
3	3	4354953	Mahatma Gan		246397923	3343265828	21	1	789	58770	0.03760756	30	0.001253585	1000000	77.5992534	12.9767243	77
4	4	4354953	Mahatma Gan		3343265828	3400262500	21	1	58770	59177	0.07492242	30	0.002497414	1000000	77.5996003	12.9767135	77
5	5	4354953	Mahatma Gan		3400262500	334484150	21	1	59177	790	0.15859316	30	0.005286439	1000000	77.6002914	12.976692	77
6	6	8294886	Kumara Krup		3676198939	3676198953	31	3	791	60324	0.038530476	40	0.000963261	1000000	77.5779982	12.9838341	77
7	7	8294886	Kumara Krup		3676198953	1614268206	31	3	60324	28159	0.01493233	40	0.000373308	1000000	77.5782293	12.9840925	77
8	8	8294886	Kumara Krup		1614268206	1614268208	31	3	28159	792	0.032841165	40	0.000821029	1000000	77.5783036	12.9842056	77
9	9	8294933	3rd Main Ro		17327092	699585165	31	3	793	864	0.009320113	40	0.000233002	0.000233002	77.5950592	12.9063347	77

Fig. 10 Road network data file stored in PostGis

```sql
CREATE TABLE bengaluru_2po_4pgr
(
  id integer NOT NULL,
  osm_id bigint,
  osm_name character varying,
  osm_meta character varying,
  osm_source_id bigint,
  osm_target_id bigint,
  clazz integer,
  flags integer,
  source integer,
  target integer,
  km double precision,
  kmh integer,
  cost double precision,
  reverse_cost double precision,
  x1 double precision,
  y1 double precision,
  x2 double precision,
  y2 double precision,
  geom_way geometry(LineString,4326),
  CONSTRAINT pkey_bengaluru_2po_4pgr PRIMARY KEY (id)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE bengaluru_2po_4pgr
  OWNER TO postgres;
```

Fig. 11 Schema for road network data storage in PostGis

III. ROAD NETWORK DATA QUERY APPLICATION

A. Application Architecture

Application uses road network data stored in PostGis database as described in section II. Application provides web based interface to run the customized user queries. In backend an execution engine is implemented in Java which accepts user queries from web service and executes in database query engine and stores the result in a result table. Result table has the geometrical column to store spatial attributes and also non-spatial attributes as well to store non-geometrical data produced as part of the query. Architecture of the application is as described in Fig. 12.

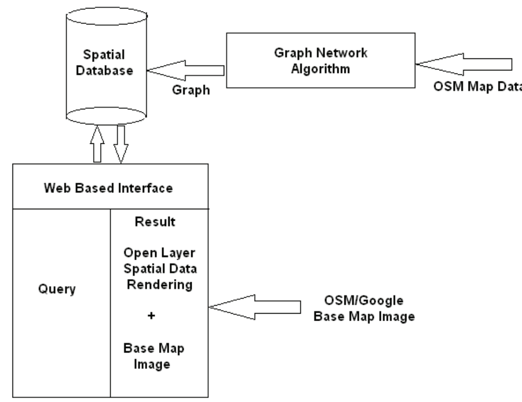


Fig. 12 Architecture of Application

From resultant table Geoserver acts as Web Map Service (WMS) which prepares the open layers and provides as input to web interface. Application uses two layers and discussed in next sections:

- Open layers (maps formed by fetching data from Data Base) and
- Base map images (Google Map Images or OSM Map Images can be used)

B. Road Network Open Layer

Open Layers are rendered using Geoserver. Geoserver is again open source software. It uses Web Map Service (WMS) standard to connect to Spatial DB (in this case PostGis) and render the spatial features stored in databases. Geoserver is works through HTTP service and can be easily embedded in Web Pages [2]. Open layer rending the road network query result table from PostGis via GeoServer is as shown in Fig. 13.

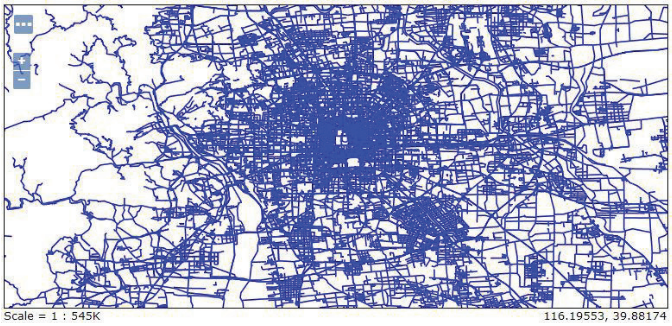


Fig. 13 Road network open layer

C. Base Map Layer

Base map layer sits behind the open layer for meaningful visualization of the result of the query. For base map layer Geoserver is not used but instead application's front end makes a directly call the service provider by invoking java script API to pull the map as the tiled image. Open layer is super imposed on top of the base map layer. Base map layer provided by Google and OSM is shown in Fig. 14 and 15 respectively. Example of road network open layer imposed over OSM base map layer is as shown in Fig. 16.

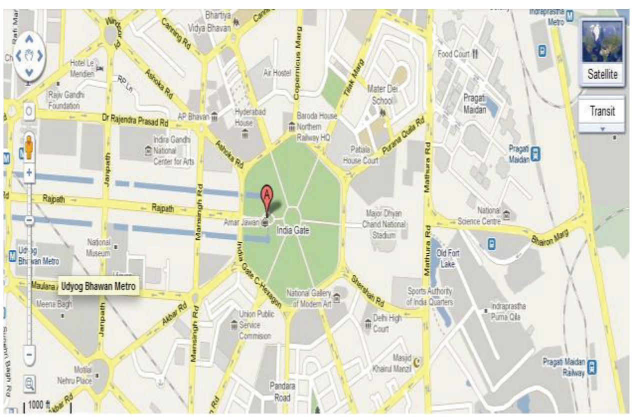


Fig. 14 Base map layer provided by Google Maps

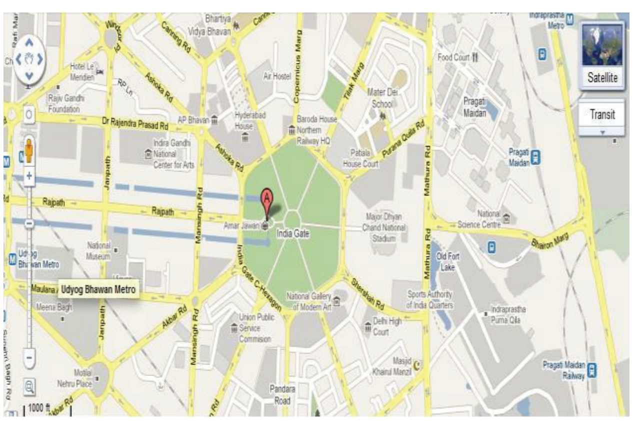


Fig. 15 Base map layer provided by OSM

Fig. 16 Open layer imposed over OSM base map

D. Query Engine Flow and Examples

Data flow for query engine is as in sequence described below:

1) User provides the customized query via web based interface.
2) Web frontend makes a call to backend web service implemented in Java.
3) Backend web service sends query to execution engine of database.
4) Database SQL engine executes the query and stores the result in PostGis table.
5) On completion of query execution, web service notifies the frontend GUI.
6) GUI makes call to Google map or OSM to obtain base image followed by calls Geoserver to prepare the open layer containing the data from resultant table.
7) Base map image is rendered followed by on top of this layer, open layer is rendered.

Example queries and resultant is as below.

Query 1:

"*Show all important places lying in input minimum bounding rectangle (MBR)".*

Result is as shown in Fig. 17.

Fig. 17 All important places lying in MBR

Query 2:

"Show all roads lying in input minimum bounding rectangle (MBR)".

Result is as shown in Fig. 18.

Fig. 18 All roads lying in input MBR

Query 3:

"Show all important places and roads lying in input minimum bounding rectangle (MBR)".

Result is as shown in Fig. 19.

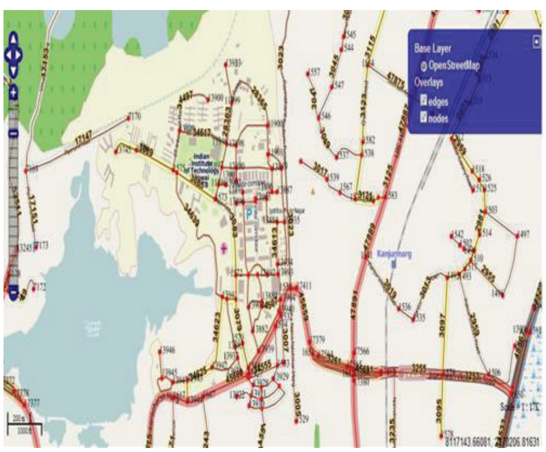


Fig. 19 All important places and roads lying in input MBR

Query 4:

"Show all entities represented by polygons input minimum bounding rectangle (MBR)".

Result is as shown in Fig. 20.

Fig. 20 All entities represented by polygons input MBR

Query 5:

"Show GPS data plotted on road network".

Result is as shown in Fig. 21.

Fig. 21 All entities represented by polygons input MBR

IV. Conclusion And Future Work

Open source technologies based application is architected and implemented which caters facility for execution of custom queries over road network datasets. It can be used for various purposes like but not limited to route planning, traffic prediction and planning, resource management etc.

Application focuses on handling queries mostly for road networks. It can be further extended to facilitate more complex datasets representing variety of entities like terrains, water bodies, land usage, state and country boundaries etc.

References

[1] Tiwari,V.S., Arya,A., Distributed Context Tree Weighting (CTW) for Route Prediction, Open Geospatial Data, Software and Standards, Springer Journal, 2018.

[2] Kavita V. V. Ganeshan, N. L. Sarda, Sanchit Gupta, Developing IITB Smart CampusGIS Grid, A2CWiC '10 Proceedings of the 1st Amrita ACM-W Celebration on Women in Computing in India ACM New York, NY, USA, 2010.

[3] Gonzalez,H, Han,J, Li,X, Myslinska,M, Sondag,J.P., Adaptive Fastest Path Computation on a Road Network: A Traffic Mining Approach, VLDB _07, Vienna, Austria, 2007.

[4] Tiwari,V.S., Arya,A., Distributed Context Tree Weighting (CTW) for Route Prediction, Open Geospatial Data, Software and Standards, Springer Journal, 2018.

[5] Touya, Guillaume & Courtial, Azelle. (2022). BasqueRoads: A Benchmark for Road Network Selection. Abstracts of the ICA. 4. 1-2. 10.5194/ica-abs-4-5-2022. 2022.

[6] Sankaranarayanan,J, Samet,H, Roads Belong in Databases, IEEE Data Eng. Bull., 2010: 4~11, 2010.

[7] Yue,Y, Zhuang,Y,Li,Q, Mao,Q, Mining time-dependent attractive areas and movement patterns from taxi trajectory data, 17th International Conference on Geoinformatics, Fairfax, VA, ISBN: 978-1-4244-4562-2, 2009.

[8] Speičys L., Jensen C.S., Road Network Data Model. In: Shekhar S., Xiong H. (eds) Encyclopedia of GIS. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-35973-1_1139, 2008.

[9] S, Satheeshkumar & A, Birundha & T, Sivabharathi & R, Subha & J, Swetha. Interpretation of Road Network using QGIS. Irish Interdisciplinary Journal of Science & Research. 06. 101-109. 10.46759/IIJSR.2022.6214, 2022.

[10] Sankaranarayanan,J, Samet,H,, Roads Belong in Databases, IEEE Data Eng. Bull., 2010: 4~11. 2010.

[11] Okte, Egemen., Determining Road Networks' Platoonability. Journal of Transportation Engineering Part A Systems. 147. 10.1061/JTEPBS.0000556. , 2021.

[12] Tiwari,V.S., Arya,A.,Chaturvedi, C., Scalable Prediction By Partial Match (PPM) and its Application to Route Prediction, Applied Informatics, Springer Journal, 2018.

[13] O'Sullivan, Patrick & Holtzclaw, Gary & Barber, Gerald., Theoretical Road Network Planning with Certainty. 10.4324/9781003182993-6, 2022.

[14] Tiwari, V.S., Arya, A. Horizontally scalable probabilistic generalized suffix tree (PGST) based route prediction using map data and GPS traces. Journal of Big Data 4, 23 (2017), https://doi.org/10.1186/s40537-017-0085-4, 2017.

[15] H. Sang, Y. You, X. Sun, Y. Zhou, F. Liu, The hybrid path planning algorithm based on improved A* and artificial potential field for unmanned surface vehicle formations. Ocean Eng. 223, 108–709 (2021).

[16] Burghardt, K., Uhl, J. H., Lerman, K., & Leyk, S., Road network evolution in the urban and rural United States since 1900. Computers, Environment and Urban Systems, 95, 101803, 2022.

[17] Fouzi Harrou, Abdelhafid Zeroual, Mohamad Mazen Hittawe, Ying Sun, Chapter 2 - Road traffic modeling, Road Traffic Modeling and Management, Elsevier, Pages 15-63, ISBN 9780128234327, https://doi.org/10.1016/B978-0-12-823432-7.00007-0., 2022.

[18] Wen-Long Jin,, Introduction to Network Traffic Flow Theory, Elsevier, Pages xvii-xviii, ISBN 9780128158401, https://doi.org/10.1016/B978-0-12-815840-1.00007-2., 2021.

[19] Saeedimoghaddam, M., & Stepinski, T. F., Automatic extraction of road intersection points from USGS historical map series using deep convolutional neural networks. International Journal of Geographical Information Science, 34(5), 947-968, 2022.

[20] Uhl, J. H., Leyk, S., Chiang, Y. Y., & Knoblock, C. A., Towards the automated large-scale reconstruction of past road networks from historical maps. Computers, environment and urban systems, 94, 101794, 2022.

[21] Michael A.P. Taylor, Chapter Two - Critical Infrastructure, Services, and Locations, Vulnerability Analysis for Transportation Networks, Elsevier, Pages 19-48, ISBN 9780128110102, https://doi.org/10.1016/B978-0-12-811010-2.00002-2., 2017

[22] Rafael S. Bressan, Pedro H. Bugatti and Priscila T.M. Saito, Optimum-path forest and active learning approaches for content-based medical image retrieval, Theory, Algorithms, and Applications, Academic Press, Pages 95-107, ISBN 978-0-12-822688-9, DOI https://doi.org/10.1016/C2019-0-04425-1, 2022.