Applied Mathematics & Information Sciences
*An International Journal*

http://dx.doi.org/10.12785/amis/092L24

# Semantic Rendering based on Information Hierarchy for Urban Navigation

*Divya Udayan J*[1], *HyungSeok Kim*[1,*], *Jee-In Kim*[2] *and Keetae Kim*[3]

[1] Dept. of Internet & Multimedia Engineering, Konkuk University, Republic of Korea
[2] Dept. of Advanced Technology Fusion, Konkuk University, Republic of Korea
[3] Daum Corporation, Republic of Korea

Received: 28 May 2014, Revised: 27 Jul. 2014, Accepted: 29 Jul. 2014
Published online: 1 Apr. 2015

**Abstract:** City street environments are highly populated with 3D building data and associated information. Therefore, a suitable data representation scheme is needed for effectively representing relevant information to an urban navigator. The focus of this paper is to propose a design methodology for providing users with just adequate information that helps them to satisfy their purpose of visiting a particular building in a city street. In addition, we suggest a mechanism that renders the requisite information with selective levels of detail by emphasizing the regions of interest and diminishing the details of less important regions, depending on the users intention. Preliminary tests of the proposed data scheme on a set of real street building data and its evaluations show the feasibility of our approach to extend to large scale mobile applications.

**Keywords:** Spatial semantic database, hierarchical data structure, level of detail, urban navigation

## 1 Introduction

Urban street environments are highly populated with 3D building information. The data quantity is too large to be represented in any data representation scheme. Even if such a representation were possible, the extend of its utilization and management in various applications including navigation systems, driving simulators, urban data management, traffic analysis, etc. would be a tedious and complex task. If a city is considered as an aggregation of large data units that can be decomposed into numerous meaningful objects, then each entity can be represented semantically [1] and its relationships can be expressed as a hierarchical data structure. This concept can be utilized in urban navigation applications, where the users main purpose is to find the destination building(s) of interest. In such a scenario, providing the entire street area data to the user would not be meaningful, as he would have to perform many cognitive interpretations himself, and as a result may sometimes lose the point of interest (POI) of his destination building. Detailed graphical content is indeed more subjective but also confusing to the user. Thus simplifying the content of street area data will lead to more usability, interactivity, and ease to find a particular building in a city street[2].

We propose a design methodology for filtering or diminishing unnecessary information, and emphasizing important areas in the street data. In the proposed design methodology, we represent the entire city information hierarchically as structural semantics using different levels of detail (LODs) for both geometry and texture. This structural hierarchy is then semantically mapped to user specific functional semantics. We represent the building model as a resource description framework (RDF) schema based on building ontology and its relationships with other building parts. This approach helps to prioritize buildings depending on functional clusters and associate them with the users intention in real time.

The remainder of this paper is organized as follows. Section II describes the related research. Section III focuses on semantic data modeling. Section IV shows the DB Schema of our building model. Section V discusses the application of our approach in a 3D platform. Section VI provides the experimental results and evaluation. Section VII presents some concluding remarks.

* Corresponding author e-mail: hyuskim@konkuk.ac.kr

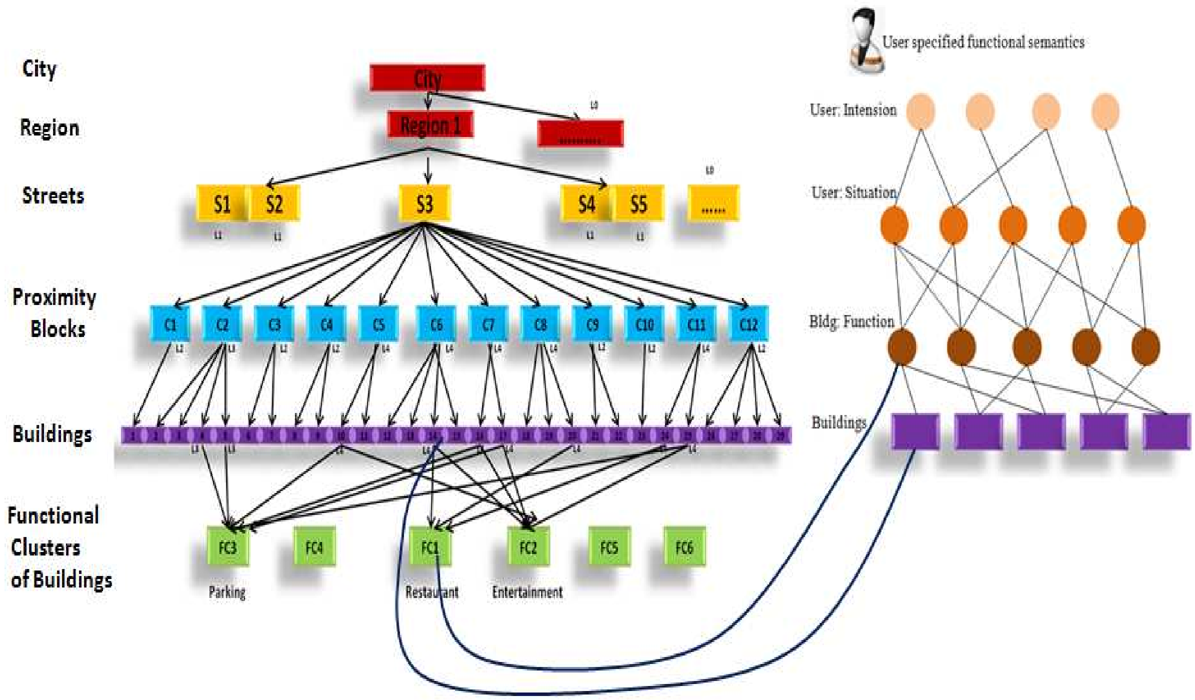


**Fig. 1:** The proposed semantic level of detail data structure definition, illustrating the mapping between structural semantics and user specified functional semantics.

## 2 Related Works

Research related to a new representation scheme for urban street navigation has become increasingly important in the last few years. A number of different standards and hierarchical classifications of 3D cities have been developed for this purpose [3]. Gao et al. [4] discuss several different visualization taxonomies, classifications, and terminologies, and describe how to use the ontology in the portal for discovery visualization services.

One purpose of the Top Level Visualization Ontology is to provide a common vocabulary to describe visualization data, processes, and products [5]. It describes some modifications for the visualization ontology that can provide better representation of the visualization process and data models. The Visualization Ontologies Workshop [6] investigated what the components were of such ontology, and how relationships from a given ontology could be derived. Such ontology could also facilitate sharing of the process models (pipelines) between visualization developers, and collaboration and interaction among users. M. Voigt et al. [7] systematically surveyed the broad corpus of visualization literature and discussed existing concepts, and found the difference between continuous and discrete ontology representation. We have designed our own visualization ontology for urban street navigation.

City Geography Markup Language (CityGML) [8][9][10] is another data model for representing the geometrical, topological, semantic, and appearance aspects of 3D city models. CityGML represents buildings and building parts using different LOD. The Open Geospatial Consortium (OGC) has adopted CityGML as an official standard for representing cities in 3D. CityGML distinguishes between different buildings and other artifacts in a city, such as transportation, vegetation, water bodies, city furniture, and streets. According to the CityGML encoding scheme, different buildings in a group may be given individual attributes such as Main Building. The representation and the semantic structure of buildings are refined from LOD1 to LOD4. The different LODs allow for a more detailed view and a low-poly representation of city models. A CityGML model may include several models of the same building but with different LODs, which is useful when different parts of the dataset have different sources and accuracy [11]. From this, we have designed a methodology for reducing the computational cost of transmission of real street area data using multiple level information representation. We have considered a photorealistic and non-photorealistic approach for providing different LODs to the user. The users intention for visiting a target building is also considered in our proposed data structure representation.

| Level | Imagery | | Terrain | | 3D Building |
|---|---|---|---|---|---|
| | Image resolution meters (/256) | Data | DEM resolution meters(/32) | Data | Data |
| 14 | 2048 | NDM | 16000 | NDM or DEM | NDM |
| 13 | 1024 | | 8000 | | |
| 12 | 512 | | 4000 | | |
| 11 | 256 | | 2000 | | |
| 10 | 128 | | 1000 | | |
| 9 | 64 | | 512 | | |
| 8 | 32 | Daum Sky View or True Ortho | 256 | | |
| 7 | 16 | | 128 | | |
| 6 | 8 | | 64 | | |
| 5 | 4 | | 32 | | |
| 4 | 2 | | 16 | | |
| 3 | 1 | | 8 | | |
| 2 | 0.5 | | 4 | | |
| 1 | 0.25 | | 2 | | |
| 0 | 0.125 | | 1 | | |

*NDM – Daum 2D Map data
*DEM – Digital elevation model data

**Fig. 2:** Quadtree level data representation

## 3 Semantic Data Modeling

Our approach for providing user friendly guidance for city navigation involves enhancing information based on the users intention and hiding less important data. Further, we consider a city as an aggregation of data entity that has numerous meaningful objects. Each entity can be represented semantically, and its relationships can be expressed as a hierarchical data structure, as shown in Fig.1. The semantics of each entity represents the meaning of building information by considering structural, functional, and user preferences aspects.

The city is decomposed into specific regions. The regions are further sub-divided into functional clusters for grouping the buildings. Each building can be further represented as multi-level information, represented using both structural and texture based LODs.

The user semantics consists of understanding the user's intention, analyzing the users situation, and mapping the users intention onto the functional group of buildings or a single building. This concept can be utilized in urban navigation applications, where the users main purpose is to find the destination buildings of interest. In that scenario, providing the entire street area data to the user would not be meaningful as he would have to perform many cognitive interpretations himself, and as a result may sometimes lose the point of interest (POI) of his destination building. Lessening the detailed content will improve the usability, interactivity and ease to find a building in a street. We propose a design methodology for filtering unnecessary information, and emphasizing important areas in the street data. Our approach is a multi-level information representation and visualization, considering the spatial hierarchy and visual level of detail.

To represent large city area, we use the quadtree data structure [12][13][14]. in which each internal node has exactly four children. This tree structure includes a set of tiles, where each tile is an image or map of the ground location at a fixed zoom level. The number of tiles in the tile set depends upon the size of the area depicted, and the resolution of the map. Tile positions within each zoom level are indexed by tile column and tile row numbers. Tile columns are numbered from left to right and tile rows from top to bottom of the global map. Any tile position can be identified by its quadtree level, tile row, and tile column. Fig.2 shows the quadtree level data representation.

We represent a cut in the quadtree as the current rendering list in the format <quadtree level, tile row, tile column>. We call the cut in the quadtree a map tile. Using our approach of semantic LODs, this map tile, which is in the form of a rendering list, is regenerated into a revised rendering list in the format <new quadtree level, tile row, tile column, rendering method>. The regeneration process is performed by generating an importance map, assigning different ranks to different tiles, as shown in Fig.3. Ranks are determined by semantic importance of entities in the tile. Having chosen the projection and scale to use at each level of detail, we can convert geographic coordinates into pixel coordinates. The pixel coordinates at each level differs because the map width and height is different at each level. The pixel

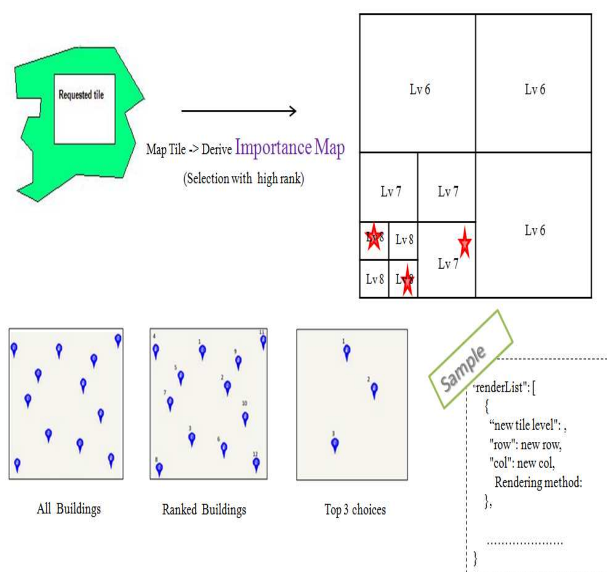


**Fig. 3:** Regenerating the rendering list based on the importance map of each tile.

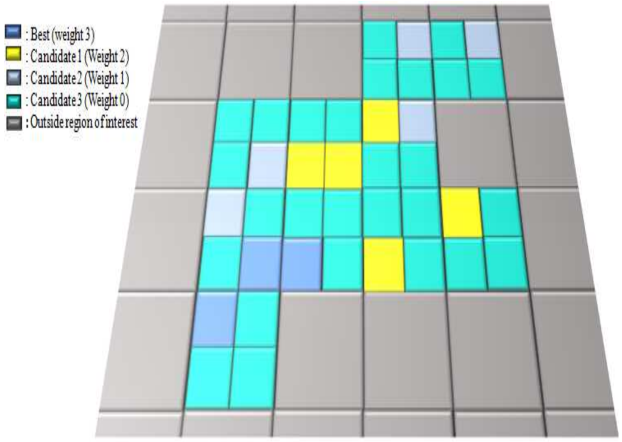


**Fig. 4:** Conceptual diagram illustrating cut in the quadtree for distributing semantic levels of detail

at the upper-left corner of the map always has pixel coordinates (0, 0). The pixel at the lower-right corner of the map has pixel coordinates (width-1, height-1). To optimize the performance of map retrieval and display, the rendered map is cut into tiles of 256 x 256 pixels. As the number of pixels differs at each level of detail, so does the number of tiles. Each tile is given XY coordinates ranging from (0, 0) in the upper left to (2level-1, 2level-1) in the lower right. Given a pair of pixel XY coordinates, we can easily determine the tile XY coordinates of the tile containing that pixel as in equation (1) and (2):

$$tileX = floor(pixelX/256) \quad (1)$$

$$tileY = floor(pixelY/256) \quad (2)$$

We need to calculate the corresponding tile levels, given the initial tile level (highest). When a zoom level changes, a single tile is broken into 4 tiles. If the zoom level increases by one then there are twice as many tile rows and columns. Hence, to derive the 4 tiles that an individual tile is broken into, we use the equations (3) to (7) :

$$tile1 : row*2, col*2; \quad (3)$$

$$tile2 : row*2+1, col*2; \quad (4)$$

$$tile3 : row*2, col*2+1; \quad (5)$$

$$tile4 : row*2+1, col*2+1; \quad (6)$$

Also,

$$nextzoomlevel = currentzoomlevel + 1 \quad (7)$$

For representing the visual LOD, we consider the users intention for visiting a particular building, and we categorize the buildings around a selective region as important or unimportant for the user. For example if User A wants to visit his/her acquaintance in a hospital, then the buildings of interest to him would be the hospital building, nearby fruit shops or flower shops, medical shops, car parking, nearest subway or bus station depending on the mode of travel. Those buildings would be emphasized and shown in higher LOD.

Other surrounding buildings can be filtered out or shown in lower LOD. This paves way for the application of multi-texture representation of the city buildings. Such representations allow regions of interest (ROIs) to be rendered in higher LOD than other non-ROIs, allowing interactive rendering of visual information. Both structural and texture hierarchies vary according to the semantic importance of the spatial location. When the user requests a particular map tile, the system regenerates the area giving different visual effects according to the importance/rank of the buildings, as shown in Fig.4. Each colored tile represents a particular rank within the ROI of the user. The buildings within one tile are given the same rank and hence similar visualization effect and detail.

## 4 DB Schema

A DB schema specifies, based on the DB administrator's knowledge of possible applications, the facts that can enter the DB, or those of interest to the possible end-users. The notion of a DB schema plays the same role as the notion of theory in predicate calculus[15]. A model of this theory can contain formulas representing integrity constraints specific for an application, and constraints specific for a type of DB, all expressed in the same DB language. In a RDB, the schema defines the tables, fields, relationships, views, packages, procedures, functions, queues, DB links, directories, XML schemas, and other elements. Although a schema is defined in text DB language, the term is often used to refer to a graphical depiction of the DB structure. In other words, schema is the structure of the DB that defines the objects in the DB.

We model our DB schema as RDF data based on building ontology and the various semantics associated in identifying a building, as shown in Fig.5. For example, the address and name of the building are unique characteristics by which a building is located in the geographic space. The appearance of the building is another factor that contributes to the recognition of a building. The functional importance is also a strong factor for building identification. The main classes and attribute relationship of the DB schema are shown in Fig.6.

## 5 Semantic LOD in 3D Platform

Semantic LOD is a mechanism by which a city street navigator is provided with just sufficient information to

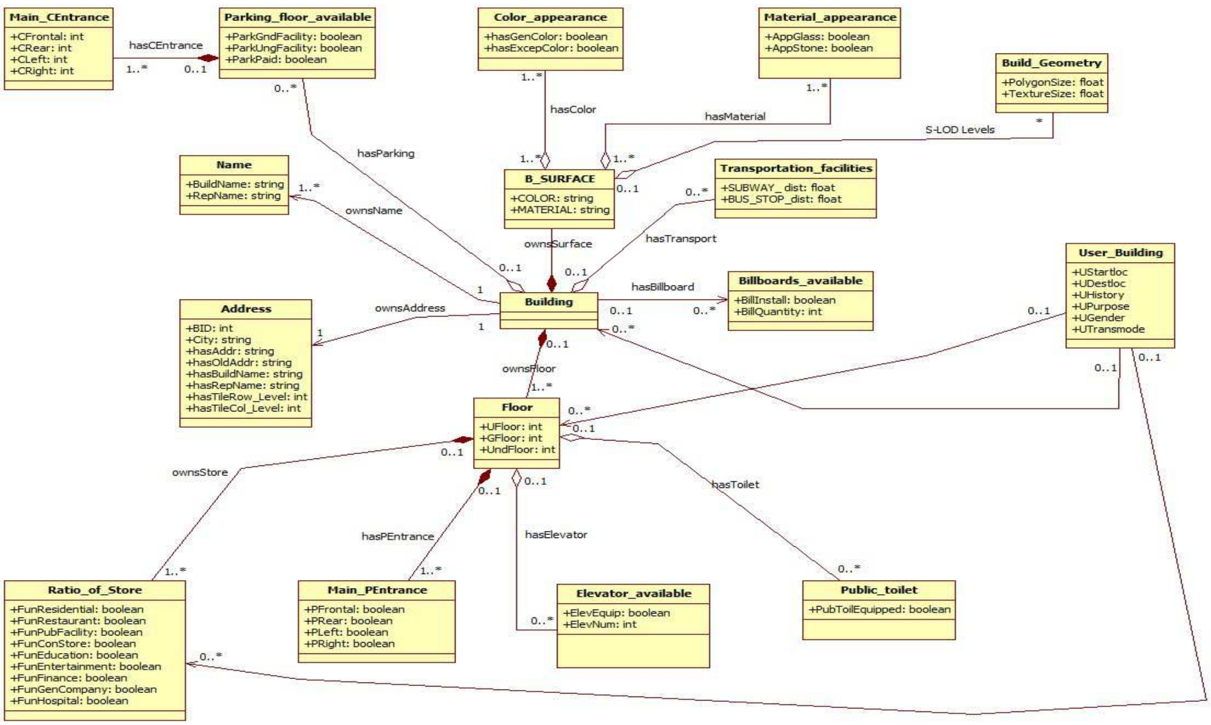


**Fig. 5:** UML diagram of the building database schema

| Class | Attribute | Relationship | Textual description |
|---|---|---|---|
| Address | BID | Building owns address | Building ID |
| | City | | City Name |
| | hasAddr | | Building address |
| | hasOldAddr | | Building's old address |
| | hasTileRow_Level | | Row of tile containing the building in quadtree |
| | hasTileCol_Level | | Column of tile containing the building in quadtree |
| Name | BuildName | Building owns Name | Building Name |
| | RepName | | Representative Name |
| B_SURFACE | COLOR | Building owns surface | Building color |
| | MATERIAL | | Building material |
| Color_appearance | hasGenColor | B_SURFACE has COLOR | Includes white, blue, grey |
| | hasExcepColor | | Includes red, yellow, mixture of colors |
| Material_appearance | AppGlass | B_SURFACE has Material | Includes glass like looking buildings |
| | AppStone | | Includes stone like looking buildings |
| Ratio_of_Store | FunResidential | Floor owns store with functionality | Residential building |
| | FunRestaurant | | Spacious Restaurants and convinient stores |
| | FunPubFacility | | Public buildings or government buildings |
| | FunConStore | | GeneralStores, clothes,shoes etc |
| | FunEducation | | Academic buildings |
| | FunEntertainment | | Threatres, music clubs, pub |
| | FunFinance | | Bank and other financial buildings |
| | FunGenCompany | | GeneralCompany and industry |
| | FunHospital | | Hospitals and pharmacies |

**Fig. 6:** Entities of building ontology that are relevant for the data modeling of semantic levels of detail.

guide the user to find the target location. Our proposed architecture helps to provide different visual effects representing semantic LODs according to the users intention. Such an approach provides a user friendly interface to find the target building. The system architecture and implementation details are described below.

## *5.1 System Overview*

The overall system architecture for applying our approach in a 3D platform involves the following main modules: semantic LOD generation, semantic LOD selection, and semantic LOD rendering module, as shown in Fig.7. The user gives input of the current location and intention to visit the target location. This user query is received by the semantic LOD generation module, which converts the user query to system query and passes it to the semantic query engine. The semantic query engine involves the Fuseki server and Jena API [16] for semantic information updating and querying. The semantic engine returns the query response in the form of <tile level, BID>, where tile levels range from 0-15, where tile 15 is the highest level; and BID denotes the building identification number. The semantic LOD generation module generates the importance map in the form <quadtree level, tile row, tile column, rendering method>. This importance map is

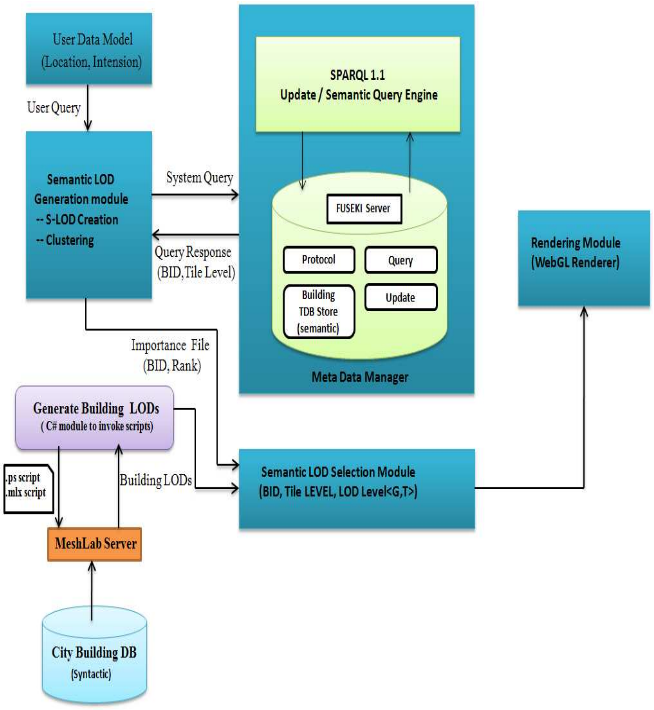


**Fig. 7:** The system architecture of applying semantic levels of detail in a 3D platform.

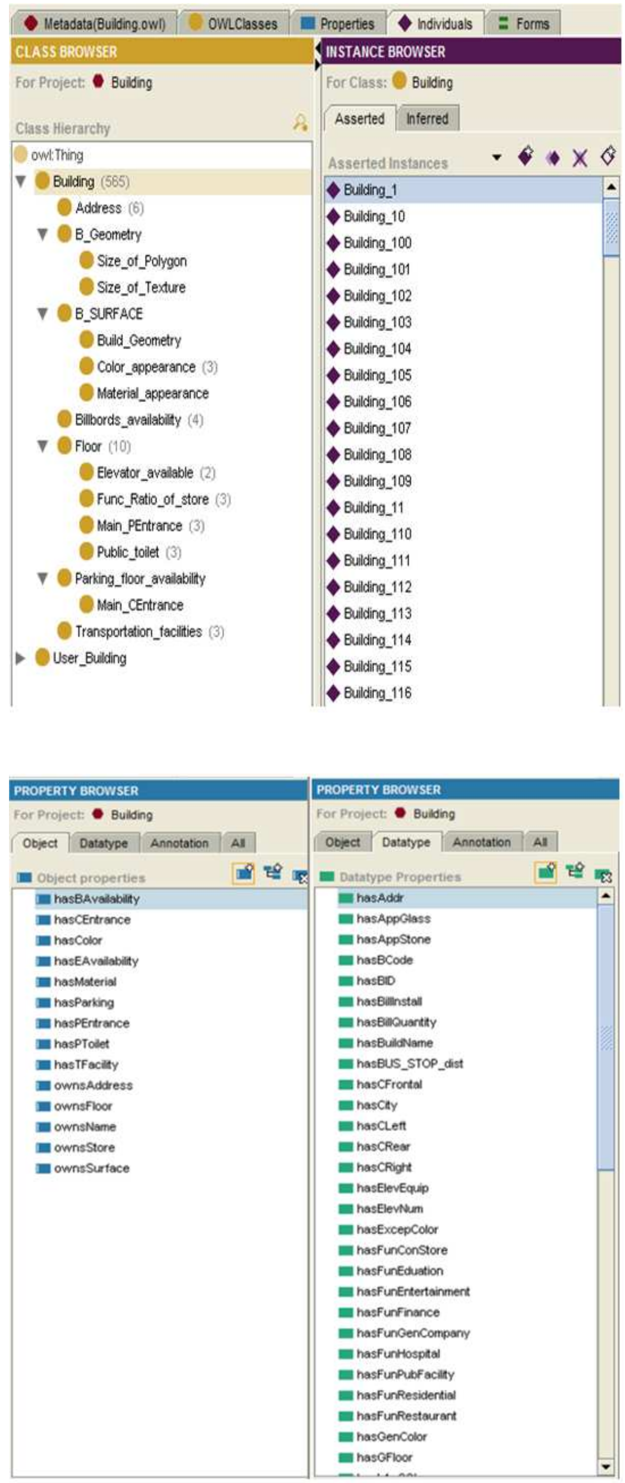


**Fig. 8:** Class hierarchy of different levels of building ontology(Top) property definition of the ontology(Bottom)

passed to the semantic LOD selection module where the tile level and LOD (geometry and texture) are selected and passed to the rendering module, which uses the WebGL renderer.

## *5.2 Semantic LOD Generation*

The main functionality of semantic LOD generation module is to create a spatial hierarchy for the distribution of semantic LODs. This is done by regenerating the rendering list in the format <quadtree level, tile row, tile column, rendering method>. We focus on an urban scenario in which cities are classified as regions and regions can be grouped as functional clusters both statically and dynamically, depending on the users intention. The city map is divided into regions using the quadtree data structure. Each quad tree tile is expected to contain a number of buildings. The semantic data associated with the buildings are stored in the form of ontologies.

Ontologies are a formal description of a shared conceptualization of a domain of interest. Designing ontology for storing building information requires good understanding of the building's geographic location, functional importance and the other semantics associated with a particular building. To design our ontology we surveyed about 600 buildings in a city street of Gangnam district in Seoul, South Korea. For example, a building

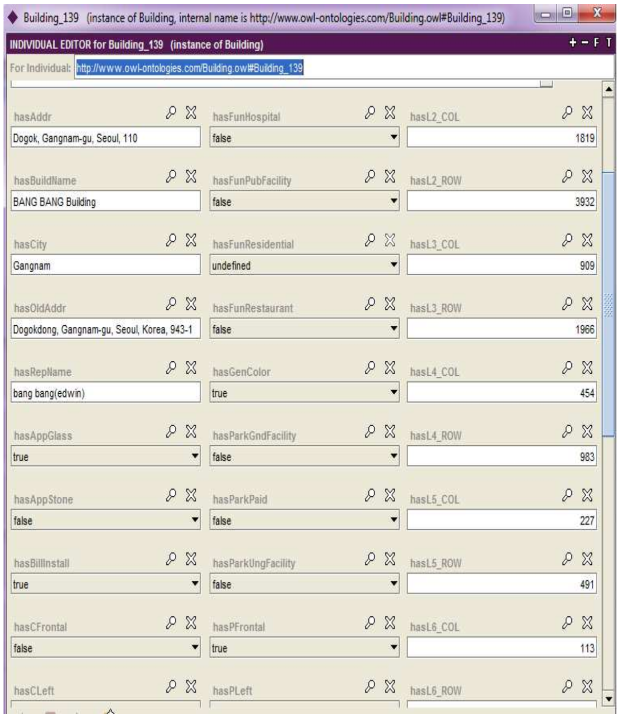


**Fig. 9:** Example of a building instance in the ontology

has its unique address, name, appearance, functionality, and so on. The properties of the building used in the ontology are shown in Fig.6. To develop our ontology, we used the open source editor Protege [17], developed at Stanford University. Protege facilitates designing concepts, properties and class instances; its plug-in based extensible architecture facilitates integration with other tools and applications; and it supports SPARQL [18] for querying RDF schema.

We use the RDF schema for storing the semantic information of city. The RDF data model is based upon the concept of making statements about resources in the form of subject-predicate-object expressions. These expressions are known as triples. The subject denotes the resource, and the predicate denotes the aspects of the resource and provides the relationship between the subject and the object. The semantic engine utilizes this information for system understandability. To do so, it needs an infrastructure that can run on the web and interpret meaningful information and provide that information to the user. Semantic engines are means to perform two important tasks. They provide a common format for integration and combination of data from different sources, and they act as a language for recording how the data is related to real world entities. Our building ontology with semantic information is shown in Fig.8 and Fig.9.

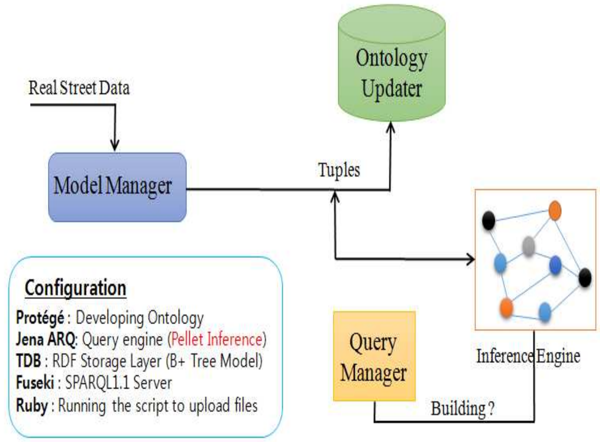


**Fig. 10:** Conceptual diagram of the querying process

## *5.3 Semantic LOD Selection*

We considered a real scenario in which the user intends to visit a building, which has restaurant, entertainment, and parking facilities. For this, we used the survey reports of Daum Corporation over 565 buildings. The semantic information and 3D models of buildings are stored as RDF data. We used the Fuseki server for updating and querying the semantic information in the semantic data store. The Fuseki server was running in a separate node for processing the semantic information. Using the proposed services, users can load the ontology, update the triples and query the triples using the SPARQL1.1 service. Communication between the application and the semantic server is established using graph store HTTP protocol. A code snippet of an ontology update is shown below.

```
  queryString ="PREFIX owl: <http://www.owl-ontologies.com/Building.owl#>\r\n" +
 "PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>\r"+
"INSERT DATA" +
     "{"+
     "<http://www.owl-ontologies.com/Building.owl#" +tt+IDNo + ">\n" +
     "owl:hasL9_ROW \"" + country[13] + "\"^^ xsd:int;" +
     "owl:hasL9_COL \"" + country[14] + "\"^^ xsd:int;" +
     "owl:hasL10_ROW \"" + country[11] + "\"^^ xsd:int;" +
     "owl:hasL10_COL \"" + country[12] + "\"^^ xsd:int;" +
     "owl:hasL11_ROW \"" + country[9] + "\"^^ xsd:int;" +
     "owl:hasL11_COL \"" + country[10] + "\"^^ xsd:int;" +
     "owl:hasL12_ROW \"" + country[7] + "\"^^ xsd:int;" +
     "owl:hasL12_COL \"" + country[8] + "\"^^ xsd:int;" +
     "owl:hasL13_ROW \"" + country[5] + "\"^^ xsd:int;" +
     "owl:hasL13_COL \"" + country[6] + "\"^^ xsd:int;" +
     "owl:hasL14_ROW \"" + country[3] + "\"^^ xsd:int;" +
     "owl:hasL14_COL \"" + country[4] + "\"^^ xsd:int;" +
     "}";
```

When the user inputs his interests, the query request will be passed to the semantic server, which processes it and returns the information to the application interface. During this process, the system finds the building IDs and their representative tile location in the spatial hierarchy. In our test demo we have structural semantic information and functional information of the buildings. The main advantage of this approach is that it is expandable to include additional semantic information. In the current work, we have considered only outdoor semantics but it could be extended to indoor semantics. The SPARQL query engine has a special type of protocol to communicate with ontologies and derive the meaningful information or for updating the triples. For querying the ontology, different rank conditions can be generated as shown below.

**Rank 1(R1):**

(RESTAURANT=true && ENTERTAINMENT=true && PARKING=true)

**Rank 2(R2):**

(RESTAURANT=true&& ENTERTAINMENT=true && PARKING=false)

**Rank 3(R3):**

(RESTAURANT=true&&ENTERTAINMENT=false&& PARKING=false)

**Rank 4(R4):**

(RESTAURANT=false&&ENTERTAINMENT=false&&PARKING=false

The semantic inference engine is where the higher level decision making occurs. This includes the model manager, ontology updater, and query manager. Fig.10 represents the conceptual design of the querying process. A code snippet for querying the building tiles, given the functional details, is shown below:

```
queryString1 ="PREFIX owl: <http://www.owl-ontologies.com/Building.owl#>\r\n" +
        "PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>\r" +
        "SELECT ?Total ?hasL14_ROW ?hasL14_COL ?hasL13_ROW ?hasL13_COL ?hasL12_ROW ?hasL12_COL "+
        "WHERE"+
        "{"+
        "?Building owl:hasL14_ROW ?hasL14_ROW." +
        "?Building owl:hasL14_COL ?hasL14_COL." +
        "?Building owl:hasL13_ROW ?hasL13_ROW." +
        "?Building owl:hasL13_COL ?hasL13_COL." +
        "?Building owl:hasL12_ROW ?hasL12_ROW." +
        "?Building owl:hasL12_COL ?hasL12_COL." +
        "?Building owl:hasL11_ROW ?hasL11_ROW." +
        "?Building owl:hasL11_COL ?hasL11_COL." +
        "?Building owl:hasFunEntertainment ?hasFunEntertainment."+
        "?Building owl:hasFunRestaurant ?hasFunRestaurant."+
        "?Building owl:hasParkPaid ?hasParkPaid."+
        "BIND (xsd:integer(?hasFunEntertainment) as ?Entertainment)"+
        "BIND (xsd:integer(?hasFunRestaurant) as ?Restaurant)"+
        "BIND (xsd:integer(?hasParkPaid) as ?ParkPaid)"+
        "BIND (?Entertainment + ?Restaurant + ?ParkPaid as ?Total)"+
        "FILTER ((?hasL11_COL=906 && ?hasL11_ROW=1972))"+
        "}"+
        "order by ?Total";
```

The updating process consists of the user requesting real street data from the server. The semantic DB engine processes the query, converts it into a system query, and search the RDF data set for BID(s) and quadtree tile(s) ( tile row and tile column). The query result is passed back to the DB engine by the Fuseki server, then converted to

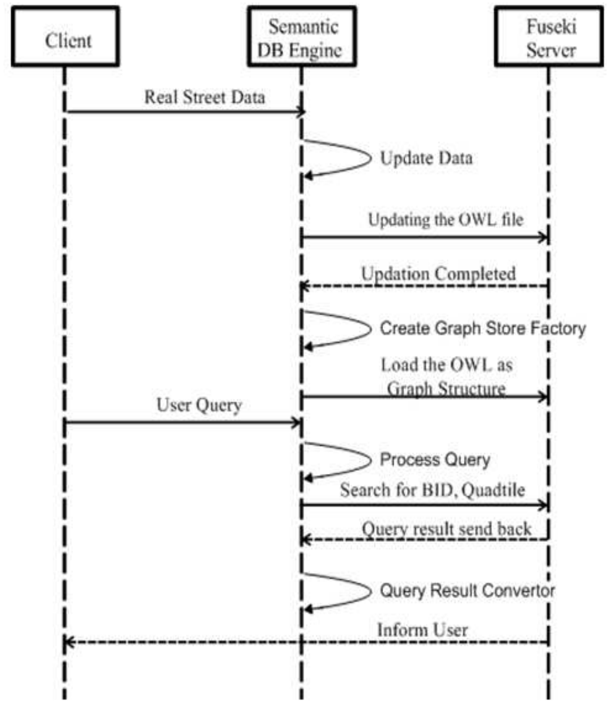


**Fig. 11:** Sequence diagram of execution flow for the updating and querying process.

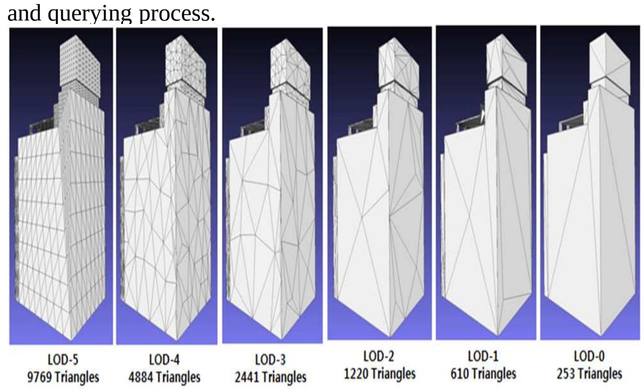


**Fig. 12:** Polygonal hierarchy showing an increment in geometry.

information for the user. Fig.11 shows the sequence diagram for the execution flow of the entire process.

## *5.4 Semantic LOD Rendering*

Automatic tile ranking is performed by the semantic LOD selection module. After selecting the tile location (row, column) corresponding to the users intention, the semantic query engine regenerates the rendering list and this is passed to the WebGL renderer. Each level is displayed in an incrementing manner of geometry and texture resolution as shown in Fig.12. The highest level corresponds to high detailed building geometry and texture. The lowest level is a footprint showing only the location of the building.

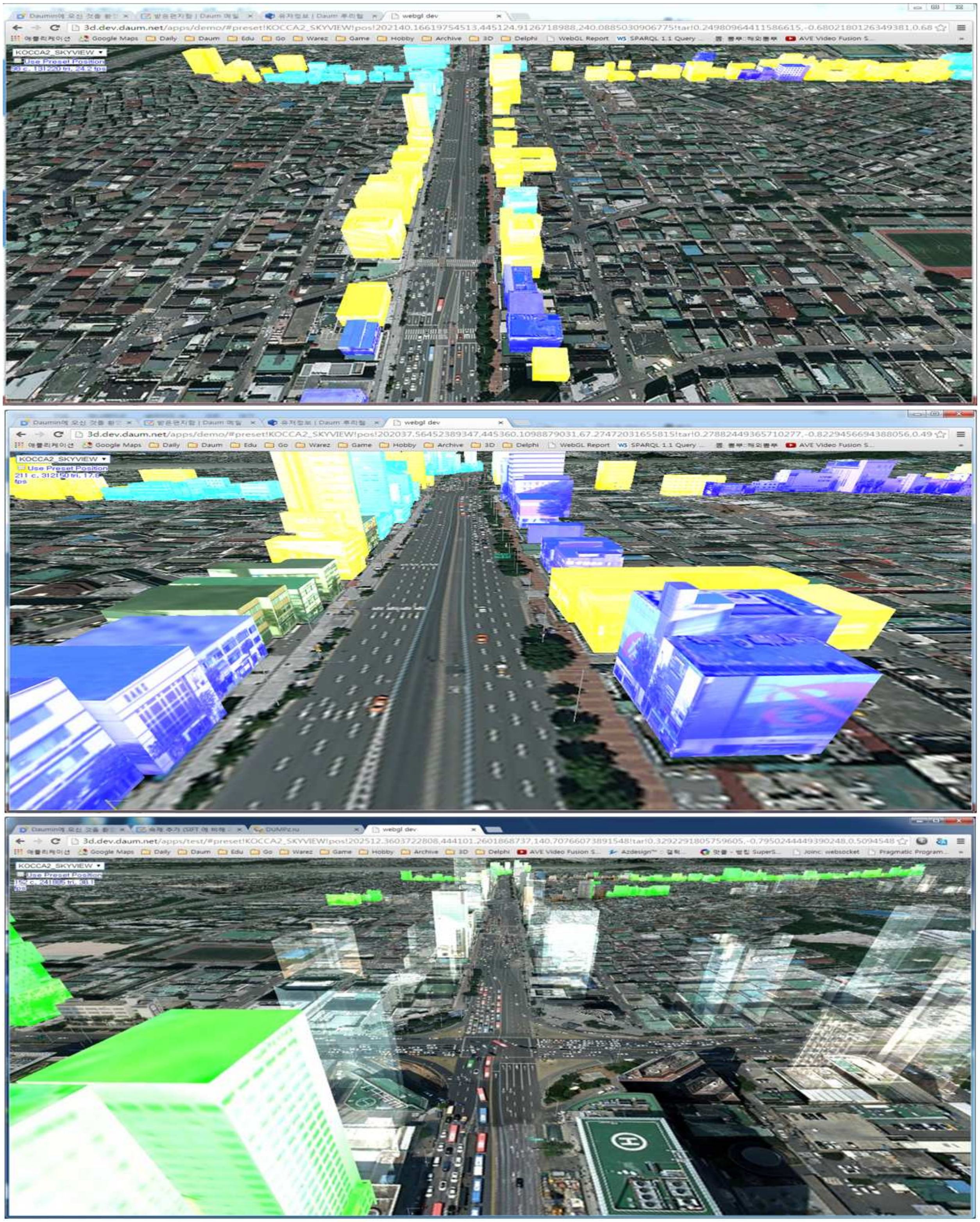

**Fig. 13:** Real street data rendered using Semantic levels of detail concept. (Top) Birds eye view of a street in Gangnam district, Seoul, South Korea depicting information hierarchy using unique colors. (Middle) closer view of the street area. (Bottom) Non photorealistic rendering technique applied to the information hierarchy, visualization of less important data is diminished.

We use the WebGL pipeline for rendering as it provides greater flexibility and adaptability for GPU programming in polygon based graphics systems. Each tile is rendered using different rendering methods. The rendering method varies in incrementing manner both in geometry and texture. The experimental results and observation are explained in the next section.

## 6 Experimental Results and Discussion

In the previous sections, we explained the data modeling and DB schema for semantic LOD used to represent a building and its different parts in multiple levels. We verify our proposed system using simulation results and performance evaluations. For the test environment, we prepared a general desktop PC as a server and a laptop as a client. The laptop contains an I7 core CPU, 4GB RAM, and NVIDIA GT 620 M GPU. The proposed system was tested and evaluated using a set of real street data of six hundred real buildings, located in the Gangnam district in Seoul, South Korea. The user intends to visit a building that has restaurant, entertainment, and parking facilities. Different rank conditions are generated as explained in the semantic LOD selection module. The proposed system dynamically changes semantic LODs of the viewed buildings according to their corresponding information levels determined by the rank conditions. The visualization levels of detail were based on both the structural and texture hierarchy. In our current implementation we define two user contexts. One is when the user is moving, and the other when the user stops to explore the surrounding buildings to find his/her destination. When the user stops, the system provides full detailed high level information of the buildings on the street. When the user starts moving, different non-photorealistic appearance of the buildings are rendered depending on the user gaze and his distance from the buildings position. Fig.13 shows real street data rendered using the semantic LOD concept. Information hierarchy is depicted using unique colors. Sky-blue represents the best choice, yellow is second, followed by blue and green.

We conducted two types of tests for performance evaluation. The first is based on the rendering speed. Each user tried the process twenty times and we measured an average rendering speed of the proposed system. Table 1 describes the total number of buildings, average number of viewed buildings, average building size, average download time per building, and average FPS. The results showed a feasible rendering speed, although it is dependent on the navigation scenario. The second test evaluated the systems running behavior.

Fig.14 shows the running behavior of the system with respect to the four system states: Geometry busy, Shader busy, Texture busy, and GPU busy. Figure 12 (top) shows the variations in the behavior of system when no semantic level of detail is applied, and (bottom) shows the variations in the behavior of the system when our proposed semantic level of detail hierarchy is applied. This system running behavior showed the validity of using different semantic level of detail to cut down the computation cost.

**Table 1:** Results of performance evaluation

| **Total buildings** | **Avg. no.of viewed buildings** | **Avg. size of the building in(Mega Bytes)** | **Avg. down loading time per one building** | **Avg. FPS** |
|---|---|---|---|---|
| 565 | 45 | 3.647 | 18ms | 30 |

## 7 Conclusion

This paper proposes a new design methodology for a spatial semantic database that provides different levels of detail using a hierarchical data structure. This semantically enriched approach is based on ontology and its associated knowledge representing the city model in different levels of detail. With this approach, the level of detail varies according to the users intention and services in the geometric level. Since we are using an ontology based representation for the city model, adjacent structures and their relationships are easily derived and adopted. In addition, the performance evaluation shows the validity of using different level of detail to cut down the computation cost.

Regarding the feasibility of our approach, we have interviewed navigation company people who are map service experts. They have reported that our approach is quite reliable and useful. In addition, they have mentioned that our current approach visual quality is still marginal. As future work, we will consider their suggestions and improve the visual quality using revised rendering methods. In addition, we will refine our present semantic level of detail data structure to include more user semantics and improve the implementation on real street data. Moreover, we will concentrate on continuously streaming the real data at different levels to users of heterogeneous devices simultaneously, using an extended server-client platform supporting various mobile devices in real time.

## Acknowledgement

This research was supported by grant from the Capture Korea Project funded by the Ministry of Culture, Sports and Tourism (MCST) and Korea Creative Content Agency (KOCCA) in the Culture Technology (CT) Research and Development Program 2014(50%). A part of this research was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded

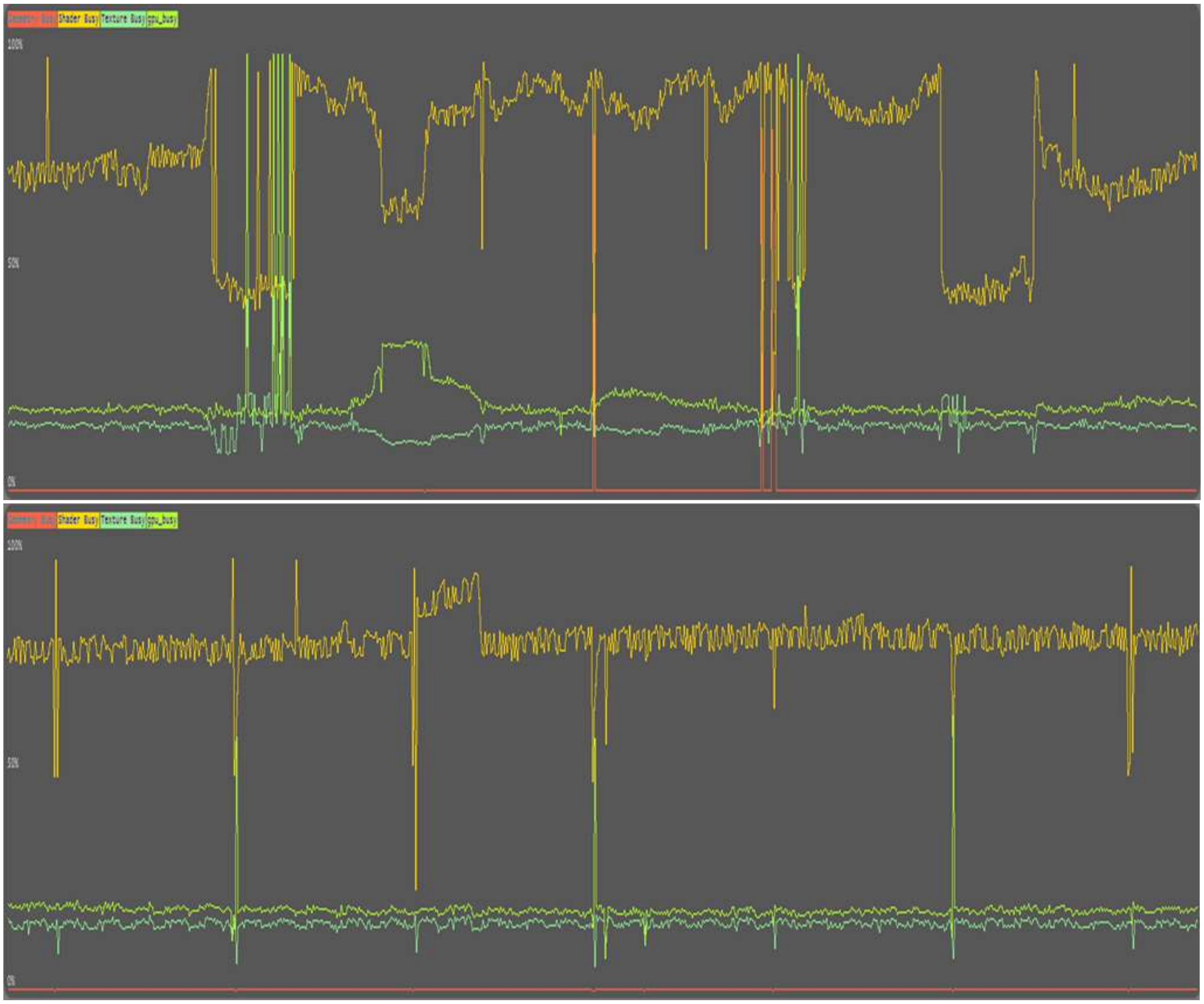

**Fig. 14:** Running Behavior of the system with respect to the four system states: Geometry busy, Shader busy, Texture busy, and GPU busy. (Top) shows the variations in the behavior of system when no semantic level of detail is applied, (Bottom) shows the variations in the behavior of the system when our proposed semantic level of detail hierarchy is applied.

by the Ministry of Education, Science and Technology (No. 2012M3C4A7032185) (50%).
The authors are grateful to the anonymous referee for a careful checking of the details and for helpful comments that improved this paper.

## References

[1] Divya Udayan J and HyungSeok Kim, Ontology of 3D Visualization for Urban Navigation, Proceedings of KHCI2013, 149-152 (2013).

[2] Divya Udayan J, HyungSeok Kim, Jun Lee, Jee-In-Kim, and Keetae Kim, Multi-Level Information Representation Based LOD Streaming for Urban Navigation, Proceedings of IEEE ISUVR2013, 30-33 (2013).

[3] Groger G, Kolbe T.H, and Czerwinski A, OpenGIS CityGML Implementation Specification, OpenGeospatial Consortium, V0.3.0. OGC Doc. No.06-057 (2006).

[4] Gao A. S., Nick J, and Omer FR., Bringing semantics to visualization services, Advances in Engineering Software, Elsevier, **39**, 514-520 (2008).

[5] Perez A. M, Perez C. R, and Gomez J. M, An Enhanced Visualization Ontology for a Better Representation of the Visualization Process, Proceedings of ICT Innovation 2010, Communications in Computer and Information Science, Springer-Verlag, Berlin Heidelberg, **83**, 342-347 (2010).

[6] Brodlie K.W., Duce D.A., Duke D.J, Visualization ontologies :- Report of a workshop held at the nationale science centre. Report e-Science Institute, (2004).

[7] Voigt M and Polowinski J, Towards a Unifying Visualization Ontology :-Technical report, Institute of Software and Multimedia, Technical University Dresden, Germany,

(2011).
[8] Stadler A and Kolbe T, Spatio-Semantic Coherence in the Integration of 3D City Models, 5th International Symposium on Spatial Data Quality ISSDQ, (2007).
[9] Groger G, Kolbe T. H, Czerwinski A, and Nagel C, OpenGIS City Geography Markup Language (CityGML), OGC Encoding Specification, V 1.0.0, Doc. No.08-007r1, (2008).
[10] Kolbe T. H, Groger G, and Plumer, L, CityGML Interoperable Access to 3D City Models, In: Proc. of the 1st International Symposium on Geo-information for Disaster Mangement, (2005).
[11] Czerwinski A, Groger G, Kolbe T, and Nagel C, City Geography Markup Language (CityGML) Encoding Standard, Open Geospatial Consortium Inc., (2008).
[12] Samet H and Webber R, Storing a Collection of Polygons Using Quadtrees, ACM Transactions on Graphics **4**, 182-222, (1985).
[13] Bing map tile systems, http://msdn.microsoft.com/en-us/library/bb259689.aspx
[14] Google maps, http://www.maptiler.org/google-maps-coordinates-tile-bounds-projection/
[15] Rybinski H, On First Order Logic Databases, ACM Transactions on Database Systems, **12**, 325-349 (1987).
[16] Fuseki server, Jena API, http://jena.apache.org.
[17] Protege 4.3, http://protege.stanford.edu/
[18] A. Seaborne, A. Polleres, L. Feigenbaum, and G. T. Williams. SPARQL 1.1 Federated Query. W3C Working Draft 17 November (2011).http://www.w3.org/TR/2013/REC-sparql11-update-20130321/

---

**Divya Udayan J** received a BTech degree in Information Technology from Kerala University in 2005 and an MTech degree in Embedded Systems from Amrita University, India in 2010. She is currently a PhD student in Internet and Multimedia Engineering at VR Lab, Konkuk University, Korea. Her research interests include augmented reality, semantic modeling, image processing, recognition and classification and HCI.

**HyungSeok Kim** received a PhD degree in Computer Science in February 2003 at VR Lab ,KAIST. He is an Associate Professor at the Department of Internet and Multimedia Engineering, Konkuk University, Korea. Before joining Konkuk University, he was a senior researcher at MIRA Lab, University of Geneva, being involved in research activities on virtual reality. His major research field is real-time interaction in virtual environments and multimodal interaction mechanisms. His current research activities are focused on topics of shape modeling for real-time rendering and evoking believable experiences in virtual environments.

**Jee-In Kim** received a BS degree in Computer Engineering from Seoul National University in 1982 and an MS degree in Computer Science from Korea Advanced Institute of Science and Technology in 1984. He received his PhD in Computer Science and Information in May 1993 from the University of Pennsylvania. He is a Professor of Internet and Multimedia and Advanced Technology Fusion at Konkuk University. He is the Director of the Digital Contents Research Center at the University. His major research fields include ubiquitous computing, visualization and HCI.

**Keetae Kim** received a PhD degree in Mapping and GIS in March 2004 at Ohio State University, Columbus. He is the local development lead at Daum Communications, Seoul, S. Korea. Before joining Daum, he was the division head of 3D solutions business division at SK C and C, S. Korea. He was also an employee at Microsoft Corporation, Boulder, CO, USA. His major research field is geospatial mapping using satellite remote sensing data, automatic 3D model reconstruction from 2D imagery, and building AR based user experiences.