Rethinking GIS Towards The Vision Of Smart Cities Through CityGML

C. Guney

Istanbul Technical Univeristy, Department of Geomatic Engineering, 34469 Maslak Istanbul, Turkey - guneycan@itu.edu.tr

KEY WORDS: 3D city model, 3D web GIS, smart city, CityGML, WebGL, virtual globe, NoSQL, semantics

ABSTRACT:

Smart cities present a substantial growth opportunity in the coming years. The role of GIS in the smart city ecosystem is to integrate different data acquired by sensors in real time and provide better decisions, more efficiency and improved collaboration. Semantically enriched vision of GIS will help evolve smart cities into tomorrow’s much smarter cities since geospatial/location data and applications may be recognized as a key ingredient of smart city vision. However, it is need for the Geospatial Information communities to debate on “Is 3D Web and mobile GIS technology ready for smart cities?” This research places an emphasis on the challenges of virtual 3D city models on the road to smarter cities.

1. MOTIVATION

In many countries, addition to people chose to live in big cities, an increasing number of people from small towns and villages is moving into big cities. Globally more people live in urban areas than in rural areas. Production, processing, query, analysis, visualization, updating, maintaining and sharing of three dimensional (3D) geospatial data related to cities are very important tasks for managing the urban environments in 3D while the world continues population growth and urbanization.

The possibility to visualize and interact with 3D city models on the Web is of interest for technicians, decision makers and citizens (Rodrigues et al., 2013). The accessibility of the online city models through web browsers can enlarge the audience of the 3D model to broad audience and professionals that typically are not expert on geospatial information but who can be benefit on their work from the usage of 3D city models (Prandi et al., 2015).

There is a rapidly increasing need for 3D geospatial information and 3D city models for many different areas of application like urban planning, landscape planning, environmental planning, 3D cadaster, real-estate, public participation, facility management, disaster management, transportation, energy planning, tourism, simulation, and analysis. These application areas are also central to the very notion of a smart city.

Today, more and more cities worldwide are undergoing transition from 2D Geospatial Information System (GIS) to 3D Web GIS, and using Web 3D city viewers or virtual globes. Recently several European cities like Berlin, Lyon, Wien and Rotterdam, for instance, have realized their official 3D city models as Open Data by means of LOD2/LOD3 textured CityGML models of the whole city (Prandi et al., 2015, Mao et al., 2014, Gaillard et al., 2015). Industry companies like Google, Apple and Here are also integrating 3D city models into their map services (Mao et al., 2014, Prandi et al., 2015). Volunteered Geographic Data/Information (VGI) communities such as Open Street Map (OSM) are building 3D city models as well (Mao et al., 2014). From this point of view opportunities to utilize 3D urban mapping and the volume of online virtual city models are expanding dramatically and virtual 3D city applications are increasingly employed in different domains. The successful utilization of 3D city models for business processes of the tasks of modern city management, such as urban planning, reveals the benefits of 3D city models and they are widely used in almost every field. As a consequence of that 3D technologies covered the whole production chain of city modeling, 3D city models have become standard component for city management tasks and virtual globes have become new medium to display and interact with the city models.

Presently the main use of 3D city models is still focused on visualization capabilities and is inadequate for performing many types of 3D spatial analysis. However, today’s applications request for common, semantic rich spatial information models to serve a wide range of use cases beyond visualization, for instance enabling a variety of GIS analytics. Unfortunately, applying 3D GIS analysis capabilities in virtual 3D city model applications is still rare.

However, semantic 3D city models describe city entities by objects with thematic and spatial attributes and their interrelationships. Web-based semantic rich 3D virtual city models provide a common platform to integrate city level information between different domains and to share 3D geo-referenced information across domains for better understanding urban processes and designing innovative solutions. Highly accurate, up-to-date, 3D virtual models allow 3D data flow along the city management processes from plan and design to maintenance. 3D city models can be updated by governments, companies and citizens using a variety of human and technical sensors (in-situ and remote). If the new data/information is added from different domains, the usefulness of the common semantic rich city model will continue to increase as additional applications and workflows are developed.

With the advancement in 3D technologies for city modeling, virtual 3D city models can be customized to employ vertical applications via more specific 3D computational analyses, like solar potential analysis of roof surfaces, urban noise distribution, building shadowing, storm water runoff, flood modeling, the visibility impacts of new development or urban growth simulations based on a variety of what-if scenarios. Due to their efficiency, numerous GIS-based applications, smart e-services and advanced tools are being developed as smart urban technologies for representing and analyzing 3D city.

This contribution has been peer-reviewed.
doi:10.5194/isprs-archives-XLII-2-W1-121-2016

In smart cities data being produced by sensors is enormous and there is a strong need to time these data streams and build applications and services to take smart decision by performing analysis of these data streams in real-time. Web-based GIS has also capabilities for analyzing sensor data in real-time and visualizing it on interactive dynamic maps to support real-time decision-making. Spatial informatics technologies, such as GIS services, Location-based Services, can be utilized to develop geo-based services for smart city applications.

2. PROBLEM DEFINITION

The motivation of this study is to design a conceptual approach allowing the user to achieve the following research objectives:

- Improving collaborative city management and support decision making tasks relying on online 3D city model
  - evaluating different urban growth scenarios, different zoning rules, selecting the most promising ones and presenting them online to the public for commenting and voting
  - aiding decision makers in policy modelling
  - displaying existing zoning laws with the 3D buildings to assess which buildings are in compliance
- Improving urban planning processes relying on online 3D city model
  - supporting urban planners to present their plans to the public and other stakeholders
  - giving the public the means to participate in urban planning processes
- Improving simulation for a variety of what-if scenarios to predict the impact of an event, like storm water runoff (flooding), storm surge, energy, noise emission, and assessing the effects of a proposed new construction/infrastructure on its surroundings
- Improving 3D spatial analyses relying on online 3D city model
  - integrating temporal component and context aware operations to perform change analysis and to assess the impact of the change

Considering the research objectives, the following main challenges have to be faced to overcome:

- Content preparation
  - Organization of 3D data
  - 3D object reconstruction
- Modeling framework
  - 3D city information modeling
  - Generalization of 3D city model
  - Semantic 3D content modeling
- 3D visualization framework
  - 3D visualization of city information models in a web browser
  - Development of a web client able to render the 3D city models
  - Interactive dynamic real-time visualization via web
  - Making the virtual 3D visualization as real as possible
  - Efficient visualization of data from different sources provided by the streaming services on the spinning globe
  - Service-based visualization
  - Progressive visualization
  - Semantic visualization
  - 3D rendering process
  - Streaming process
  - Tiling process
  - GPU Memory Management
- Storage framework
  - 3D city model management system
  - Managing massive amounts of data, ranging from several gigabytes to terabytes
  - Gathering different types of data from different sources
- 3D Functionality
  - 3D Query on the web
  - 3D Analysis on the web
  - 3D Editing of the city objects on the web
- Sharing virtual city models

There is no mainstream solution yet which handles all these challenges. The goal of this study is an attempt to conceptualize a method in order to manage increasingly complicated 3D city models for different kinds of vertical applications by focusing on smart cities. This paper draws a conceptual framework to the formulation of a 3D semantic web GIS for a complex 3D city information models accessible on a WebGL based virtual globe by a HTML5 enabled web browser compliant to OGC Web Services based on NoSQL database. The development of such a robust and adaptive GIS-based system can be valuable for the GIS community to be well prepared for building smart cities.

Figure 1 presents the high-level client-server architecture of the proposed framework designed in the study. It is composed of different frameworks, standards, components, technologies and their interaction with each other based on RESTful SOA architecture.

Although there are many points to be considered, two points, in particular, have been concentrated in this research. These are 3D visualization framework on the client-side and storage framework on the server-side. OGC web services allow fast retrieval of large amount of data/information requested by the client from the servers.

3. CONTENT PREPARATION

3.1 Data Acquisition

Geospatial data such as 3D city models, digital terrain models, point clouds, aerial images, 2D/3D geometries, TIN, DEM and metadata are inherently heterogeneous and oftentimes very large (Kramer and Gutbell, 2015). Hence, an Urban Spatial Data Infrastructure (Urban SDI or City Information Infrastructure) is a prerequisite for a city model in order to organize 3D geospatial data and generate 3D models of city objects. Such an infrastructure can be used to visualize virtual 3D city models based on the various data from different sources in different resolutions, e.g. detailed 3D textured city models.

Recent technological advances in sensor and platform technologies, such as nadir and oblique high resolution stereo cameras, aerial and terrestrial laser scanners, have greatly improved the 3D data collection techniques. This development provides new sources for constructing building models and generating the terrain to meet the requirements of 3D city modeling and mapping.

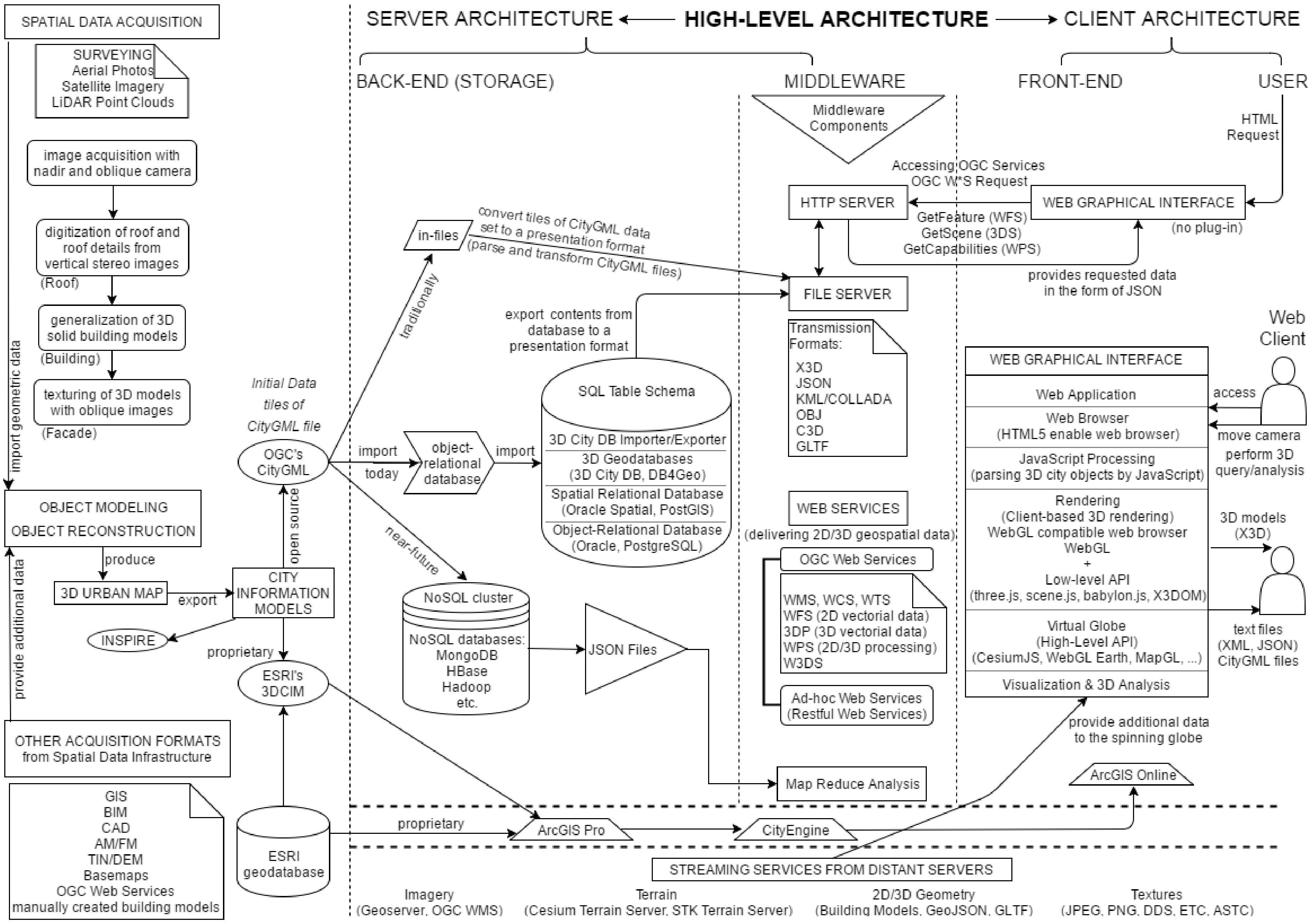
SPATIAL DATA ACQUISITION
SURVEYING
Aerial Photos
Satellite Imagery
LiDAR Point Clouds
image acquisition with nadir and oblique camera
digitization of roof and roof details from vertical stereo images
(Roof)
generalization of 3D solid building models
(Building)
texturing of 3D models with oblique images
(Facade)
import geometric data
OBJECT MODELING
OBJECT RECONSTRUCTION
produce
3D URBAN MAP
export
CITY INFORMATION MODELS
INSPIRE
provide additional data
OTHER ACQUISITION FORMATS from Spatial Data Infrastructure
GIS
BIM
CAD
AM/FM
TIN/DEM
Basemaps
OGC Web Services
manually created building models
SERVER ARCHITECTURE
HIGH-LEVEL ARCHITECTURE
CLIENT ARCHITECTURE
BACK-END (STORAGE)
MIDDLEWARE
FRONT-END
USER
Initial Data tiles of CityGML file
OGC's CityGML
open source
proprietary
ESRI's 3DCIM
ESRI geodatabase
traditionally
in-files
convert tiles of CityGML data set to a presentation format
(parse and transform CityGML files)
import
today
object-relational database
import
near-future
export contents from database to a presentation format
SQL Table Schema
3D City DB Importer/Exporter
3D Geodatabases (3D City DB, DB4Geo)
Spatial Relational Database (Oracle Spatial, PostGIS)
Object-Relational Database (Oracle, PostgreSQL)
NoSQL cluster
NoSQL databases:
MongoDB
HBase
Hadoop
etc.
JSON Files
Map Reduce Analysis
proprietary
ArcGIS Pro
CityEngine
ArcGIS Online
Middleware Components
HTTP SERVER
FILE SERVER
Transmission Formats:
X3D
JSON
KML/COLLADA
OBJ
C3D
GLTF
WEB SERVICES
(delivering 2D/3D geospatial data)
OGC Web Services
WMS, WCS, WTS
WFS (2D vectorial data)
3DP (3D vectorial data)
WPS (2D/3D processing)
W3DS
Ad-hoc Web Services (Restful Web Services)
Accessing OGC Services
OGC W*S Request
GetFeature (WFS)
GetScene (3DS)
GetCapabilities (WPS)
WEB GRAPHICAL INTERFACE
(no plug-in)
HTML Request
provides requested data in the form of JSON
Web Client
WEB GRAPHICAL INTERFACE
Web Application
Web Browser (HTML5 enable web browser)
JavaScript Processing (parsing 3D city objects by JavaScript)
Rendering (Client-based 3D rendering) WebGL compatible web browser WebGL + Low-level API (three.js, scene.js, babylon.js, X3DOM)
Virtual Globe (High-Level API) (CesiumJS, WebGL Earth, MapGL, ...)
Visualization & 3D Analysis
access
move camera
perform 3D query/analysis
3D models (X3D)
text files (XML, JSON) CityGML files
provide additional data to the spinning globe
STREAMING SERVICES FROM DISTANT SERVERS
Imagery (Geoserver, OGC WMS)
Terrain (Cesium Terrain Server, STK Terrain Server)
2D/3D Geometry (Building Models, GeoJSON, GLTF)
Textures (JPEG, PNG, DDS, ETC, ASTC)

Figure 1. Three level architecture for the 3D geospatial applications on the web

Approach	Method	Technique	Source of Data	Advantages	Disadvantages	When Used?	Software Tools
Photo-Realistic Approach	Semi-Automatic	Remote Sensing (comprising Aerial and Close Range Photogrammetry, Satellite Imagery, Airborne and Ground based Laser Scanning (LiDAR))	High Resolution Satellite Images, Point Clouds, Aerial Photographs, Orthophotos, Orthoimagery, DTM (TIN and DEM)	Effective visualization with roof modeling and facade texturing, high quality DTM production, accurate and informative, partially automated	Inadequate for performing many types of 3D spatial analysis	If a significant number of building models need to be constructed	CyberCity3D, EagleView, CityGRID, etc.
Detailed Reconstructing Approach	Manual	Creating each detailed building model individually and Entering architectural specifications manually	Building component libraries	A very high degree of detail and accuracy, very mature and widely used	Extremely time-consuming, labour-intensive, can only model of individual object, not possible modeling whole city	When representing a city's important buildings with historic value or unique architecture	Common modeling tools: SketchUp, 3DSMax, CAD softwares
Procedural Building Modeling Approach	Semi-Automatic	Creating robust 3D building models with rule files	Information contained in existing GIS databases, topographic data, existing maps, basemaps, land use/cover data, BIM, CAD, AM/FM	Very fast, considerable time and expense can be saved creating the building models, automatic mass modelling of 3D city models, enabling query and analysis of the model, quality of data	Block models of buildings, not visually attractive as the photo-realistic variety, the detail of roofs cannot be modelled	Satisfactory for applications that do not need high accuracy and many details, such as roofs	GIS softwares: ESRI's ArcGIS, ESRI's CityEngine, QGIS, FME, etc.

Table 1. A variety of object reconstruction approaches

Aerial photogrammetry, satellite imagery, and LiDAR are often the primary sources for creating building models and digital terrain models. Other common source of data is information from various types of terrestrial measurements, GNSS, 2D GIS databases, unmanned aerial vehicles/drones and mobile mapping. Geographic information generated by citizens, VGI, has emerged as an additional information source.

3.2 3D Object Reconstruction

Various approaches can be used for creating 3D models/3D urban maps (including building models, terrain models and landscape models) based on different data sources at different scale and qualities. Table 1 summarizes these diverse methods for 3D city object reconstruction.

The approach to use will greatly depend on the requirements of the application and how they will be utilized, the type and quality of the input datasets available, the LOD desired (ESRI, 2014). Depending on the accuracy and resolution of the desired 3D urban map, a combination of a variety of technical means can be used for the 3D digital city construction procedure to generate the map in an efficient, timely, and cost-effective manner.

4. CITY MODELING

4.1 City Modeling for Representation

A city in itself is very complex (Mao et al., 2012) since a 3D city is in general a vast collection of features, networks and surfaces (Reitz and Schubiger-Banz, 2014). To transform this huge amount of data into useful information and support future developments and applications, it is needed to structure 3D data into a geometric and semantic data model (Prandi et al., 2015). There are many approaches to model 3D city for the purpose of processing, analysis and visualization (Reitz and Schubiger-Banz, 2014). Two common information models to allow modelling of much of this complexity include OGC's CityGML as an open standard, and ESRI's 3D City Information Model (3DCIM) as a proprietary city model format (ESRI, 2014). Although there are many possibilities, it is considered as prerequisite that all data must be converted to an information model of CityGML on the back-end since CityGML provides an open common platform to integrate city level information from different resources.

CityGML is a common information model for the representation of 3D urban objects (OGC, 2012). It is realized as an open data model and XML-based format for the storage and exchange of virtual 3D city (OGC, 2009) with all its appropriate information. It encodes the geometry, topology, semantics and appearance of 3D city objects (Rodrigues et al., 2013). It covers broad thematic fields of city objects: geometrical and topological aspects can be accurately described and linked with their semantic part (Prandi et al., 2015). 3D models incorporate a very important aspect in their visualization, namely appearance i.e. the observable characteristics of their surfaces (Kolbe, 2009).

Semantics and geometry modeling obeys to a coherent model with two hierarchies in which objects are linked, this way it is possible to query/analyze the city model either by thematic or geometrical object properties or both simultaneously (Kolbe, 2009).

CityGML extends the GML standard with semantic and appearance aspects of 3D city models and introduces the concept of Levels Of Detail (from LOD0 to LOD4) in which objects become more detailed, both geometric and thematically, while the LOD increases (Kolbe, 2009). Meanwhile, the CityGML files can contain multiple representations for each object in different LODs simultaneously and show the generalized objects over different scales (Mao et al., 2012) according to the needs of users. Different modes of representation of a same city (high detail when objects are close by and low detail when objects are further away) in a model improve efficiency of navigating through a model (Kofler, 1998, Pasman and Jansen 2002).

Although it is possible to render 3D views directly from CityGML a client may still have difficulty visualizing several of Gigabytes, which includes fully textured buildings models, a terrain model and aerial imagery. (Rodrigues et al., 2013, Gaillard et al., 2015, Mao et al., 2012, Prandi et al., 2015, Chatuverdi, 2014). Hence, CityGML models are essential to represent and analyze 3D city objects, but not to present or visualize 3D city models directly (Rodrigues et al., 2013, Mao et al., 2012). Additionally, the detailed and complex structure of CityGML based on heavy XML causes pressure for efficient visualization of CityGML files since reading XML based schemas is complex in JavaScript applications (Gaillard et al., 2015, Mao et al., 2012). In order to visualize CityGML data on the web, parsing methods capable of retrieve information from CityGML and recode it to a more presentation friendly format, namely X3D, JSON and KML/COLLADA, are used (Rodrigues et al., 2013, Chatuverdi, 2014) while maintaining the richness, in terms of semantic information, of models contained in CityGML files (Gaillard et al., 2015).

4.2 City Modeling for Presentation

3D city models can be built with procedural modeling based on existing 2D GIS vector data and terrain data. Procedural modeling is performed with semantic rules to automatically generate virtual 3D city models. This approach can shorten the modeling cycle, reduce modeling costs and combine with 2D GIS data perfectly without data conversion (Hu et al., 2013). Semantic rules of 3D modeling can be encoded by Computer Generated Architecture (CGA) Shape Grammar and are defined in a text file. Semantic rule file contains a series of definitions, based on the elements (such as shapes and textures), attributes and relationships, and operations (such as extrude and split), which make 2D graphics turn to 3D graphics. Therefore, the key step in procedural modeling process is the creation of the rule files (Hu et al., 2013).

3D web content can be used in different contexts and for various purposes. Hence, creation, modification and customization of interactive 3D web content in a semantic way to specific requirements of the different content consumers simplifies dissemination of content on the web. Mappings of scene-independent domain-specific ontologies (e.g. describing city models) to content definition ontologies (e.g. 3D computer graphics) enable semantic modeling of interactive 3D web content and on-demand generation of interactive 3D web content. The domain specific ontologies (OWL classes and properties) providing elements and properties that are equivalent to the components and properties specified in the semantic 3D content representation (OWL classes and properties). Semantic 3D content is encoded using the semantic web standards (RDF, RDFS and OWL) and this mapping can be achieved by using a sematic editor, e.g. Protégé. In such a way, on-demand 3D content generation can be implemented using domain ontologies by taking into account the meaning of content elements at different levels of abstraction. (Walczak and Flotynski, 2015)

The issue of extending GIS services with semantic web technologies, i.e. geospatial semantic web technology, is a promising application area for the construction of semantic web GIS. Although semantic annotations, geo-ontologies and reasoning have been utilized in numerous GIS applications, using ontologies and reasoning in 3D web GIS is unusual. The current situation of the 3D geospatial semantic web technologies prevent the use of GIS in smart city applications pervasively and efficiently.

4.3 Generalization of 3D city models

The CityGML models are often extremely detailed and should be generalized presentation (Mao et al., 2012). For the problem of handling several LODs in the visualization of city models, generalization process can be employed by converting the model from higher LoDs to lower LoDs (Mao et al., 2012). The different levels of details are stored in the multiple representation data structure and this structure supports real-time generalization for dynamic real-time visualization. In 3D city visualization, multiple representation of the city is essential to reduce the loading time of 3D models and to improve the visual efficiency (Mao et al., 2014).

5. CLIENT SIDE IMPLEMENTATION STANDARDS

5.1 City Modeling for Representation

HTML5 is an open standard format to develop platform-independent web applications (W3C, 2014). It involves several useful elements such as canvas, scalable vector graphics, geo-location, web workers and web sockets. HTML5 utilizes the canvas element for visualizing 3D contents. Web worker allows the scripts to run in background along with the processing at main page. This functionality improves the processing speed at client side. The web-based 3D GIS application is developed to be run on top of an HTML5 browser utilizing WebGL, which is an extension of HTML5 canvas element, without any need to install any third-party plugin. (Chatuverdi, 2014)

5.2 WebGL frameworks for 3D geospatial applications on the web

WebGL is a cross-browser, cross-platform, royalty-free web standard (by the Khronos group) for a low-level 3D graphics API (JavaScript API) based on OpenGL ES 2.0, exposed through the HTML5 Canvas element as Document Object Model interfaces (Khronos, 2014). It allows the programmer to access the GPU directly from the browser via JavaScript (Gaillard et al., 2015). The WebGL specification provides hardware accelerated 3D functionality on the web, which helps in improving the performance of the application while working with 3D contents/3D objects (Chatuverdi, 2014).

Numerous libraries offering high-level API on top of WebGL, low-level 3D graphics API, have appeared recently. (https://www.khronos.org/webgl/wiki/User_Contributions) in order to avoid complex low level programming and provide ease of development. Some of the important pre-defined WebGL libraries worth mentioning are: Three.js, Babylon.js, Scene.js, GLGE, and X3DOM.

However, the frameworks differ in their range of features, their way to define 3D content and their purpose for different applications. There is no visualization framework yet which fits all needs of the geospatial community. The selection of the right framework highly depends on the use case and a requirement analysis. (Kramer and Gutbell, 2015)

All these WebGL-based libraries with the help of CesiumJS API based on Web-GL enabled web browser which does not require the installation of any plugins may be used to render 3D city objects of CityGML file within HTML5 canvas element and perform various GIS analyses on the globe. The high level approach Cesium is a Javascript based virtual globe on the web utilizing WebGL (Cesium, 2016).

6. 3D VISUALIZATION FRAMEWORK

The emerging standards and technological advancements in CityGML, OGC services, HTML5, JavaScript, GeoJSON, WebGL and increasing capacity of client devices allow to generate robust 3D geospatial web applications. Based upon those, a web client can be developed to visualize the geometric and semantic information of the 3D city models. Addition to 3D city model visualization, querying and analyzing can be performed on the client side through a web-based virtual scene.

A web graphical interface can be implemented for improving representation, visualization, navigation, interaction and animation through large 3D models in 3D environments. It exploits virtual globe technology, such as CesiumJS, WebGL Earth, Open Web Globe, MapGL. These spinning globes can be accessed using the web browser without having to install any software locally.

The web graphical interface enables interaction with 3D objects and relate them to get information about features from GIS databases (Rodrigues et al., 2013). The client can select the 3D model geometries and retrieves the information on the storage environment by means of OGC web services, like WFS, WMS, 3DP, W3DS, deployed on the middleware. All the functionalities such as parsing of geometries and 3D analyses can be implemented by the client using JavaScript (Chatuverdi, 2014).

To make a realistic render, photorealistic textures, color, transparency and illumination conditions such as light, shading, reflection etc. can be added to the geometry of the 3D content. Additionally, 3D modelling of trees and waterbody is produced based on its location to visually enrich the final virtual 3D scene (Padsala and Coors, 2015).

In order to create a complete landscape the terrain data is combined with geo-referenced imagery such as orthophotos and topographic maps. Rendering high resolution terrain data with imagery is not easy task because of this large amount of data. Similarly, the original CityGML dataset is several of Gigabytes, which includes fully textured buildings models, a terrain model and aerial imagery. This is too much to be loaded and displayed at once in the web browser (Kramer and Gutbell, 2015).

In order to manage visualization of this large amount of data first the dataset must divided into rectangular grid of tiles. In the tile-based approach, tiles can be loaded into memory on-demand and discard when they were not needed anymore. Then, streaming algorithm must be implemented, which allows loading tiles depending on the current camera position. When the camera is moved in any direction the streaming algorithm checks if more tiles need to be loaded or if those that are now out of sight can be removed from the scene graph. (Kramer and Gutbell, 2015) The major advantage of expanding streaming approach to tiling process is that it allows progressive visualization, which means, only the area required to be visualized will be fetched from the server (Chatuverdi, 2014). This facilitates covering larger areas easily.

The biggest limitation to handle very large city models is GPU memory amount. Memory of the device fills up with the usage of voxels, triangles, polygons and surfaces. This means even though the streaming algorithm removes tiles from the scene graph when they are not visible anymore, they will not be removed from the browser's memory and the graphics card. This causes the memory to fill up rather quickly and finally makes the browser crash after a couple of minutes of browsing through the scene. (Kramer and Gutbell, 2015) Memory management must be implemented through low-level API that allows for low-level access to the 3D renderer, which frees resources when the algorithm removes tiles from the scene.

Furthermore, cache systems are implemented both at application and middleware level, textures are compressed, the resolution of textures (mostly aerial images) are reduced and lightweight formats, like C3D, are used to stream geometry data to the client in order to improve loading time and speed up GPU rendering process. (Prandi et al., 2015)

The texture of the element is a representation of exterior of the modeling object, which is saved as a regular formats, like JPEG or PNG, or an OBJ file (Hu et al., 2013). The usage of traditional image file formats on the GPU takes a lot of space on the VRAM since the GPU needs to decompress JPEG files in the VRAM in order to be able to use them. Since textures are such a big part of the data, different formats, such as DDS, ETC, or ASTC, can be used to solve the problem of limited texture memory. Such files are larger than JPEG files on the disk, but can be read by the GPU while compressed, saving a lot of graphical memory. (Gaillard et al., 2015)

3D city model visualization framework should support LOD levels (both for geometry and textures) to improve memory usage, loading times and performance. LOD switching is the practice of displaying different geometric representations of the same object at different times, less detailed representation when the object is far away; more detailed when it closer the observer. This effect is normally achieved by tiling approach. Each tile can have multiple LOD representations and can switch between these representations independently. (Beck, 2004) The web-based client enables streaming of 3D content for real-time rendering in the browser with the adaptation of multiple representations of dynamic LODs based on user viewpoint.

7. 3D CITY MODEL STORAGE

It is needed to choose a suitable storage environment for multiple representation structures of 3D city models to handle different layers and LODs. The CityGML dataset are produced traditionally as single files. Today, object-relational databases are the most used means to storage 3D model building information and these databases are extended with dedicated city model tools such as 3DCityDB and DB4Geo. The more recent approach for storing 3D information is NoSQL databases.

For the various spatial operations, firstly the geometric and the thematic characteristics of objects and their spatial relationships should be integrated in a database. However, current DBMSs do not support the organization and implementation of 3D objects in their geometrical models and topological models (Stoter and Zlatanova, 2003). Current trend is to develop specific ad hoc solutions for using 3D geo-information on top of the object-relational databases such as PostGIS and Oracle Spatial. Storage specific extensions for 3D city models, such as 3DCityDB and DB4Geo, facilitates complex modeling of the semantic part of the CityGML and also making specific queries (Mao et al., 2014, Prandi et al., 2015) by implementing CityGML schema in an SQL table schema (Kunde, 2013). In order to mature 3D GIS, a 3D geometrical model should be fully supported by DBMSs based on OGC specifications for 3D features, which still have to be completed (Stoter and Zlatanova, 2003).

The 3D city database is an open source geodatabase schema, which provides the functionality to store, manage and represent virtual 3D city models on top of a standard relational database (3DCityDB, 2016). By importing the CityGML to the 3D city database, all the city objects are stored in their respective tables (Chatuverdi, 2014). One of the main features of 3DCityDB is the possibility to export the geometric information of the city models in different format such as the KML/COLLADA or other optimized graphical formats, which is a format more suitable for visualization purpose if compared to CityGML (Prandi et al., 2015).

However, the CityGML schema is complex and leads to hundreds of tables in the database but most of these tables may not be used in most applications, and such a storage method has limitations for big data processing (Mao et al., 2014). For data intensive applications an alternative storage method is the combination of NoSQL databases and cloud computation (Mao et al., 2014). There's more than one type of NoSQL database and a large number of individual NoSQL DBMSes, some of the most popular ones, Hadoop/Hbase (Wide Column Store / Column Families), MangoDB (Document Store Database).

Cloud computation method such as Map-Reduce can be deployed on NoSQL database to increase the analysis speed. It is suitable for big data applications such as 3D city model generalization and visualization. MongoDB supports Map-Reduce analysis and geospatial indexes, which can speed up the 3D city model related spatial search. (Mao et al., 2014)

The proposed storage framework is needed to be capable of integrating geometric and thematic characteristics of objects and topological relations of objects, like adjacency, inclusion, overlapping etc.

8. 3D SPATIAL FUNCTIONS

3D city models can be queried interactively, according to some given conditions by the users, to find various different information about 3D city objects corresponding with the attribute data using topology.

3D city applications on the web can employ 3D analyses, like line-of-sight, flood modeling, noise distribution, air flow analysis, simulating urban growth scenarios.

In the context of applications related to 3D city models, (Moser et al., 2010) have categorized 3D GIS analysis in four broad aspects: Spread/flow analysis, 3D density, visibility analysis, proximity and overlay analysis. These analyses utilize 3D operations like 3D intersection, 3D difference, 3D buffer, 3D union on virtual 3D city models. (Moser et al., 2010, Prandi et al., 2015)

These on-the-fly 3D spatial analyses can be applied on 3D city objects directly on the client-side. Other functionalities, like updating or modifying 3D city objects by adding a new feature/object or deleting an existing feature/object, can be performed on the web.

9. CONCLUSION

GIS (or Urban Information System, Urban GIS) for smart cities should be an integrated cross-sectoral platform to collect, store, manage, analyze and visualize spatiotemporal information for sustainable urbanization. Despite of the all advancements of GIS in the management of the urban environments in 3D, GIS technology is still lacking of utilizing semantic web technologies and spatial intelligence in GIS solutions. As a result of this, the shortage prevents the transition from existing GIS to Smart City GIS. Consequently there is no any holistic geospatial solution to the smart cities.

REFERENCES

3DCityDB, 2016, “The CityGML Database: 3D City Database”, Available at: http://www.3dcitydb.org/3dcitydb/3dcitydbhomepage/ (1 June 2016).

Beck, M., 2004, “Real time Visualization of big 3D City Models”, ISPRS Vol. XXXIV-5/W10

Cesium, 2016. Available at: https://cesiumjs.org/ (1 June 2016)

Chatuverdi, K., 2014, “Web-based 3D Analysis and Visualization using HTML5 and WebGL”, Master Thesis, the Faculty of Geo-information Science and Earth Observation (ITC), University of Twente, The Netherlands

ESRI, 2014, “3D Urban Mapping: From Pretty Pictures to 3D GIS”, ESRI White Paper, December 2014

Gaillard, J., Vienne, A., Baume, R., Pedrinis, F., Peytavie, A., Gesquière, G., 2015, “Urban Data Visualisation in a web browser”, Web3D 2015 20th International Conference on 3D Web Technology, June 18-21, 2015, Heraklion, Crete, Greece

Hu, X., Liu, X., He, Z., Zhang, J., 2013, “Batch Modeling Of 3D City Based On Esri CityEngine”, IET International Conference on Smart and Sustainable City 2013 (ICSSC 2013), 19-20 August, China

Khronos, 2014, “WebGL Specification”, Version 1.0.3, Available at: https://www.khronos.org/webgl/ (1 June 2016).

Kofler, M., 1998, R-trees for the visualisation of large 3D GIS Database, Ph.D. thesis, Technical University, Graz, Austria, 1998

Kolbe, T.H., 2009, “Representing and exchanging 3D city models with CityGML,” in 3D Geo-Information Sciences, Eds: J. Lee and S. Zlatanova, Springer Berlin Heidelberg, 2009, Chapter 2, pp. 15–31.

Kunde, F., 2013. CityGML in PostGIS : portability, usage and performance analysis using the example of the 3D City Database of Berlin. Master Thesis conducted at the Department of Geography, University of Potsdam, 2013.

Krammer, M. and Gutbell, R., 2015, “A case study on 3D geospatial applications in the Web using state-of-the-art WebGL frameworks”, Web3D 2015 20th International Conference on 3D Web Technology, June 18-21, 2015, Heraklion, Crete, Greece

Mao, B., Ban, Y., Harrie,L., 2012, “A Framework For Generalization Of 3D City Models Based On CityGML And X3D”, ISPRS Commission VI, WG II/2, 2012

Mao, B., Harrie, L., Cao, J., Wu, Z., Shen, J., 2014, “NOSQL Based 3D City Model Management System”, ISPRS Commission VI, WG VI/4, Volume XL-4, ISPRS Technical Commission IV Symposium, 14-16 May, China

Moser, J., Albrecht, F., Kosar, B., 2010, “Beyond Visualisation – 3D GIS Analyses For Virtual City Models”, ISPRS Volume XXXVIII-4/W15, 5th International 3D GeoInfo Conference, November 3-4, 2010, Berlin, Germany

OGC, 2012, “OGC City Geography Markup Language (CityGML) Encoding Standard, Version 2.0. Available at: http://www.opengeospatial.org/standards/citygml (1 June 2016).

Padsala, R., Coors, V., 2015, “Conceptualizing, Managing and Developing: A Web Based 3D City Information Model for Urban Energy Demand Simulation”, 3rd Eurographics Workshop on Urban Data Modelling and Visualisation, November 23rd, 2015, Delft, the Netherlands

Pasman, W. and Jansen, F. W., 2002, “Scheduling Level of Detail with Guaranteed Quality and Cost”, Web3D 2002 7th International Conference on 3D Web Technology, February 24-28, 2002, Tempe, Arizona, United States

Prandi, F., Devigili, F., Soave, M., Di Staso, U., De Amicis, R., 2015, “3D web visualization of huge CityGML models”, ISPRS Commission III, WG III/5, Volume XL-3/W3, ISPRS Geospatial Week 2015, 28 September-03 October, 2015, France

Reitz, T., Schubiger-Banz, S., 2014, “The Esri 3D City Information Model”, IOP Conference Series Earth and Environmental Science 18(1)

Rodrigues, J.I.J., Figueiredo, M.J.G., Costa, C.P., 2013, “Web3DGIS for city models with CityGML and X3D”, 17th International Conference on Information Visualisation

Stoter, J.E., Zlatanova, S., 2003, “3D GIS, Where Are We Standing?”, ISPRS Joint Workshop on ‘Spatial, Temporal and multi-dimensional data modelling and analysis’, Québec, October, 2003

W3C, 2014, “HTML5 W3C Recommendation 28 October 2014”, Available at: https://www.w3.org/TR/html5/ (1 June 2016).

Walczak, K., Flotynski, J., 2015, "Semantic Query-based Generation of Customized 3D Scenes", Web3D 2015 20th International Conference on 3D Web Technology, June 18-21, 2015, Heraklion, Crete, Greece