Technische Universität München
Department of Civil, Geo and Environmental Engineering
Chair of Cartography
Prof. Dr.-Ing. Liqiu Meng

3D Thematic Mapping and Visualization in *CesiumJS* Virtual Globe

Dayanc Moradi

Master's Thesis

Duration: 15.06.2016 - 26.05.2017

Study Course: Cartography M.Sc.

Supervisor: Dr. Mathias Jahnke

2017

Declaration of Originality

I hereby declare that I am the sole author of this thesis report titled as: "**3D Thematic Mapping and Visualization in CesiumJS Virtual Globe**", which is compulsory to fulfill the requirement in order to obtain **"Master of Sciences"** degree in the field of **"Cartography"** from **"Technische Universität München"**. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work I have not used any sources other than those listed in the bibliography and identified as references. I further declare that I have not previously submitted any part of this thesis to acquire a degree or any other qualification at this University or any other institution.

München, 30 May 2017

Dayanc Moradi

Acknowledgements

Herewith I would like to express my greatest appreciations to all the staff in cartography group in all three universities: Technische Universität München, Technische Universität Wien and Technische Universität Dresden who organized this amazing course and made this wonderful journey happen for us.

Furthermore, I would like to express my sincere gratitude to my supervisor, Dr. Mathias Jahnke for his encouragement; who patiently supported me throughout all the phases of my master thesis and sincerely guided me through the technical problems and difficulties.

I would also like to express my special gratitude to the Prof. Dr.-Ing. Liqiu Meng and Prof. Georg Gartner for their inspiring attitudes; It was always a great honor for me to be a little part of their teams in Technische Universität München and Technische Universität Wien.

I would like to particularly thank our coordinator Juliane Cron from Technische Universität München for her kind and sincere help and support during our journey as students.

In addition, hereby it is necessary to appreciate the cooperation of the respondent staff of the 'Lehrstuhl für Geoinformatik, Ingenieurfakultät Bau Geo Umwelt, Technische Universität München', for their support in providing CityGML dataset of Munich.

Also, I would like to give a special thanks to Mr. Jannes Bolling from Technische Universität Berlin for his constructive consultants in technical implementation phase of the thesis project.

Finally, I offer my specially appreciations to all the participants in the questionnaire survey including my friends, for their time and their support in providing accurate and precise insights towards our questions and helping us to draw meaningful conclusions.

Abstract

Development and transmission of the OpenGL technology to the web environment and desktop browsers through introducing the WebGL concept, has drastically facilitated the process of transferring 3D animations to the final users. Online 3D virtual globes take advantage of WebGL technology to transfer 3D city data to the browsers and, process and visualize the transferred bulks of data by taking advantage of the user's personal computers and smartphones processing hardware. This shortly explains the general architecture of 3D virtual globe platforms e.g. Google Earth and Cesium.

Cartographers can take the most advantage of these developments by providing 3D interactive city models through uprising virtual globes. Due to the novelty of the 3D modeling concept; there are limited number of references regarding codification of design principles in 3D domain. These shortcomings challenge the first section of the thesis research: "*3D City Cartographic Modelling…*" which is classified under the cartographic design section. In the cartographic design segment, the concentration is on defining and classifying practical visual variables in a 3D environment and assigning them to the model in a meaningful fashion.

Second phase of the thesis research is concerned with "…*Visualization in CesiumJS Virtual Globe*", which covers the technical perquisites in order to transfer and visualize the models on the user side platforms. In order to do so, an empirical workflow pipeline to transfer the designed 3D models in a form which is understandable and processable by CesiumJS was designed.

For understanding the appropriateness of the designed models regarding our defined applications for 3D city models, a user evaluation questionnaire was conducted. The results of the evaluation admitted that conventional cartographic design principles are not (at least in the examined practice) directly applicable in a 3D environment. To the other words, the conventional design concepts do not necessarily affect the 3D model users in a similar fashion to the 2D environment. Despite the relative satisfaction towards the 3D models, most of the users stated that they won't prefer 3D city models over 2D depictions.

This thesis is one of the first master degree thesis studies for outlining and evaluating cartographic visual variables in 3D environment. The rapid advances in 3D technology (e.g. Virtual Reality and Augmented Reality) indicates the necessity to define of a developed and solid structure to embed cartographic design principle in 3D environment; and also introduces new challenges and questions for cartography experts and researchers.

Keywords: Cartographic design principles, 3D cartographic modelling, visual variable, 3D city models, virtual globes, Cesium, WebGL, navigation.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

3DCityDB	3 **D**imensional **C**ity **D**ata **B**ase
3GK4	**3-degree G**auss-**K**ruger **zone 4**
AEC	**A**rchitecture, **E**ngineering, **C**onstruction
API	**A**pplication **P**rogram **I**nterface
B3DM	**B**atched **Three D**imensional **M**odel
CAD	**C**omputer **A**ided **D**rawing
CityDB	**C**ity **D**ata**b**ase
CityGML	**C**ity **G**eography **M**arkup **L**anguage
COLLADA	**COLLA**borative **D**esign **A**ctivity
CZML	**Cesium L**anguage
DEM	**D**igital **E**levation **M**odel
DHDN	**D**eutsches **H**aupt **D**reiecks **N**etz
DOM	**D**igital **O**rtho **I**mage
DTM	**D**igital **T**errain **M**odel
EDA	**E**xploratory **D**ata **A**nalysis
EPSG	**E**uropean **P**etro**l**eum **S**urvey **G**roup
ESRI	**E**nvironmental **S**ystems **R**esearch **I**nstitute
ETH Zürich	**E**idgenössische **T**echnische **H**ochschule **Zürich**
FM	**F**acility **M**anagement
GDRMS	**G**eospatial **D**igital **R**ights **M**anagement **S**ystem
GeoJSON	**G**eographic **J**ava**S**cript **O**bject **N**otation
GIS	**G**eographic **I**nformation **S**ystem
GL	**G**raphics **L**ibrary
glb	**B**inary **gl**TF
glTF	**G**raphics **L**ibrary **T**ransmission **F**ormat
GML	**G**eography **M**arkup **L**anguage
GPU	**G**raphics **P**rocessing **U**nit
GUI	**G**raphical **U**ser **I**nterface
Gvis	**G**eographic **Vis**ualization
HTML	**H**yper**t**ext **M**arkup **L**anguage
HTTP	**H**yper**t**ext **T**ransfer **P**rotocol
ICRF	**I**nternational **C**elestial **R**eference **F**rame
JPEG	**J**oint **P**hotographic **E**xperts **G**roup
JSON	**J**ava**S**cript **O**bject **N**otation
KML	**K**eyhole **M**arkup **L**anguage
KMZ	**K**eyhole **M**arkup Language, **Z**ipped
LDBV	**L**andesamt für **D**igitalisierung**, B**reitband und **V**ermessung
LOD	**L**evel **o**f **D**etails
NASA	**N**ational **A**eronautics and **S**pace **A**dministration
NICTA	**N**ational **I**nformation **C**ommunications **T**echnology **A**ustralia
OBJ Model	**O**bject **F**ormat **M**odel
OGC	**O**pen **G**eospatial **C**onsortium

OpenGL	**Open G**raphics **L**ibrary
OS X	Macintosh **O**perating **S**ystem **X**
SDK	**S**oftware **D**evelopment **K**it
SQL	**S**tructured **Q**uery **L**anguage
texel	**Tex**ture **El**ement
TIFF	**T**agged **I**mage **F**ile **F**ormat
TIN	**T**riangulated **I**rregular **N**etwork
UML	**U**nified **M**odeling **L**anguage
URI	**U**niform **R**esource **I**dentifier
URL	**U**niversal **R**esource **L**ocator
UTF-8	**U**nicode **T**ransformation **F**ormat **8**
UTM	**U**niversal **T**ransverse **M**ercator
WebGL	**Web G**raphics **L**ibrary
WFS	**Web F**eature **S**ervice
WGS84	**W**orld **G**eodetic **S**ystem **84**
WMS	**W**eb **M**ap **S**ervice
WMTS	**W**eb **M**ap **T**ile **S**ervice
WTS	**W**eb **T**errain **S**ervice
X3D	**Ex**tensible **Three D**imensional Format
XML	**Ex**tensible **M**arkup **L**anguage

To my Mother:

Anna

1 Introduction

1.1 Background and Previous Works

New means for communicating geo-information with audiences has always been attractive for geo-experts and cartographers. In parallel to the recent technological developments and appearance of new technologies, innovative formats for presenting geo-spatial data have been developed by mapping experts. Nowadays, the word 'map' dose not just predicates to the conventional inked and colored piece of paper; it also embraces concepts like multimedia and even virtual reality e.g. virtual 3D city tours.

In the last two decades, 3D city models due to their high potential for representing geo-spatial information has been considered as the best complement or even alternative for prevalent 2D paper maps. 3D city models because of their inherent resemblance to the reality and human's cognition to their surroundings, minimizes common errors and recognition and, enforcement which normally occurs in confronting conventional 2D maps. Defining additional dimension to the 2D maps and introducing 3D era in cartography and geo-visualization context, emphasizes the necessity to ordain new concepts for cartographic design in 3D environments.

Evolution of 3D city model have accelerated and their applications in the different fields of environmental sciences have expanded drastically in the recent years. Urban and environmental planning, energy management and demographics are just a sample of their vast application in different branches of science. This immerse potential of the 3D models enforced scientists to define new standards and frameworks for the storage and exchange of the 3D data. CityGML and standards defined by OGC are the result of these efforts to unite 3D models definitions and facilitating their conversion and exchange. In cartographic context most of the challenges in the conventional mapping applies in the 3D environment as well; challenges like human's perception limitations, cultural influences on the cognition and environmental conception, human's limitation in memorizing and processing colors and shapes and etc.

On the other hand, despite the common challenges in 2D and 3D cartographic visualizations, 3D modeling confronts mappers and designers to new challenges in designing and visualizing phases which need to be addressed in order to provide an unambiguous illustration for users that obviates their needs and tackles their questions in their desired context. Challenges in 3D domain can be classified under the software and hardware classes and 3D design rules and relations. The first classes of challenges mostly are related to other branches of science and engineering whilst design rules and concepts are discussed by mappers and cartography experts. Challenges like optimal screen resolution or optimizing the user's perception of 3D models by using different cartographic design criteria Has been discussed in several articles and journals. In addition, minimizing the size and the volume of the 3D data and the required processing power with innovative, as well as minimizing the required time and effort for designing realistic and reliable 3D models has recently considered as new challenge for cartographic community and scientific environments.

1.2 Problem Definition and Motivation

3Dcity models are rather a new concept in context of cartographic and semantic design. Comparing the existing comprehensive and admitted principles of cartographic design in the conventional 2D environments there are less numbers of specific and well-defined references for communicating visual information and semantics in 3D design procedure. One solution is to adapt existing principles to 3D environment. Graphical variables like size, color, texture and etc. are dimension independent and can be utilized in models in a 3D environment.

In addition to the possibility of using the conventional options, developing new options based on the 3D environment specialties and exclusives; edge enhancement and texture stylization of the 3D objects are some of the examples of graphical variables in 3D environment. Due to the novelty of the 3D concept, the existing hardware and software shortcomings and user's interaction limitations with 3D models, exclusive cartographic design rules for communicating semantics has less developed comparing to technical developments. One of the main reasons for conducting this research topic in can be addressed with the mentioned shortcomings. In this thesis the main focus is to identify existing sematic design principle, modify (if necessary) and then utilize them for different applications of 3D city models. The main motivation for the thesis was to apply visual variables in 3D environment in novel fashion e.g. mixing two different visual variables; size and transparency and then evaluate their efficiency towards predefined applications.

Appearance and the fast development of the server based applications has accelerated the application of 3D visualizations in different branches of science for the purpose of services and analysis. This thesis answers the question of choosing the best and the most appropriate visual variables regarding the defined application; this question is considered as the main aim of the thesis research. CesiumJS is a JavaScript online platform for visualizing 2D and 3D geoinformation which provides wide options for developers to design their desired optimum interface for communicating the intended information with audiences. Deployment of this comprehensive online platform for visualizing the designed models was the motivation to conduct this thesis.

1.3 Structure of the Thesis

The main focus of the following master thesis research is toward assigning the appropriate cartographic design concepts to the 3D city models regarding their intended application, visualizing the designed models in cesium.js and at the end, evaluating the models with user experiences toward their conformation with their desired application.

In the second chapter a deep investigation throughout the relevant existing literature and journal articles is performed. Concepts including cartographic semantic design concepts in 2D and 3D, CityGML and CityDB, and Cesium.js will be discussed in details. In addition, importance and necessity for performing an evaluation for researches will be considered as a critical part of literature review.

Chapter three is dedicated to codifying the required theories and explain the technical direction of the thesis to reach the meaningful results for further analysis. The required software tools for converting the acquired dataset, assigning the cartographic designs, and visualizing the results will be explained in this chapter. Technical necessities for performing a valid survey to achieve reliable and authentic results will be profoundly discussed at the final part of third chapter.

Defined theories in the chapter three will be practically implemented in the chapter four. This chapter contains detailed description of the performed conversion and designs for visualizing the resulted 3D models. The implemented procedure for importing the acquired dataset into a city database, exporting the 3D dataset, designing phase in Google's SketchUp and finally visualizing the resulted 3D city models on a browser through Cesium.js platform will be profoundly described. Also, the designed survey questionnaire will be shortly previewed in this section.

Fifth chapter is dedicated to investigate profoundly through the survey's outcomes. The results of the questionnaire will be statistically analyzed and comprehensively discussed based on different aspects and criteria. Finally, the summery of the master thesis research and future possible developments will be introduced and explained in the sixth chapter.

2 Literature Review and Previous Works

In this chapter a deep investigation throughout the relevant existing literature and journal articles is performed. Concepts including cartographic semantic design concepts in 2D and 3D environments, CityGML and CityDB, and Cesium.js will be discussed in details. In addition, importance and necessity for performing an evaluation for researches will be considered as a critical part of literature review.

2.1 Cartographic Visualization

In this sub-chapter definition of the term visualization and its meaning in the cartographic domain will be explained.

2.1.1 Definitions of Visualization

Wood (1994) believe that visualization is a mental process. It is a human ability to develop mental images, often of relationships that have no visible form (MacEachren & Ganter, 1990). Maps and other visual representations are valuable to science, not because of their realism, but because they are abstractions. The abstraction process, if successful, helps to distinguish pattern from noise. Visualization is a general term to denote the process of extracting data from the model and representing them on the screen (Zlatanova, 2000).

2.1.2 Cartographic Visualization

Maps have been a successful form of representation for centuries by making world understandable through systematic abstraction that retains the iconicity of space depicting space (MacEachren & Kraak, 2001).

In the past, paper maps were designed to be both database and presentation media; the advent of digital cartography and GIS, in the 1960s, split these tasks (MacEachren & Kraak, 2001). Over the past recent decades many different terminologies and definitions has been introduced for the concept that nowadays we call specifically “Cartographic Visualization”. As Hallisey (2005) has specified, genesis of this concept goes back to the early 50s; when “[…] Balchin and Coleman wrote their article on *graphicacy* notion”. The *graphicacy* can be defined as the ability to communicate types of information, relationships and patterns which cannot be handled effectively through words, numbers and art to the audience. Since than visualizations as a communication model became the dominant paradigm in cartography. For more than three decades by the early 90s cartographers focused on establishing the best possible visualization and communication tools for 2D depictions and optimizing the application them, and understanding the perceptual and cognitive issues involved in using these tools. However, since early 90s spatial data analysis added to the definition of the term ‘Cartographic Visualization’. Since than cartographic visualization doesn’t just refer to a communication model; it also helps us to search for the unknowns through high levels of human-map interaction.

As an exchange for the term cartographic visualization, Crampton (2001) employs the term geographic visualization (Gvis): “geographic visualization (Gvis) uses the map’s power to explore, analyze and visualize spatial datasets to understand patterns better". Kraak (2003) introduces the terminology "geovisualization" as "the use of visual geospatial displays to explore data and through that exploration to generate hypotheses, develop problem solutions and construct knowledge". The 'cartography as communication science' paradigm does not apply to cartographic visualization (MacEachren & Ganter, 1990). Treating the map as a visualization tool leads to a different perspective on cartography than that generally taken when the map is viewed as a communication device (MacEachren & Ganter, 1990). The significant problem for the latter definition prompts when we ask: If visualization includes both visual thinking and visual communication (as discussed in the previous paragraph), so what it does not include? Is "cartographic visualization" simply a new name for cartography? Saying that visualization involves computer graphics does not help much. It simply equates visualization with computer cartography (MacEachren & Taylor, 2013). Considering discussed definitions for cartographic visualization or visualization in realm of cartography in general, MacEarchen (2013) introduces the most comprehensive explanation for this concept (Figure 2-1). MacEachren (1990) believes restricting the concept of visualization to computer use (and computer sciences) obscure the long history of visualization in scientific advances. Longley (2001) represents a definition for scientific visualization which supports MacEachren's model for visualization: "Scientific visualization allows users to interpret, validate, and explore their data in greater detail than was possible hitherto".

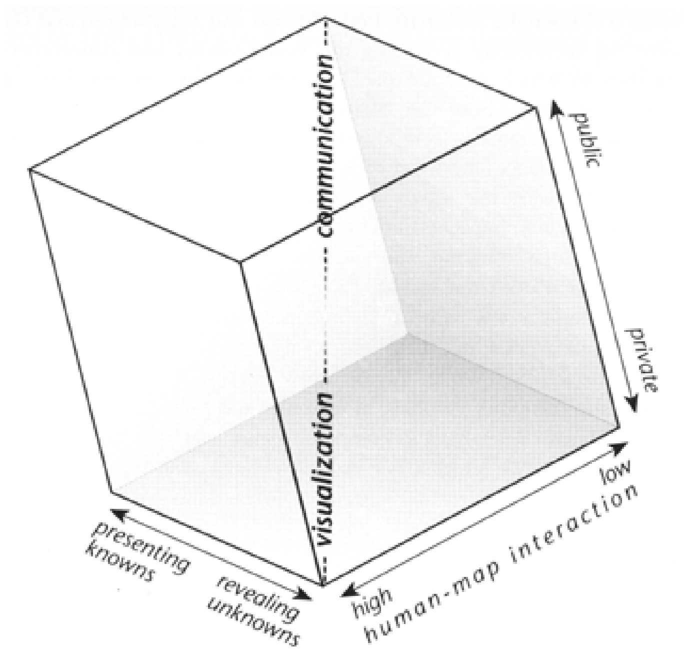


Figure 2-1: Cartography, a representation of the "space" of map use and the relative emphasis on visualization and communication at various locations within this space. This representation deals, not with kinds of maps, but with kinds of map use. Thus a particular category of map (e.g. a topographic map) might occupy any position within the space, depending upon what a user does with the map for what purpose (MacEachren & Taylor, 2013).

In MacEachern's (1994) introduced model, a symmetrical 3D cube indicates the concept of cartography as a metaphor, with three intersecting continua:

1. the continuum from private realm interaction where individuals explore spatial data to public realm interaction where a wide audience may use a published map,

2. the continuum from spatial data exploration in which the focus is revealing unknowns to map use in which ''knowns'' are presented, and
3. the continuum from high human-map interaction to low human- map interaction (Hallisey, 2005).

Beside MacEachern's discussed model for defining cartography as a concept and it's link to the visualization domain, Taylor (2013) suggests a conceptual basic model for cartography; which visualization still considered as central core for cartography (Figure 2-2). Taylor's (2013) approach presented visualization as the intersection of research on cognition, communication, and formalism (with the later implying strict adherence to rule structures dedicated by digital computer systems (MacEachren & Taylor, 2013, p. 4)).

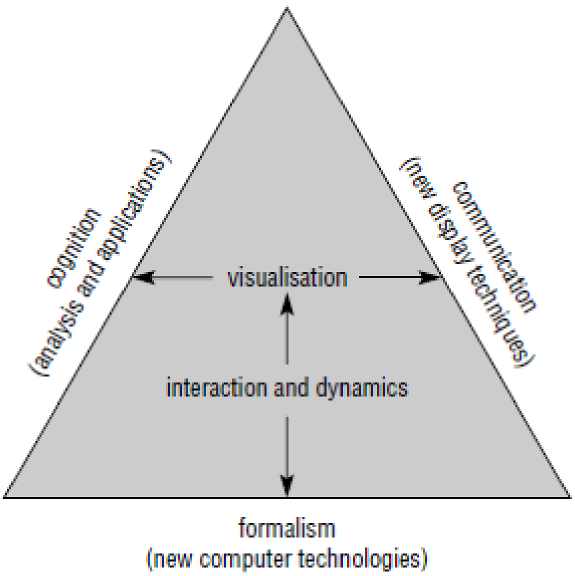


Figure 2-2: Taylor's „Conceptual basis“ triangular model for cartography (MacEachren & Taylor, 2013).

Taylor (2013) made it clear that he does not equate visualization with cartography. Instead Taylor (2013) argues for a view of visualization as a distinct development in cartography: "Visualization will never be all of cartography, but it will affect all three major aspects of cartography (these elements are: formalization or cartographic production techniques, communication and cognition and analysis) as shown on the diagram [figure 2-2] and will do so in an increasingly important way".

DiBiase borrowed from exploratory data analysis (EDA) literature of statistics to propose a graphic model of stages in map-based scientific visualization applied to the earth sciences (MacEachren & Kraak, 1997) (Figure 2-3). Exploratory Data Analysis (EDA) can be described as "*a willingness to look for what can be seen, whether or not anticipated*"; Revelation of patterns and anomalies through visual representations of data is the primary mode of exploratory analysis (DiBiase, 1990). DiBiase's (1990) model represents visualization as a four stage process consisting of two private visual thinking stages (exploration and confirmation) and two public visual communication stages (synthesis and presentation. It is noteworthy that the published maps we are used to seeing only represent the last of these four stages; data exploration, generation and confirmation (or disconfirmation) of hypotheses, and synthesis of these hypotheses are 'hidden' processes of map use (i.e., they are private), but which are nevertheless extremely critical (Crampton, 2001).

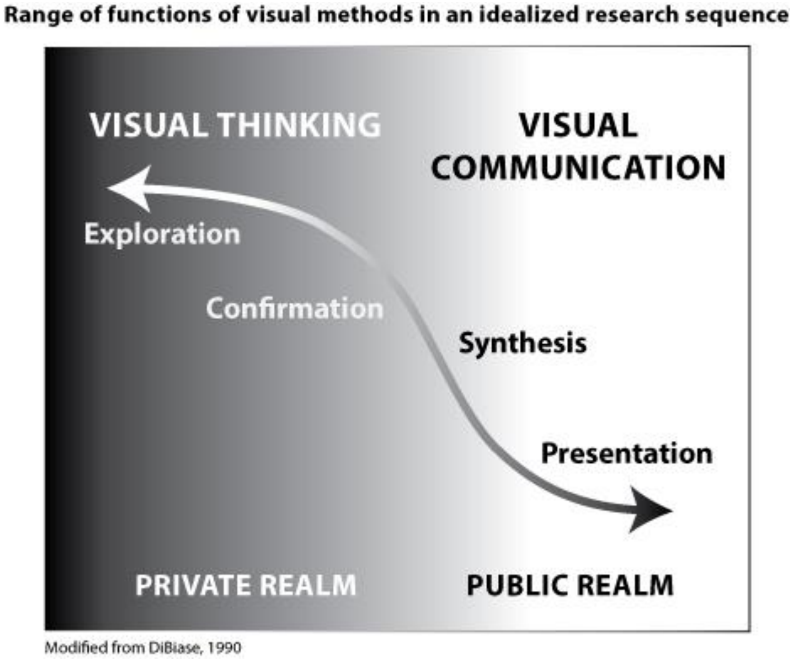


Figure 2-3: The range of functions of visual methods in an idealized research sequence (DiBiase, 1990).

DiBiase (1990) also explains the relevance of the visual thinking and visual communication realms intensions: "Visual thinking implies the generation of ideas through the creation, inspection, and interpretation of visual representations of the previously nonvisible, while visual communication refers to the effective distribution of ideas in visual form". An intent of the model was to encourage cartographers to direct attentions the role of maps at the early (private) stages of scientific research where maps and map-based tools are used to facilitate data sifting and exploration of extremely large data sets (MacEachren & Kraak, 1997).

Cartography occupies a critical position in the growing array of scientific visualization tools, particularly for geographers, earth scientists and atmospheric scientists.

Based on the discussed concepts in the previous paragraphs there is clear contradistinction between cartographic visualization and communication (as you can also see in the Fig. 2-1); the major conceptual difference is that in communicational visualizations the cartographer tries to deliver a *known* message whilst in visualization the message is *unknown* and analyst discovers conclusions while interacting with the model. MacEachren (1990) suggest the following approach for defining the cartographic visualization concept:

1. Visualization is a mental process. As such, it has existed for centuries. This fact seems to have been overlooked in the recent excitement about computer 'visualization:'
2. Computer graphics can *facilitate* visualization. Recent emphasis, however, has been solely on how to generate images, rather than on how images ma generate new ideas;

3. The goal of cartographic visualization (as with any form of scientific visualization) is to produce scientific insights by facilitating the identification of patterns, relationships, and anomalies in data;
4. Reconstruction of problems (looking at them from new perspective) is a key to insight;
5. Graphics designed simply to 'communicate' what we know are unlikely to foster the necessary new perspectives required to achieve insight about what is unknown.

The most important role of cartographic visualization, then, is in prompting mental visualization of spatial patterns and relationships with schematic bits and pieces of information. Cartographic visualization tools can, if properly designed, facilitate our abilities to notice (these) geographic patterns, relationships, etc. and to reason about what initially seems apparent. For cartographic visualization tools to succeed, interaction is a paramount. It is particularly critical to recognize that the visual display is not reality, but a *depiction* of a *presentation* of reality (MacEachren & Ganter, 1990).

After defining the term 'cartographic visualization' and discussing relation and dependency of cartography and visualization, it's time to define visualization functions and obligation in cartography domain. Kraak (1999) defines three major roles for visualization in contemporary cartography:

1. Present: visualization may be used to present spatial information. The results of spatial analysis operations can be displayed in well-designed maps easily understood by a wide audience. Questions such as 'what is?', or 'where is?', and what belongs together?' can be answered. The cartographic discipline offers design rules to help answer such questions through functions which create proper well-designed maps (Kraak & Ormeling, 2011).
2. Analyze: Second, visualization may be used to analyze, for instance in order to manipulate known data. In a planning environment the nature of two separate datasets can be fully understood, but not their relationship. A spatial analysis operation, such as (visual) overlay, combines both datasets to determine their possible spatial relationship. Questions like 'what is the best site?' or 'what is the shortest route?' can be answered. What is required are functions to access individual map components to extract information and functions to process, manipulate, or summarize that information (Bonham-Carter, 1996).
3. Explore: Third, visualization may be used to explore, for instance in order to play with unknown and often raw data. In several applications, such as those dealing with remote sensing data, there are abundant (temporal) data available. Questions like 'what is the nature of the dataset?', or 'which of those datasets reveal patterns related to the current problem studied?', and 'what if . . .?' have to be answered before the data can actually be used in a spatial analysis operation. Functions are required which allow the user to explore the spatial data visually (for instance by animation or by linked views – (MacEachren, 1995); (Peterson, 1995)).

Kraak (1999) declares that these three visualization strategies can be mapped into MacEachren's conceptual cartographic cube; including these strategies as a conceptual complementary spheres for introducing 'cartographic visualization' concept cube (Figure 2-4).

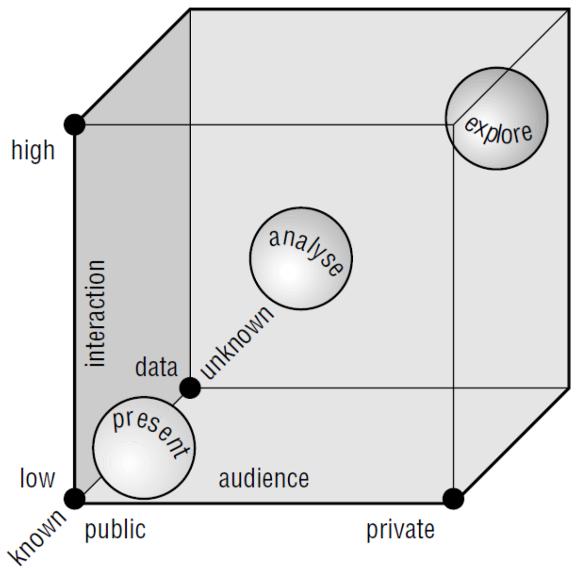


Figure 2-4: Cartographic visualization conceptual cube, three visualization strategies plotted in MacEachren's (1994) conceptual cartography cube (Kraak, 1999).

While the axes of MacEachren's cube indicates the nature of the depicted data for the user (known to unknown), the audience of the cartographic product (public to private) and the (possible) interactivity (low to high), three visualization strategies are represented as diagonal spheres replacing the terminologies visualization and communication in the original MacEachren's cube. The spheres representing the visualization strategies can be positioned along the diagonal from the lower left front corner (present: low interactivity, known data, and wide audience) to the upper right back corner (explore: high interactivity, unknown data, private person)(Kraak, 1999). Kraak (1999) also emphasizes each one of these visualization (so called) strategies requires its own visualization approach.

The technological, scientific, and social environment in which cartography operates, and in which maps are produced and used, has dramatically changed over the past two decades(MacEachren & Kraak, 2001). Recent developments in software and hardware infrastructures has caused fundamental changes in acquisition, management, analysis, and cartographic visualization of the georeferenced datasets. Undoubtedly MacEachren (2001) represent the best evolution process of cartographic visualization since the genesis of the first communication paper maps till the emergence of so called modern cartography terminology: "*In the past, paper maps were designed to be both database and presentation media. The advent of digital cartography and GIS, in the 1960s, split these tasks. Recent developments are making it possible to rejoin data storage with display, in ways never before possible. The map is now an interface that (if well designed) can support productive information access and knowledge construction activities (while it retains its traditional role as a presentation device). In modern map-based environments, the map can literally use the World Wide Web as its 'database'*.".

The recent technological improvements in visualization as the core concept of modern cartography has been resulted emerging the concept of "dynamic cartography"; in which provides almost real time, high quality visual illustrations and interactive manipulation of the geospatial information for the analysts or moderate users.

Nowadays modern cartography is characterized by two key component in order to support the emerging tasks including information exploration and synthesis, knowledge construction and goal-driven analysis: interaction (interactivity in human-computer domain) and dynamics (e.g. map objects behavior) (modern) cartography has much to offer the scientific community through its long history of design and production of visual representations of the Earth; on the other hand, cartography (still) has much to gain from collaborations with wider scientific visualization community where approaches to interactive computer tool development, interface design (human-computer interaction doctrine), three-dimensional computer modelling (e.g. 3D city modelling), and related methods and technologies are more fully developed (MacEachren & Kraak, 1997).

In modern cartographic visualization paradigm, as discussed, the message is unknown and therefore, there is no optimal map. The goal is to assist an analyst in *discovering patterns and relationships* in the data (MacEachren & Ganter, 1990). MacEachren (1990) outlines approaches to cartographic visualization based on the following tenets:

1. Visualization is a mental process. As such, it has existed for centuries. This fact seems to have been overlooked in the recent excitement about 'computer visualization';
2. Computer graphics can *facilitate* visualization. Recent emphasis, however, has been solely on how to generate images, rather than on how images may generate new ideas;
3. The goal of cartographic visualization (as with any form of scientific visualization) is to produce scientific insights by facilitating the identification of patterns, relationships, and anomalies in data;
4. Restructuring of problems (looking at them from new perspectives) is a key to insight;
5. Graphics designed simply to 'communicate' what we know are unlikely to foster the necessary new perspectives required to achieve insight about what is unknown.

2.1.2.1 2D Cartographic Visual Variables

As soon as a compact, yet attractive presentation is required, perceptional, cognitive and graphical design issues need to be considered (Semmo, 2012). Jacques Bertin (1983) as the leading theorists in the field of cartography, in his famous reference book "Semiology of Graphics: diagrams, networks, maps" indicates: Each cartographic representation - as a kind of graphical representations - is supported by a system of signs (symbols). Humans are good at interpreting visual data – much more so than interpreting numbers, for example – but conventions are still necessary to convey the message that the map-maker wants the data to impart (P. Longley, 2005). Conventions in the latter sentence of Longley which implies to the Bertin's term 'system of signs', has profoundly discussed in different cartographic references and journal articles under the process of attribute mapping and assigning appropriate visual variables to the data.

For an efficient communication, it requires having an adequate representation on feature level (Semmo, 2012). Graphical variables stimulate a certain perceptual behavior with the map user

(Kraak, 1999). Many of these conventions relate to use of symbols and colors (blue for rivers, green for forested areas, etc.), and have been developed over the past few hundred years (P. Longley, 2005). The data that have to be visualized will always refer to objects or phenomena in reality (Kraak & Ormeling, 2011). Attribute mapping entails use of graphic symbols, which (in two dimensions) may be referenced by points (e.g., historic monuments and telecoms antennae), lines (e.g., roads and water pipes) or areas (e.g., forests and urban areas) (P. Longley, 2005). Despite Longley's (2005) mentioned concept of attribute mapping with points, lines and areas, Kraak (2011) indicates: In cartography we use dots, dashes and patches to represent the location and attribute data of point, line, area and volume objects. It's necessary to mention here that the definition of point, line and area objects, i.e. objects that refer to point, line and area locations, is a matter of scale: a line which represents a river would have to be exchanged for an area if the scale of the map would increase, the built-up area of a settlement would be rendered by a dot if the scale of the representation were to decrease enough (Kraak & Ormeling, 2011).

After the process of attribute mapping, assigned attributes will be designed with different graphical variables based on their inherent types. Basic point, line, and area symbols are modified in different ways in order to communicate different types of information. The ways in which these modifications take place adhere to cognitive principles and the accumulated experience of application implementations (P. Longley, 2005). The basics of these modifications initially introduced by Bertin (Bertin, 1983) and then partially extended by MacEachren (1994); primitives including arrangement, texture and focus has been introduced by MacEachren (Figure 2-5).

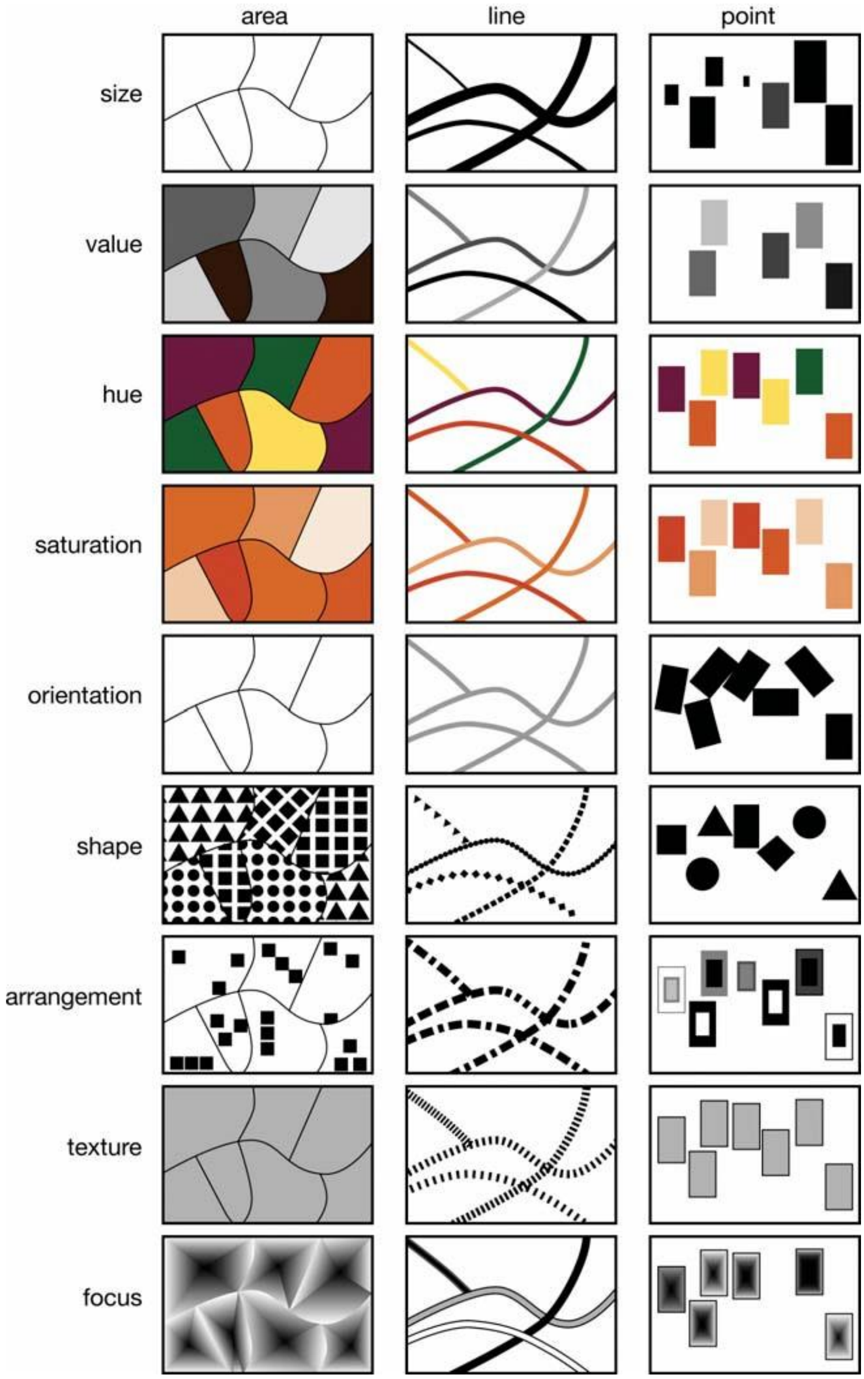


Figure 2-5: Bertin's graphic primitives, extended from seven to ten variables (the variable location is not depicted). Source: MacEachren 1994, from Visualization in Geographical Information Systems, Hearnshaw H.M. and Unwin D.J. (eds), Plate B (P. Longley, 2005).

Bertin's visual variables can be explicitly modified by changing (Häberling, 2003):

- Size — Variation of surface area
- Value — Variation of black and white parts
- Grain/Texture — Variations of the number of distinguishable spots per unit area
- Color — Variations of the color differentiations
- Orientation — Variations of the angular differences of fields with rectified character
- Shape — Variations of the same or different outlines

The added graphical variables can also be modified by changing their primitives. Each one of these visual variables has a certain length. Bertin (1983) identifies the length as the number of elements or categories that allows them (visual variables) to be identified. To the other words, the length of a variable corresponds to the number of possibilities (variations) that these variables can be assigned to (Häberling, 2003).

The selection of appropriate graphic variables to depict spatial locations and distributions presents one set of problems in mapping (P. Longley, 2005). In addition to the measurement scale, it is also important to take into account the distribution of the phenomenon, whether continuous or discontinuous, whether boundaries are smooth or not, and whether the data refer to point, line, area, or volume objects (Kraak, 1999). Longley suggests some of the common ways in which these graphic variables are used to visualize spatial object types and attributes in Table 2-1.

Table 2-1: Common methods of mapping spatial object types and attribute data with examples (2-D = two dimensional) (P. Longley, 2005).

Spatial object type	Attribute type		
	Nominal	Ordinal	Interval/Ratio
Point (0-D)	Symbol map	Hierarchy of symbols or lettering (color and size)	Graduate symbols (Color and size)
Line (1-D)	Network connectivity map (color, shape, orientation)	Graduated line symbology (color and size)	Flow map with width or color lines proportional to flows (color and size)
Area (2-D)	Unique category map (color, shape, orientation, pattern)	Graduated color or shading map	Continuous hue/shading
Surface (2.5-D)	One color per category (color, shape, orientation, pattern)	Ordered color map	Contour map

However, if all rules are applied mechanically the result can still be quite sterile and uninteresting; There is an additional need for a design that is appealing as well (Kraak, 1999).

2.1.2.2 3D Cartographic Visual Variables

The 3D environment is multiscalar and multipurpose; because 3D environments are believed to be more easily understood than either 2D article maps or 2D interactive representations, a range of possible users and applications is possible, depending on the datasets implemented (Crampton, 2001). The semantics-based, cartography-oriented visualization of virtual 3D city models represents a promising approach for improving the efficiency of 3D geoinformation communication (Semmo, 2012). Conventionally computer-generated perspective cartographic representations, and their creation processes are often cited under the subject of "map-related representations" (Häberling, 2003). Only Buchroithner (2001) defines the new explicit term, “3D Cartography”, he specifies it as:

> *"[…] the field of cartography, which includes classical three-dimensional physical map-related representations, pseudo-3D and the real three-dimensional cartographic representations"*.

Even though Buchroithner's definition considered as one of the first recorded definitions in the realm of 3D cartography, it leaves a wide range of representations unaddressed. The subject area of three-dimensional cartography must deal with both the analysis of geographic and spatial data as well as the fundamental considerations for creative and effective creation, and the perception processes for spatially perceptible spatial images (Häberling, 2003). Häberling (2003) defines the concept of "3D Map" as:

> *"As a "3D Map" we refer to a cartographic representation in a perspective oblique view with cartographic contents."*.

Häberling's (2003) definition of the "3D Map" is an inadequate interpretation since it does not address 3D environment specialties including dynamic cartographic visualization, possibility of dynamics, interaction with the objects (interactivity), selectable object or possible atmospheric effects in a 3D environment. Another perspective for defining 3D maps by Kraak (1988), considers neither of 3D maps specialties in defining it, rather his definition is based on the map user's perceptual behavior:

> *"[...] a map [...] is said to be three-dimensional when it contains stimuli which make the map user perceives its contents as three-dimensional*".

The most comprehensive definition of the "3D Map" expressed by Döllner (2001) which defines the term from a creative and user-oriented view:

> *"Interactive (dynamic) 3D maps can be understood as map-related representations that visualize space-related data and processes on the basis of a digital terrain model in three-dimension computer- graphically; And dynamically set the map composition and the map design depending on the visual situation, user preferences and user behavior."*.

Utilization of the 3D concept in the realm of modern cartography and the appearance of 3D cartographic products (e.g. 3D city models, 3D topographic maps and etc.) have not just changed the means of communication with the audience, but also has revolutionized cartographer's tasks and roles: "the modern 3D maps are less determined by the intuition and creativity of a painter or constructor (cartographer). Rather, they are dependent on the availability of (digital) data, powerful software functions, or special devices (e.g., monochromatic coherent light and photoplate in holograms)" (Häberling, 2003).

Despite profound literatures and research investigations regarding defining cartographic design principles in 2D environment, a well-defined and explicit theory for 3D design does not exist yet today; there is no reference book in cartographic design concept, explicitly defining 3D cartographic design as an independent well-stablished chapter. A part of this deficiency can be explained by the novelty of the 3D concept in realm of cartography, and in the world wide web domain. Bertin's (1983) hypothesis in visual variables deliberately draws its boundaries to the (two-dimensional) graphical system; nevertheless, he also recognizes visual variations for the recognition of the 3rd dimension by variables for depth recognition e.g. perspective introduced by Häberling (2003).

Map authors adopt usually the cartographic principles out of their experience with classic maps (Häberling, 2005). Absence of a solid and well-defined design principle in 3D environment results in appearance of map designer's personal preferences in the design process which eventually leads to inconsistency in 3D cartographic design environment. The cartographic design of 3D maps is affected by a large number of design variables; design variables – grouped in design aspects, which characterize specific fields in the design process (e.g., modeling of digital terrain model objects, appearance of objects, perspective, atmospheric effects and natural phenomena) - have an effect on the graphic appearance in the visualized 3D map (Häberling, 2005). A first trial for an inventory of design variables relevant for map concepts and production has been assembled by Häberling (2003). These 3D design variables are structured along the design process of 3D maps with the steps of *modelling, symbolization* and *visualization* (Table 2-2). At the same time, a 3D map is always affected by two basic aspects of design, the degree of *abstraction* and *dimension*, which both are a function of the map's objects and purpose (Häberling, 2005).

Table 2-2: Design aspects of modeling, symbolization and visualization (Haeberling, 2005).

Modeling

- ***Modeling of digital terrain model objects***
- ***Modeling of map objects***
- ***Modelling of orientating map objects***

Symbolization

- ***Graphic appearance***
- ***Special graphic aspects***
- ***Textures***
- ***Text objects***
- ***Object animation***

Visualization

- ***Perspective (projection)***
- ***Camera (viewing)***
- ***Lighting***
- ***Shading and shadow***
- ***Atmospheric effects and natural phenomenon***

In a research project at the Institute of Cartography, ETH Zurich some design variables for 3D maps were evaluated by Häberling (2003). Variables including *inclination angle* of the viewing direction and the *viewing distance* (*zoom factor*) both of the *camera* aspects, the *horizontal lighting direction* (from the *lighting* aspects) and the *sky structure* and *haze density* (both from the aspects of *atmospheric effects and natural phenomenon*) have been respectively designed and evaluated in 3D cartographic models. Accordingly, these variables can be listed as the result of one of the first attemps for defining 3D visual and variables in cartographic design environment. One of the most recent reasearches dedicated for defining visual variable in non-photorealistic 3D cartographic design environment has been conducted by Semmo (2012): "One possibility to visually emphasize important information is based on non-photorealistic rendering, which comprehends artistic depiction styles and is characterized by its expressiveness and communication aspects". On the other hand, non-photorealistic visualization in 3D environment results to sterile, monotonic and visually uninteresting depiction which is far from the essence of the cartography. Semmo's (2012) visualization concepts based on coupling non-photorealistic rendering techniques and semantics-based information for a user, context, and media-dependent representation of (intended) thematic information.

Semmo (2012) indicates: To enable an effective and efficient communication of geoinformation, in general, the used stylization needs to be adapted to the specific user, the user's task, the application, and the model. Considering the mentioned principles, Semmo (2012) classifies his cartographic stylization techniques in 3D environment as Table 2-3:

Table 2-3: Semmo's (2012) suggested Techniques for cartographic stylization and implementation.

3D cartographic stylization technique	**Definitions**
Colorization	• *In combination with an edge enhancement, color schemes can improve the perception of city structures.* • *Color and outline (edge enhancement) can be blended to stylize features of a specific class differently for highlighting important or prioritized information.*
Stylization of Object Textures	• *Performed resolution-dependently to enable a view-dependent level-of-abstraction.* • *A bilateral filter and a difference-of-Gaussian's- filter can be used for an automated abstraction of textures.*
Edge Enhancement of 3D Objects	• *Highlights structural aspects of virtual 3D city models by emphasizing and separating features located in the background of an image.* • *A texture-based edge enhancement is a well-known method to emphasize 3D objects and communicate uncertainty.*
Transparency of 3D Objects	• *Not a new approach in the context of virtual 3D city models.* • *Transparency effects are a well-known method to improve comprehension and visibility of occluded model entities in 3D-space.* • *Aids the perception of complex structures or the architecture of 3D building models.*

Real-Time Rendering	• *Which is essential for modern cartography's characteristics including dynamic stylization and interactivity.*

The identified approaches for a standardized visualization of cartography-oriented illustrations exemplify how high-quality geospatial visualization services can be technically implemented (Semmo, 2012). Semmo (2012) suggests to further improve the visualization quality:

- The implementation of additional typographical characteristics of cartography for certain feature types, for example to communicate land use information or visualize street networks in an abstract style; and,
- The incorporation of interactive concepts for a dynamic, contextaware abstraction, for example to enable a saliency-guided visualization.

Semmo also discusses the necesity of developing a suffieciently expressive and standardized style description language to promote technical progress, to make these stylizations broadly available and finally to improve the effectiveness and effieciency of (3D) geoinformation communication.

2.2 3D City Models

2.2.1 Principles

In the last recent decades, technological developments have revolutionized the conventional concepts of paper mapping and the way we interact with geospatial data and, defined new functions and applications for the realm of geovisualization. New technologies allow us to overcome certain limitations of traditional 2D maps and immerse into 3D space (Schilling et al., 2005). Online tools are dramatically changing the way we interact with spatial data, from 2D map to 3D virtual geographic environment; nowadays three-dimensional representations of geographic information, provide us more accurate and flexible mathematical models, effective tools and user interfaces for geospatial communication (Zhu et al., 2009).

The term 'model' is one of the most frequently used words in many disciplines. Scientists build and prove hypotheses, make predictions, exchange ideas and gain knowledge on the basis of models (Zlatanova, 2000). A universal model to comprise all the aspects of reality is not practically realizable due to the high complexity of the real world (Zlatanova, 2000).

City model implicates to an abstraction of the real city which comprises some of the 3D city's properties and attributes, interpreted to a digital (non-digital is also an option) form to be understandable for users, as well as computers for further investigations and analysis. 3D city modeling is an active research topic in distinct application areas. It's worthy to note that different modeling paradigms are employed in 3D geographical information systems, computer graphics, and architecture, engineering, construction, and facility management (Kolbe et al., 2005). Different disciplines (e.g. architecture, engineering, etc.) emphasize different aspects and only these aspects are included in the model; Thus a model considered good for the description of particular phenomena might be hardly appropriate for another's (Zlatanova, 2000).

A 3D city model is a representation of an urban environment with a three-dimensional geometry of common urban objects and structures, with buildings as the most prominent feature (Figure 2-6). Virtual 3D city models represent spatial and geo-referenced urban data by means of 3D geovirtual environments that basically include terrain models, building models, vegetation models as well as models of roads and transportation systems (Döllner, Baumann, et al., 2006). A typical 3D city model is derived from various acquisition techniques, for instance, photogrammetry and laser scanning, […], synthetic aperture radar, architectural models and drawings, handheld devices, […] and volunteered geoinformation (Biljecki et al., 2015).

Figure 2-6: Buildings as the prominent feature in a 3D city model.

3D city models are not just an abstracted photorealistic visualization model of the reality, but also they contain accurate and credible geometry, involved with organized attributes and semantics, and precise topological/geometrical information. Complex urban information spaces refer to virtual 3D city models integrating thematic and application-specific georeferenced information that is jointly presented and related to the geometric entities of virtual 3D city models (Döllner, Baumann, et al., 2006). Zhu (2009) classifies the *geometrical* hierarchy/complexity of the 3D city models in his research as follows:

- 2D digital ortho-image map (DOM), the texture of terrain surface, multiresolution DOMs from high resolution satellite imagery to aerial imagery provides the real world details and offers a time stamped record of land use, urban development and the general environment;
- 2.5D DEM, a digital representation of ground surface topography or terrain, is the most common foundation of 3D city models;
- 2.5D linear elements, the boundaries or centerlines of road, rivers, railways and land use, represent the outline of an object;
- 3D solid objects, external and/or internal surfaces of buildings, bridges overpasses, pipelines, stratum, etc., represent the entire volume of an object.

After defining the application and tenet of use for the city model we should specify the possible data sources to create a virtual 3D model based on the discussed hierarchy schema. In practice, the creation and maintenance of virtual 3D city models is based on a number of independent data sources since the sustainable management of 3D city models requires tight links to existing administrative work flows and databases (Döllner, Baumann, et al., 2006). As Döllner (2006) specifies possible data sources are:

- Cadastral Data
- Digital Terrain Models and Aerial Photography
- 3D Building Models
- Architectural Models
- Classical Georeferenced 2D Raster-data: as an auxiliary database, superimposed on the digital terrain mode.

2.2.2 3D City Model Theoretical Architecture

Döllner (2006) classifies a virtual 3D city model as a coherent system of functions (Figure 2-7) and suggests the following principles as components of his designed for virtual 3D model:

- 3D Authoring System: It is responsible for creating, editing, and versioning of the 3D city models and its components, e.g., importing, exporting, grouping, and annotating buildings, vegetation plans, landscape plans, etc. Technically, it provides an interactive access to the 3D geo-database
- 3D Geo-Database System: The database for storing and managing 3D city models. It also supports semantic and thematic properties, taxonomies and aggregations. Its principal object, the city object, represents geo-referenced, geometric entities. Specialized classes of city objects include buildings, green spaces, street spaces, transportation networks, water bodies, vegetation, and plants. It is implemented as an independent subsystem, and it does not provide visualization functionality
- 3D Editor Systems: These systems are responsible for creating and editing specific 3D objects such as architectural building models or 3D landscape models. Editor system supports a broad spectrum of digital 3D contents and fulfills the needs of specific applications and users with respect to 3D digital contents
- 3D Presentation Systems: The presentation systems provide real-time visualization of and interaction with the virtual 3D city model. In contrast to the 3D authoring system, the presentation systems are targeted at specific media (e.g. Internet and DVD) and specific user groups (e.g., general public, experts, and politicians). For example, within a showroom, a large screen projection can give impressive presentations tailored to the specific needs of clients based on pre-defined 3D points-of-interests
- Geospatial Digital Rights Management System: As a complementary functionality, a geospatial digital rights system allows for enclosing, compressing, and controlling digital contents of the virtual city model. Technically, a virtual city model can be serialized into a single data stream, compressed, and encrypted for export. In addition, a number of visualization techniques, such as adaptive visual watermarks and user interaction restrictions complement the DRM repertoire (Döllner, Kolbe, et al., 2006).

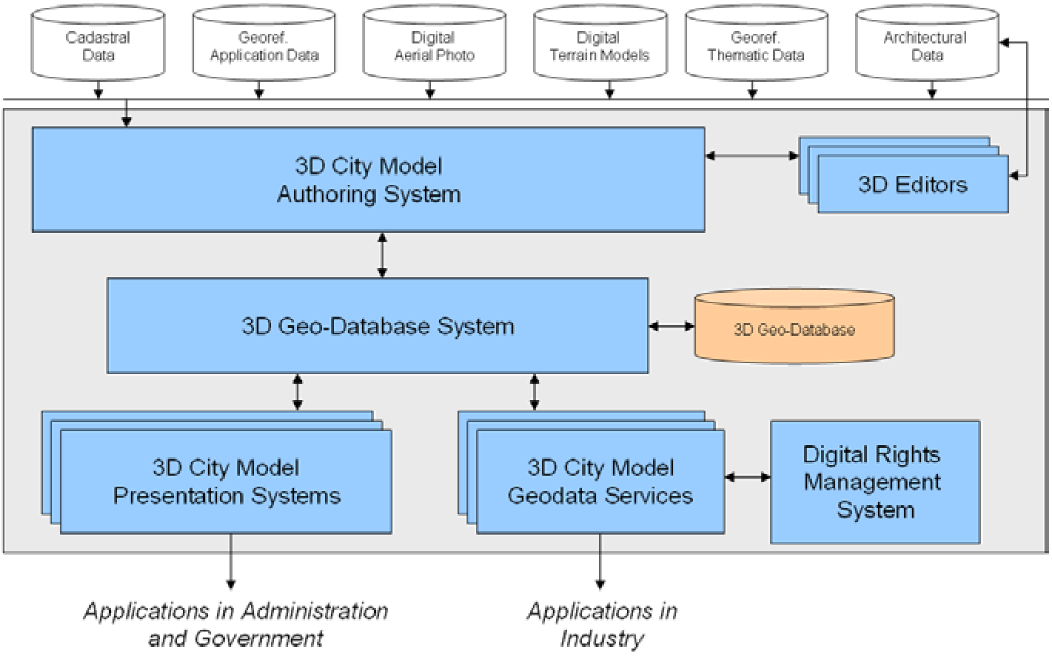


Figure 2-7: Architecture and principal components of a virtual 3D city model system (Döllner, Baumann, et al., 2006).

Based on the presented schema for 3D city model, cartographic design and visualization is a part of 3D editors' paradigm. Defining architecture, which exploits the concept of decoupling of the system's functionality into subsystems for content authoring, editing, storing and presentation, reminds us the necessity of an interoperable standard data format for an open, coherent, extensible and transparent geoinformation system. CityGML as the most developed and most efficient schema for 3D data storage and exchange will be profoundly discussed in section 2.2.4.

2.2.3 3D City Models Applications

3D city modeling is an active research topic in distinct application areas. Virtual 3D city models serve as an effective medium with manifold applications in geoinformation systems and services (Semmo, 2012). But nowadays, due to the technological developments and revolutionary advances in computer hardware and software, 3D city models have widespread applications in other scientific fields rather than being just a visualization tool for communicating 3D information to the users. Equipped with interactivity and visual dynamics, has introduced 3D city models as a reliable tool for information investigations and analysis.

3D city models have evolved to substantial tools for urban decision processes and information systems, especially in planning, simulation, documentation, heritage planning, mobile networks planning and navigation. Their power to support naïve geography is used by the wide public and results in easy understandable geospatial presentations. In contrary to traditional 2D maps with highly abstracted contents, map elements in 3D follow some natural/naive coding that can easily be accessed even by layman in map reading (Jobst & Döllner, 2008).

Virtual 3D city models provide important information for different aspects of disaster management (Kolbe et al., 2005). First, they memorize the shape and configuration of a city (Kolbe et al., 2005). In case of severe destruction of infrastructure e.g. caused by earthquakes, immediate access to this reference data allows to quickly assess the extent of the damage, to guide helpers and last but not least to rebuild the damaged sites (Kolbe et al., 2005). Second, 3D city models enable 3D visualizations and facilitate localization in indoor and outdoor navigation (Kolbe et al., 2005).

Different modeling paradigms are employed in 3D geographical information systems, computer graphics, and architecture, engineering, construction, and facility management. Whereas in 3D GIS the focus lies on the management of multi-scale, large area, and geo-referenced 3D models, the architecture, engineering, construction, and facility management domain addresses more detailed 3D models with respect to construction and management processes, computer graphics rather concentrates on the visual appearance of 3D models. The possible applications of a 3D city model mainly depend on the concrete development of the four distinct representation aspects:

- *geometry*
- *topology*
- *semantics*
- and *graphical appearance*.

Whereas geometry and topology describe the spatial configuration of 3D objects, the semantic aspect comprises the thematic structures, attributes and interrelationships. Information about the graphical appearance like façade textures, object colors, and signatures are employed for the visualization of the model (Kolbe et al., 2008). As the technology developed, 3D city models have become valuable for several purposes beyond visualization, and are utilized in a large number of domains (Figure 2-8) (Biljecki et al., 2015).

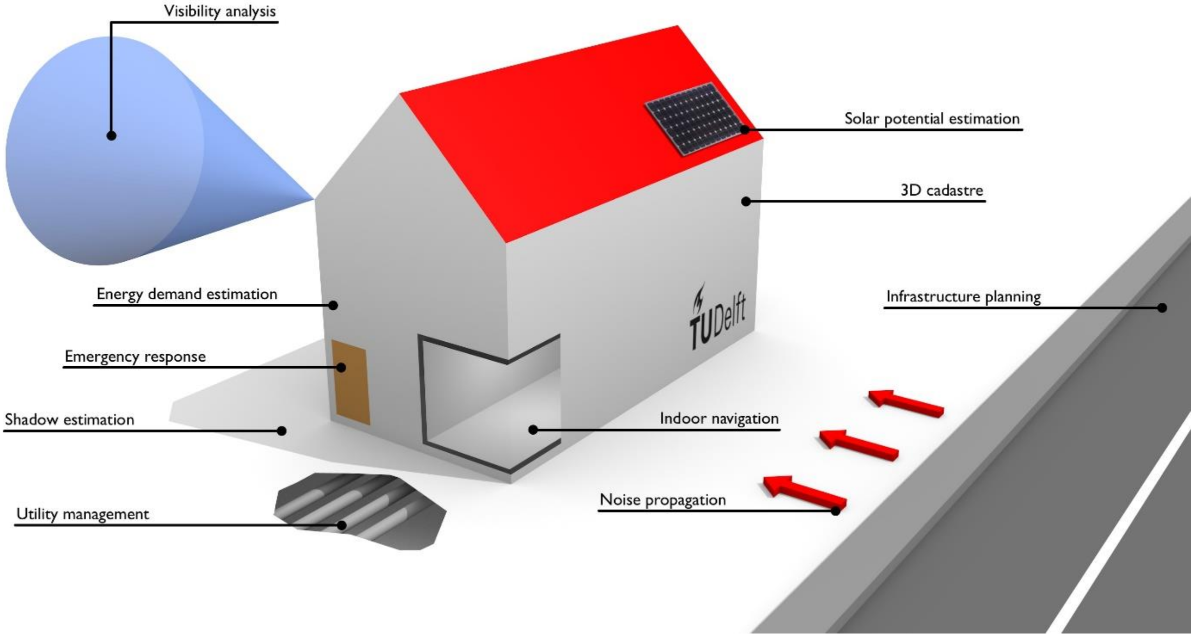


Figure 2-8: 3D city models may be applied in a multitude of application domains for environmental simulations and decision support (Biljecki et al., 2015).

One of the first attempts to identify and organize the use of 3D city models with focused on visualization and spatial planning have been performed by Betty (2000). They have classified the use of 3D models based on their application domain into 12 independent categories of industries: emergency services, urban planning, telecommunications, architecture, facilities and utilities management, marketing and economic development, property analysis, tourism and entertainment, e-commerce, environment, education and learning, and city portals.

Ross (2010) provides a general taxonomy of 3D city models use cases:

- Applications that are based only on geometry (e.g., estimation of the shadow)
- Analyses based on geometry and semantic information (e.g., estimation of the solar potential)
- Analyses based on domain specific extensions and external data (e.g., noise emission calculation).

This general classification despite being a straightforward theme for categorizing 3D city models, does not introduce an exclusive categorization for use cases. For example, it might be possible to categorize a specific application under two use cases.

Apart from mentioned applications which are mostly brief, specified for the paper's focus and are not necessarily always supported with references, Biljeki (2015) have performed a profound research regarding identifying and classifying different applications of 3D city models and designing an inventory for 3D city models applications. Biljeki (2015) claims "[...] *despite the near-ubiquitousness of 3D city models, a comprehensive inventory of 3D applications does not exist* […]". One of his major motivations for organizing such a comprehensive inventory was specifying and linking the required 3D data for specific applications, and providing a platform to identify the requirements for emerging different models across different domains to generate 3D data that is fit-for-purpose and last but not the least, contribute to identifying the eventual understanding of the model's fitness-for-use. Biljeki (2015) classify 3D city models based on their use cases; while each use case might have a several application domains e.g. tourism, architecture, etc. The reason for classifying based on the use case instead of application domain is to avoid developing additional parallel taxonomy.

However, the only criteria which Biljeki (2015) utilizes for classification of use cases is the visualization aspect:

1. Non-visualization use cases, which do not require visualizing the 3D models and the results of the 3D spatial operations. That is, the outcome of the spatial operation(s) can be stored in a database, e.g., solar potential of a roof surface, without the need of being visualized. The results *can be* visualized, but that is not essential to achieve the purpose of the use case, and it is not *essential* to visualize it in 3D (e.g., we can show the calculated information using color density instead)

2. Visualization-based use cases. This includes:
    - Use cases that require running computations as in the group 1., but where visualization is very important and the use cases would not make

much sense without it (e.g., navigation, serious gaming, and urban planning)

- Visualization-only use cases such as communication of urban information and virtual reality, which do not necessarily rely on spatial operations, but where 3D city models have been found as an important component (Biljecki et al., 2015).

Based on the discussed principles, Biljeki (2015) identify and organize 29 distinct use cases used in several application domains and provides examples for each use case (Table 2-4).

Table 2-4: Overview of the documented use cases of 3D city models, divided into two groups: non-visualisation(1.1 till 1.5) and visualisation (2.1 till 2.24) use cases (Biljecki et al., 2015).

§	Use case	Example of an application
1.1	Estimation of the solar irradiation	Determining the suitability of a roof surface for installing photovoltaic panels
1.2	Energy demand estimation	Assessing the return of a building energy retrofit
1.3	Aiding positioning	Map matching
1.4	Determination of the floorspace	Valuation of buildings
1.5	Classifying building types	Semantic enrichment of data sets
2.1	Geo-visualization and visualization enhancement	Flight simulation
2.2	Visibility analysis	Finding the optimal location to place a surveillance camera
2.3	Estimation of shadows cast by urban features	Determination of solar envelopes
2.4	Estimation of the propagation of noise in an urban environment	Traffic planning
2.5	3D cadastre	Property registration
2.6	Visualization for navigation	Navigation
2.7	Urban planning	Designing green areas
2.8	Visualization for communication of urban information to citizenry	Virtual tours
2.9	Reconstruction of sunlight direction	Object recognition
2.10	Understanding SAR images	Interpretation of radar data
2.11	Facility management	Managing utilities
2.12	Automatic scaffold assembly	Civil engineering
2.13	Emergency response	Planning evacuation
2.14	Lighting simulations	Planning lighting of landmarks
2.15	Radio-wave propagation	Optimizing radio infrastructure
2.16	Computational fluid dynamics	Predicting air quality
2.17	Estimating the population in an area	Crisis management
2.18	Routing	Understanding Accessibility
2.19	Forecasting seismic damage	Insurance
2.20	Flooding	Mitigating damage to utility management
2.21	Change detection	Urban inventory
2.22	Volumetric density studies	Urban studies

2.23	Forest management	Predicting tree growth
2.24	Archaeology	Visualizing ancient sites

The group of use cases relying on visualisation is larger than the other one, indicating that visualisation is an inseparable part of the workflows involving 3D city models (Biljecki et al., 2015). In addition to this classification, Biljecki (2015) analysis has revealed intersting patterns: "[…] *For instance, the development and utilization of some use cases appear to be more popular in some countries than others, e.g., solar studies are encountered mostly in papers published by authors from Germany*."; furthermore "[…] *it seems that there is no strong relation between the actual usage of a use case and the quantity of research papers describing it. As an example, navigation is arguably one of the most prominent use cases with a high usage share, but the topic of a few research papers*".

Quantity of the mentioned use cases proves the valuable role of 3D city models and indicates uprising demand for them. Due to the recent advances in fields including augmented reality, virtual reality, computer graphics fusion, appearance of new use cases for 3D models seems to be imminent. Focus of the thesis is on 3D cartographic designs variables for use cases tourism and visualization for communication of urban information to citizenry; considering Biljecki's (2015) introduced use case taxonomy (see Table 2-4).

2.2.4 The Semantics of CityGML

As discussed at the end of the section 2.2.2, 3D city model theoretical architecture, the concept of decoupling of the 3D system's functionality into subsystems for content authoring, editing, storing and presentation, reminds us the necessity of an interoperable standard *data format* for an open, coherent, extensible and transparent geoinformation system. To the other words, to achieve an efficient and flexible 3D city system, the prerequisites including syntactic and semantic interoperability of the participating GIS components are of utmost importance.

Syntactic interoperability can be achieved by using the XML-based Geography Markup Language (GML3) of the Open Geospatial Consortium (OGC); GML3 is an XML-based abstract format for the concrete specification of application specific spatial data formats (Kolbe et al., 2005). It is open, vendor-independent, and based on ISO standards; it can be extended and specialized to a specific application domain; and it explicitly supports simple and complex 3D geometry and topology (Kolbe et al., 2005). Furthermore, GML is the native data format of OGC's Web Feature Service (WFS), a standardized web service that implements methods to access and manipulate geodata within a spatial data infrastructure (Cox et al., 2002).

Semantic interoperability presumes common definitions of objects, attributes, and their interrelationships with respect to a specific domain (Kolbe et al., 2005). Based on the mentioned prerequisites, CityGML, a multi-purpose and multi-scale representation, and as an interoperable syntactic and semantic schema for data storage and exchange in the 3D city models environment has been introduced. CityGML defines a common information model for cities and regions, including their semantic properties, i.e. the generalization hierarchy between classes, relations between objects, and spatial properties as well the appearance of objects (Kolbe et al., 2008). CityGML covers the geometrical, topological, and semantic aspects of 3D city models and supports interoperability, consistency and functionality (Kolbe et al., 2005).

The CityGML core module defines the basic concepts and components of the CityGML data model (see figure 2-10). It is to be seen as the universal lower bound of the overall CityGML data model and a dependency of all thematic extension modules (Gröger et al., 2008), to the other words, core module should be considered as the backbone of CityGML data schema for any other extension modules to assure consistency in 3D data structures. The following eleven thematic extension modules are introduced by version 1.0 of the CityGML standard. Gröger (2008) indicates that the thematic extension modules are directly related to clauses of this document each covering the corresponding thematic field of CityGML:

- CityGML Core: Core module defines the basic components of the CityGML data model.
    - Appearance: Provides the means to model appearances of CityGML features,
    - Building: Representation of thematic and spatial aspects of buildings, building parts, building installations and interior building structures in four levels of detail,
    - CityFurniture: Represents immovable objects like lanterns, traffic signs, advertising columns, benches, etc.,
    - CityObjectGroup: Arbitrary city objects may be aggregated in groups according to user-defined criteria to represent and transfer these aggregations as part of the city model,
    - Generics: Used to model and exchange additional attributes and features not covered by the predefined thematic classes of CityGML.
    - LandUse: Representation of areas of the earth's surface dedicated to a specific land use,
    - Relief: Representation of the terrain in a city model; regular raster or grid, as a TIN, by break lines, and by mass points,
    - Transportation: Represents the transportation features within a city, for example roads, tracks, railways, or squares,
    - Vegetation: Provides thematic classes to represent vegetation objects.
    - WaterBody: Represents the thematic aspects and 3D geometry of rivers, canals, lakes, and basins,
    - TexturedSurface: Allows for assigning visual appearance properties (color, shininess, transparency) and textures to 3D surfaces.

.

Spatial properties of 3D city models e.g. buildings, store in CityGML database under the geometry model schema which is a subset of GML3's geometry package which is based on *Boundary Representation* concept (Hughes et al., 2014). The geometry model of GML 3 consists of primitives, which may be combined to form complexes, composite geometries or aggregates. For each dimension, there is a geometrical primitive: a zero-dimensional object is a *Point*, a one-dimensional a *_Curve*, a two-dimensional a *_Surface*, and a three-dimensional a *_Solid* (Gröger et al., 2008) (Figure 2-9). Hence combined geometries can be aggregates, complexes or composites of primitives.

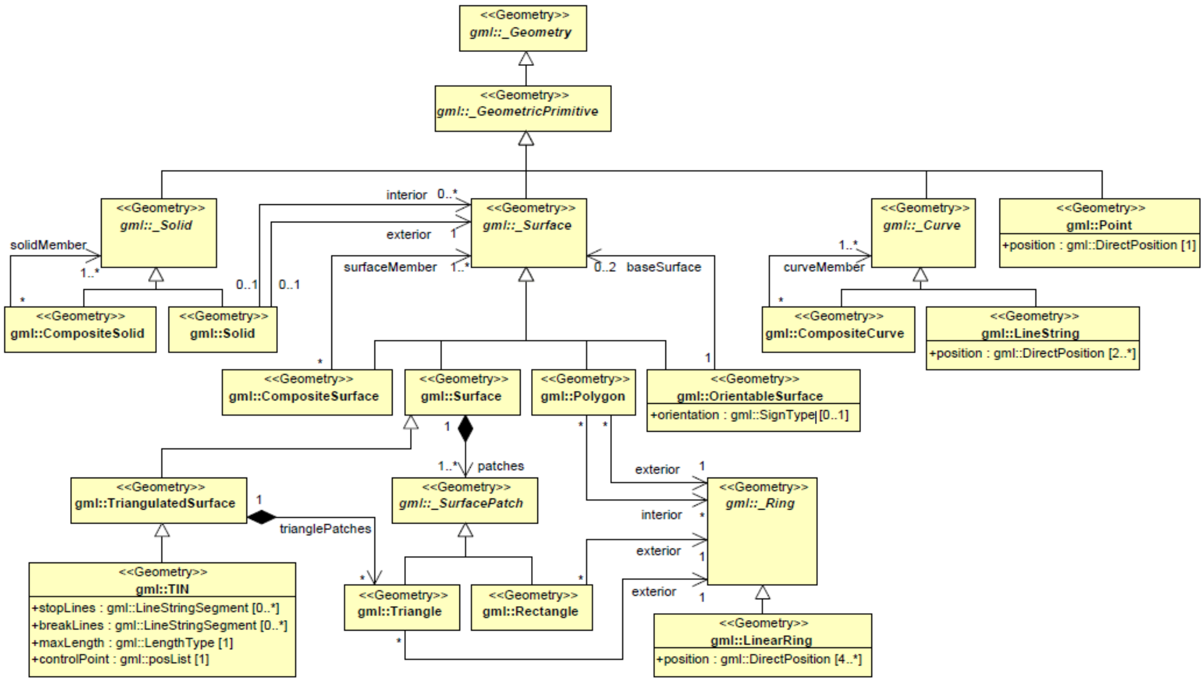


Figure 2-9: UML diagram of CityGML's geometry model (subset and profile of GML3): Primitives and Composites (Gröger et al., 2008).

In the thesis, CityGML is considered as the source format for 3D portraying. The semantic information contained in the model can be used in the styling process which generates computer graphics represented e.g. as KML/COLLADA or X3D files (Gröger et al., 2008). In the following sub-sections, the general concepts implemented by CityGML will be briefly presented.

2.2.4.1 Taxonomies and Aggregations

In CityGML encoding standard each object in 3D environment classifies under a certain taxonomy. On its thematic level CityGML defines classes and relations for the most relevant topographic objects in cities and regional models comprising built structures, elevation, vegetation, water bodies, city furniture, and more (Booch, 2005). CityGML thematic fields include:

- Digital Terrain Models as a combination of (including nested) triangulated irregular networks (TINs), regular rasters, break and skeleton lines, mass points;
- Sites (buildings, bridges and tunnels);
- Vegetation (areas, volumes and solitary objects with vegetation classification);
- Water bodies (volumes, surfaces);
- Transportation facilities (both graph structures and 3D surface data)
- Land use (representation of areas of the earth's surface dedicated to a specific land-use)
- City furniture
- Generic city objects and attributes
- User-definable (recursive) grouping

The dependency relations between CityGML's modules are illustrated in Figure 2-10 using Booch's (2005) UML (Unified Modeling Language) package diagram. This diagram gives an overview regarding different CityGML module components and their relationship and dependencies regarding CityGML's core module.

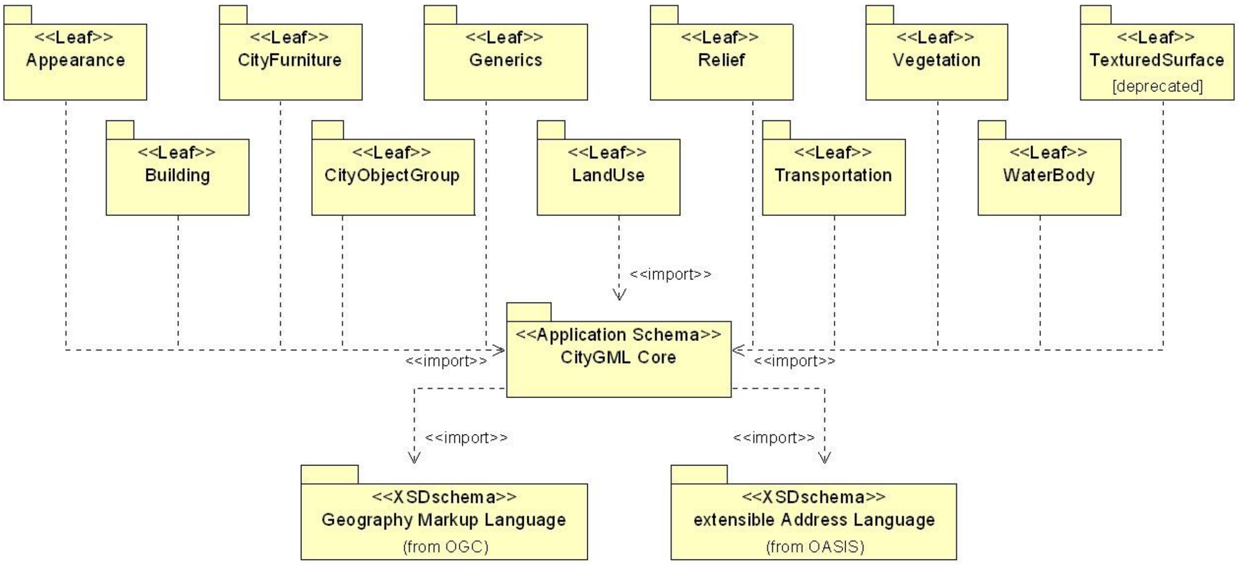


Figure 2-10: UML package diagram illustrating the separate modules of CityGML and their schema dependencies. Each extension module (indicated by the leaf packages) further imports the GML 3.1.1 schema definition in order to represent spatial properties of its thematic classes. For readability reasons, the corresponding dependencies have been omitted (Gröger et al., 2008).

A dashed arrow in the figure indicates that the schema at the tail of the arrow depends upon the schema at the head of the arrow (Booch, 2005).

2.2.4.2 Level of Details (LODs)

CityGML supports different Levels-of-Detail (LOD) which facilitates data interoperability, visualization and analysis. In a CityGML dataset, the same object may be represented in different LOD simultaneously, enabling the analysis and visualization of the same object with regard to different degrees of resolution (Kolbe et al., 2008). For instance, two different CityGML datasets containing the same object (e.g. a building) in different LODs can be combined and integrated as a more comprehensive dataset.

Kolbe (2008) provides five well-defined consecutive Level-of-Details for CityGML standard (Figure 2-11) (considering building as a simple example):

1. LOD0: The coarsest level, essentially a two and a half dimensional *Digital Terrain Model* (DTM)
2. LOD1: Is the well-known blocks model, without any roof structures
3. LOD2: Different buildings have distinctive roof structures
4. LOD3: Denotes architectural models with detailed wall and roof structures, balconies and bays
5. LOD4: Completes a LOD3 model by adding interior structures like rooms, interior doors, stairs, and furniture.

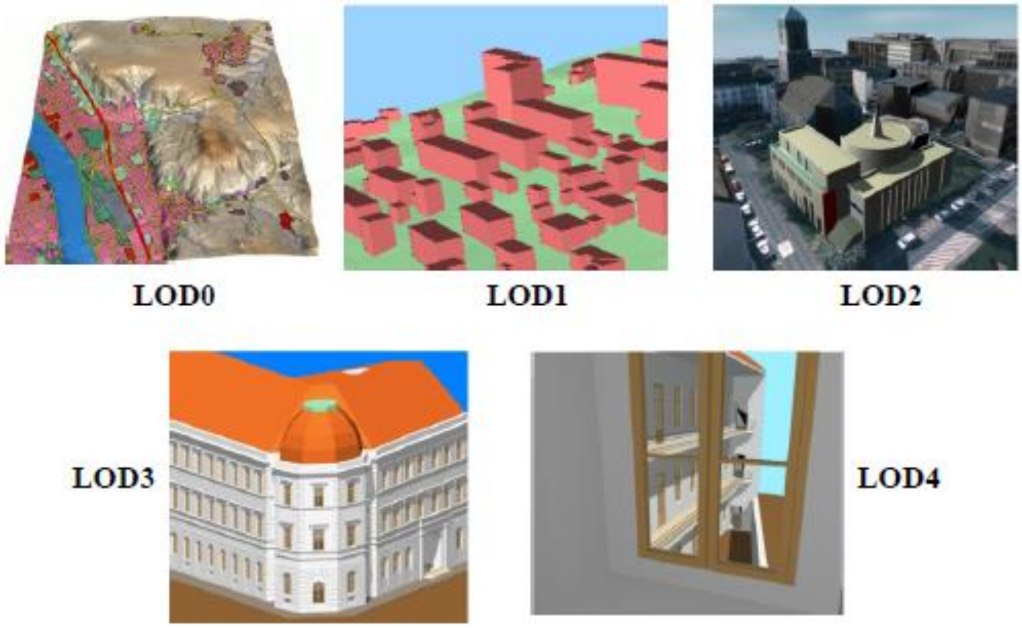


Figure 2-10: Illustration of the five Levels-of-Detail (LOD) defined by CityGML (Gröger et al., 2008).

LODs are also characterized by differing accuracies and minimal dimensions of objects (Table 2-5). The LOD categorization makes datasets comparable and provides support for their integration.

Table 2-5: LOD 0-4 of CityGML with its accuracy requirements (Gröger et al., 2008).

	LOD0	**LOD1**	**LOD2**	**LOD3**	**LOD4**
Model Scale Description	regional, landscape	city, region	city districts, projects	architectural models (outside), landmark	architectural models (interior)
Class of Accuracy	lowest	low	middle	high	very high
Absolute 3D point accuracy (position / height)	lower than LOD1	5/5m	2/2m	0.5/0.5m	0.2/0.2m
Generalization	maximal generalization (classification of land use)	object blocks as generalized features; > 6*6m/3m	objects as generalized features; > 4*4m/2m	object as real features; > 2*2m/1m	constructive elements and openings are represented
Building installations	-	-	-	representative exterior effects	real object form
Roof form/structure	no	flat	roof type and orientation	real object form	real object form
Roof overhanging parts	-	-	n.a.	n.a.	Yes
City Furniture	-	important objects	prototypes	real object form	real object form
Solitary Vegetation Object	-	important objects	prototypes, higher 6m	prototypes, higher 2m	prototypes, real object form
Plant cover	-	>50*50m	>5*5m	< LOD2	<LOD2

Whereas in CityGML each object can have a different representation for every LOD, often different objects from the same LOD will be generalized to be represented by an aggregate object in a lower LOD. CityGML supports the aggregation / decomposition by providing an explicit generalization association between any CityObjects (Gröger et al., 2008).

2.2.4.3 Geometrical Modeling

Spatial properties of the objects in CityGML are stored by subsets of geometrical primitives. For each dimension, there is a geometrical primitive: a zero-dimensional object is a point, a one-dimensional a curve, a two-dimensional a surface, and a three-dimensional a solid, whereas a solid is bounded by surfaces, a surface by curves, and a curve by points (Kolbe et al., 2008). Kolbe (2008) has specified some of the integrity constrains assure the consistency of the 3D model:

- A curve is restricted to be a straight line, and each surface must be planar
- Curves, surfaces and solids may be aggregated to *CompositeCurves*, *CompositeSurfaces* and *CompositeSolids*, respectively
- The interiors of the primitives must be disjoint. This constraint assure that primitives do not overlap, and touch at most at their boundaries.

In addition of geometrical constrains, the graphical appearance of the model's surfaces, is covered by concepts adopted from the graphics standard X3D (Kolbe et al., 2008).

2.2.4.4 Coherent Semantic-Geometrical Modeling

CityGML model consists of two hierarchies: the semantic and the geometrical, where the corresponding objects are linked by relations (Kolbe et al., 2008). On sematic level, real-world objects are represented by their features and their attributes, relationship and aggregation hierarchies; while on geometrical level, objects are assigned to features representing their spatial location and extent.

The advantage of this approach is, that it can be navigated in both hierarchies and between both hierarchies arbitrarily, for answering thematic and/or geometrical queries or performing analyses (Kolbe et al., 2008).

2.2.4.5 Closure Surfaces and Subsurface Objects

A new concept in CityGML is the *ClosureSurface*, which is employed to seal objects, which are in fact open, but must be closed in order to compute its volume (Figure 2-12). The concept of *ClosureSurfaces* also is employed to model the entrances of *subsurface objects*. Those objects like tunnels or pedestrian underpasses have to be modeled as closed solids in order to compute their volume, for example in flood simulations (Kolbe et al., 2008).

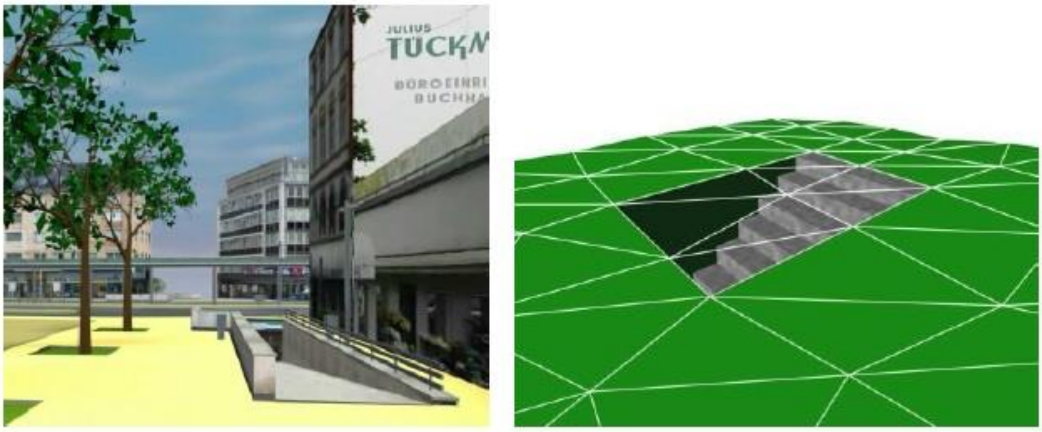


Figure 2-11: Passages are subsurface objects (left). The entrance is sealed by a virtual ClosureSurface, which is both part of the DTM and the subsurface object (right) (Gröger et al., 2008).

2.2.4.6 References to Objects in External Data Sets

Each CityGML thematic object may have external connections to the other datasets, tailoring some of its semantic or thematic properties. linking external dataset to achieve a coherent body of database is one of the CityGML specialties (Figure 2-13). The link between individual object's attributes in different datasets is achieved through the concept of *Uniform Resource Identifier (URI)*, which is a generic format for referencing to any kind of resources in the internet.

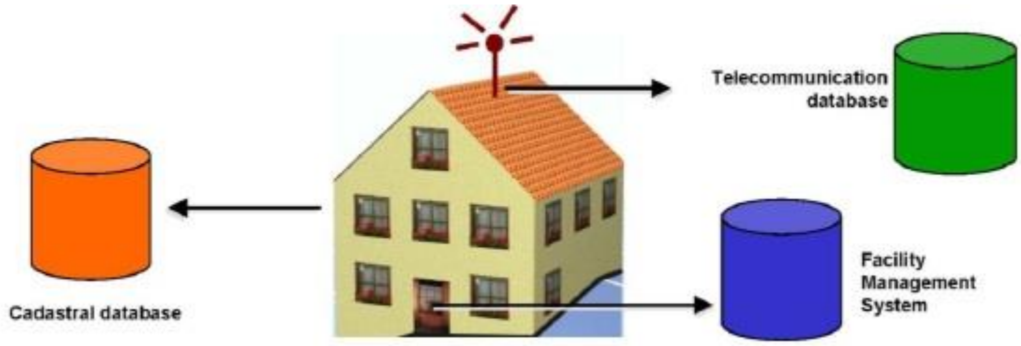


Figure 2-12: CityGML external references (Gröger et al., 2008).

2.2.4.7 Dictionaries and Code Lists for Attributes

GML3 *Code Lists* or *Dictionaries*, maps object's string attributes to the numerical values. Some of the advantages of this notion are:

- Translation of attribute values into other languages is facilitated;
- Code Lists can be extended or redefined by the users;
- Eliminates the possibility of misspelling of the same attribute in different data bases;
- Assures semantic and geometrical interoperability.

2.2.4.8 City Object Groups

All the concepts of database science, including grouping, querying, aggregating, etc. are applicable in CityGML paradigm. Applying database consistency concepts in design CityGML paradigm allows user to regroup the objects based on their desire index or arbitrary attribute.

The grouping concept of CityGML allows for the aggregation of arbitrary city objects according to user-defined criteria, and to represent and transfer these aggregations as part of a city model (Booch, 2005).

2.2.4.9 Appearance

Information about a surface's appearance, i.e. observable properties of the surface, is considered an integral part of virtual 3D city models in addition to semantics and geometry (Booch, 2005). Appearance theme can be related to any arbitrary attribute, designed based on the cartographic principles. As a part of CityGML paradigm (see Figure 2-9) appearance module can also include georeferenced textures. This indicates which any cartographic design can be included in the CityGML 3D city models as a coherent and established module, without the risk of disrupting the 3D models consistency for further analysis.

2.3 3D City Models Architecture for Web Applications

The visualization and especially interaction with complex 3D city models represented in CityGML on the web is still a challenging area. The most important problem is that the CityGML documents containing the entire semantic of 3D city model are very large in size. This results a significant performance issues in their management and visualization in especially web based applications. Due to a large size of the entire city model, the performance and rich interaction in combination is a significant challenge (Chaturvedi et al., 2015).

Chaturvedi (2015) introduces the multi-level system architecture in order to solve the mentioned challenges for making 3D city models an applicable, flexible and practical online tool for different purposes and analysis. Chaturvedi's (2015) designed model is based on three independent levels (Figure 2-14):

- Information backbone: lowermost level, where 3D city models from different possible sources with different format are organized and stored
- Application level: as a bridge between end user and 3D data. Based on the intended application, the relevant data (structural, thematic or spatial; or mixture of them) will be recalled from backbone inventory. The acquired data will be presented to the user on the *Graphical User Interface* (GUI)
- End user.

Despite providing a coherent, consistent and interoperable platform for storing, organizing and exchanging thematic, semantic and geometrical 3D city data, CityGML standard is not suitable for visualization purposes due to its complex structure and schema. CityGML as discussed in previous chapter, has designed to store and exchange 3D data; not for visualization purposes. This reminds the necessity of establishing application to export and convert CityGML to viewable frameworks and formats.

Emerging the concepts of WebGL, HTML5 and open source WebGL-based virtual globe libraries (e.g. *CesiumJS*) have revolutionized accessibility and availability of the 3D data and drastically broaden the audience and applications of 3D city models. In the last part of this chapter these emerging concept will be profoundly discussed and investigated through authoritative references.

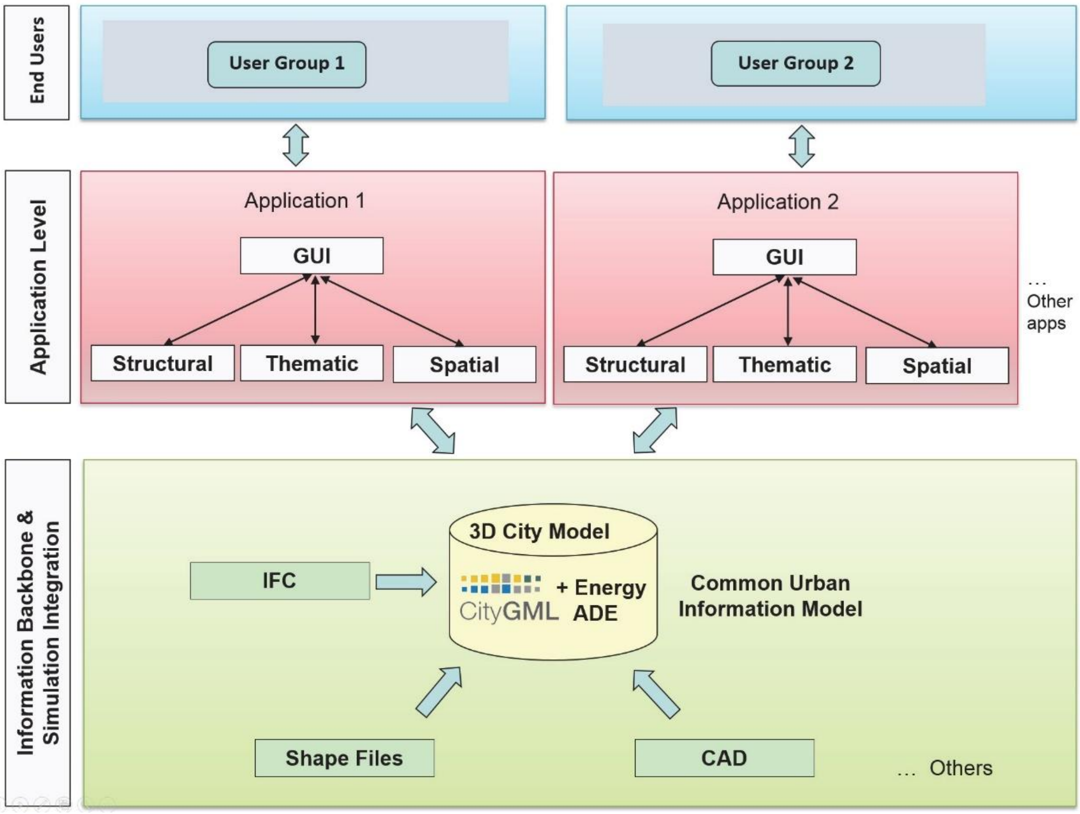


Figure 2-13: Multi-level architecture for the 3D city model web application (Chaturvedi et al., 2015).

2.3.1 HTML5 Canvas

HTML5 is the current iteration of HTML, the *HyperText Markup Language* (Fulton & Fulton, 2013). HTML5 is an open standard format and provides common platform for applications to be developed and used on the web; HTML5 enables the new generation browsers to support multi-threading, which allows to perform parallel execution of different tasks within one web page. (Chaturvedi et al., 2015).

HTML5 *Canvas* is an immediate mode bitmapped area of the screen that can be manipulated with *JavaScript*. Immediate mode refers to the way the canvas renders pixels on the screen. HTML5 Canvas completely redraws the bitmapped screen on every frame by using Canvas API calls from JavaScript (Fulton & Fulton, 2013). To the other words, HTML5 Canvas makes it possible to embed 3D graphical representation (e.g. 3D city models) in the web pages; JavaScript is the common programming language to create Canvas applications.

The basic HTML5 Canvas API includes a 2D context that allows a programmer to draw various shapes, render text, and display images directly onto a defined area of the browser window. In itself, the HTML5 Canvas 2D context is a display API used to render graphics on a bitmapped area, but there is very little in that context to create applications using the technology. By adding cross-browser-compatible *JavaScript* functionality for keyboard and mouse inputs, timer intervals, events, objects, classes, sound, math functions, and so on, you can learn to take HTML5 Canvas and create stunning animations, applications, and games (Fulton & Fulton, 2013).

2.3.2 OpenGL & WebGL

In order to understanding the concept of WebGL, first capsulizing OpenGL notion is necessary. OpenGL is a software interface to graphic hardware which consist of more than 700 distinct commands that you use to specify the objects and operations needed to produce interactive three-dimensional applications (Shreiner, 2009).

OpenGL is designed as a streamlined, *hardware-independent* interface to be implemented on many different hardware platforms. To achieve these qualities, no commands for performing windowing tasks or obtaining user input are included in OpenGL; instead, you must work through whatever windowing system controls the particular hardware you're using (Shreiner, 2009). In the realm computer graphics, the terms *Rendering* and *Model* have a key role for defining and explaining more advance concepts;

- Model: Or objects, are constructed from geometric primitives—points, lines, and poly-gons—that are specified by their vertices;
- Rendering: The *process* by which a computer creates images from models;
- Pixel: The smallest visible element the display hardware can put on the screen;
- Bitplane: Is an area of memory that holds one bit of information for every pixel on the screen, the bit might indicate how red a particular pixel is supposed to be, for example;
- Framebuffer: The bitplanes are themselves organized into a framebuffer, which holds all the information that the graphics display needs to control the color and intensity of all the pixels on the screen.

Infinite OpenGL applications and programs can be design by utilizing OpenGL graphics sys-tem. However, the basic structure of a useful program can be simple: its tasks are to initialize certain states that control, how OpenGL renders and to specify objects (model) to be rendered (Shreiner, 2009). Most implementations of OpenGL have a similar order of operations, a series of processing stages called the OpenGL rendering pipeline Figure 2-15 provides a reliable guide for predicting what OpenGL rendering pipeline will do.

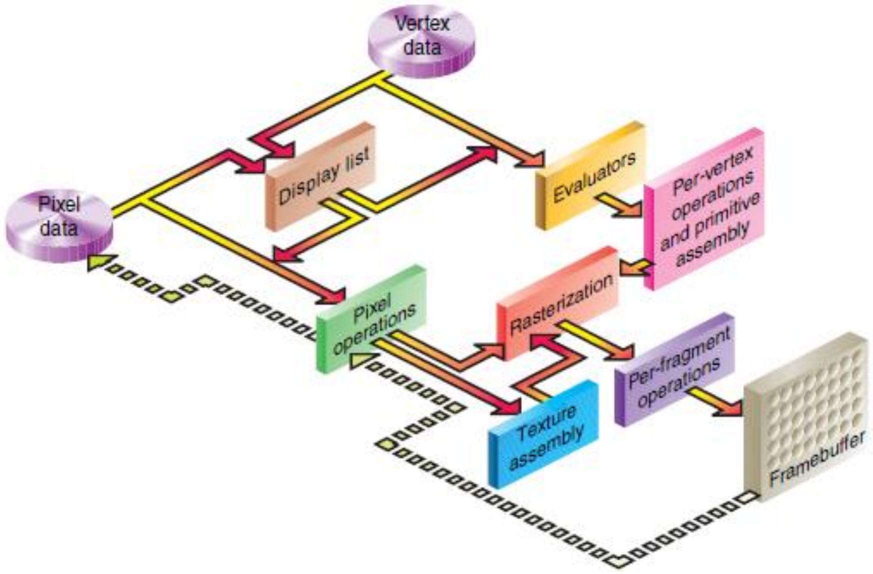


Figure 2-14: OpenGL rendering pipeline (Shreiner, 2009).

This diagram shows the Henry Ford assembly line approach; which OpenGL takes to processing data. Geometric data (vertices, lines, and polygons) follow the path through the row of boxes that includes evaluators and per-vertex operations, while pixel data (pixels, images, and bitmaps) are treated differently for part of the process. Both types of data undergo the same final steps (rasterization and per-fragment operations) before the final pixel data is written into the framebuffer (Shreiner, 2009).

For a better understanding the concept of visualization regarding OpenGL rendering pipeline, some of the operators has been further discussed by Shreiner (2009):

- Display Lists: An inventory which all data, whether it describes geometry or pixels, can be saved in a display list for current or later use.
- Evaluators: All geometric primitives are eventually described by vertices. Parametric curves and surfaces may be initially described by control points and polynomial functions called basis functions. Evaluators provide a method for deriving the vertices used to represent the surface from the control points. The method is a polynomial mapping, which can produce surface normal, texture coordinates, colors, and spatial coordinate values from the control points.
- Per-Vertex Operations: Converts vertices to primitives. Some types of vertex data (for example, spatial coordinates) are transformed by 4 * 4 floating-point matrices. Spatial coordinates are projected from a position in the 3D world to a position on your screen. If texturing is used, texture coordinates may be generated and transformed here. If lighting is enabled, the lighting calculations are performed using the transformed vertex, surface normal, light source position, material properties, and other lighting information to produce a color value.
- Primitive Assembly: The results of this stage are complete geometric primitives, which are the transformed and clipped vertices with related color, depth, and sometimes texture-coordinate values and guidelines for the rasterization step.
- Pixel Operations: While geometric data takes one path through the OpenGL rendering pipeline, pixel data takes a different route. Pixels from an array in system memory are first unpacked from one of a variety of formats into the proper number of components. Next the data is scaled, biased, and processed by a pixel map. The results are clamped and then either written into texture memory or sent to the rasterization step.
- Texture Assembly: OpenGL applications can apply texture images to geometric objects to make the objects look more realistic. If several texture images are used, it's wise to put them into texture objects so that you can easily switch among them.
- Rasterization: Rasterization is the conversion of both geometric and pixel data into fragments. Each fragment square corresponds to a pixel in the framebuffer. Color and depth values are generated for each fragment square.
- Fragment Operations: Before values are actually stored in the framebuffer, a series of operations are performed that may alter or even throw out fragments. The first operation that a fragment might encounter is texturing, where a texel (texture element) is generated from texture memory for each fragment and applied to the fragment. Next, primary and secondary colors are combined, after the final color and depth generation of the previous operations, the scissor test, the alpha test, the stencil test, and the depth-buffer test are evaluated. Then, blending, dithering, logical operation, and masking by a bitmask may be performed. Finally, the thoroughly processed fragment is

drawn into the appropriate buffer, where it has finally become a pixel and achieved its final resting place.

WebGL is designed as a rendering context for the HTML Canvas element (see 2.3.1). WebGL brings OpenGL (see 2.3.2.1) to JavaScript, and since JavaScript is primarily a browser-based programming language, we can say WebGL brings OpenGL to web. Unlike C++, there is no compilation step for JavaScript programs, so all that is necessary to execute JavaScript on a web page is to add a script tag to an HTML page for each JavaScript file we want to include (Cozzi & Riccio, 2012).

WebGL is an extension of HTML5 canvas element, which is now widely used for developing web applications requiring 3D visualization. A WebGL context is created by requesting a WebGL context from an HTML canvas element. Applying such an approach, 3D capabilities have been realized directly in all major web browsers running on all major operating systems without needing additional plug-ins or extensions. Another benefit of WebGL is that it utilizes hardware's graphics card memory for displaying and performing operations on 3D contents and hence, it provides hardware accelerated 3D functionality on the web.

Basically WebGL brings all of the OpenGL processing and visualization capabilities to the web browsers and also introduces a new way to deliver application: the web. Cozzi (2012) has listed advantages of the web over the traditional desktop applications:

- ***Zero-Footprint***: Plugins aside, browsing to a web page does not require an install, nor does it require the user to have administrator privileges. Users simply browse to a URL and expect their content.
- ***Cross-Platform***: The web provides a convenient way to reach all the major desktop operating systems: Windows, Linux, and OS X. Nowadays desktop browsers supporting WebGL include Chrome, Firefox, Safari, and the Opera 12 alpha.
- ***Cross-Device***: Another advantage of WebGL is that web browsers supporting WebGL are starting to become available on tablets and phones.
- ***It's Easy***: For OpenGL developers, WebGL is easy. For OpenGL developers, the challenge of moving to WebGL is not in learning WebGL itself. It is in moving to the web in general and developing in JavaScript.
- ***Strong Tool Support***: Both Chrome and Firefox with Firebug have excellent JavaScript debuggers with the features we expect: breakpoints, variable watches, call stacks, etc. They also provide built-in tools for profiling. Currently, both browsers have six-week release cycles for stable, beta, and developer releases.
- ***Performance***: Comparing to C++ as a native developing environment of OpenGL, JavaScript is slow. Given the nature of the JavaScript language, with its loose type system, functional features, and garbage collection, it's not expected to run as fast as C++ code.

WebGL enables web mapping to move from flat 2D maps to immersive 3D globes (see 2.3.3) (Cozzi & Bagnell, 2013).

2.3.3 Cesium

Cesium is an Open Source JavaScript package supporting the presentation of 3D contents within the web browser where users can dynamically switch between 3D globe visualization and 2D map projection. It utilizes WebGL to provide hardware acceleration and plugin independence and provides cross-platform and cross-browser functionality (Chaturvedi et al., 2015). To the other words, cesium is a WebGL virtual 3D globe, for visualizing dynamic geospatial data. Cesium 3D virtual globe is organized into the three assemblies[1]: Core.dll, Renderer.dll, and Scene.dll (Figure 2-16). these assemblies are layered such that Renderer depends on Core, and Scene depends on Renderer and Core (Cozzi & Ring, 2011). The different layers of the architecture are responsible to add specific functionality and to raise the level of abstraction and in general, the layer is usually dependent on the layers underneath it (Chaturvedi et al., 2015). Chaturvedi (2015) has classified the Cesium's architecture as:

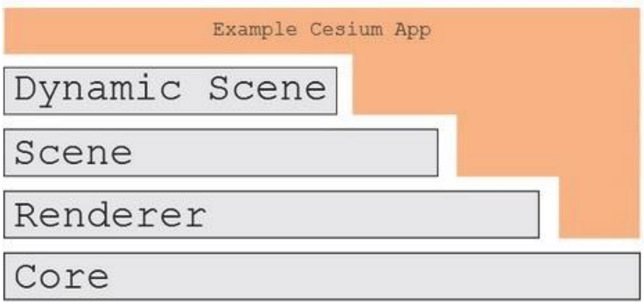


Figure 2-15: Cesium architecture (Chaturvedi et al., 2015).

- Core: This is the lowest layer in Cesium and includes mainly low-level functions. These functions majorly include computations and calculations such as mathematical conversions, transformations and projections. This assembly also contains geometric algorithms and engine infrastructures.
- Renderer: This layer is a thin abstraction over WebGL. It includes already available GLSL functions to provide shader programs, textures and buffers.
- Scene: This layer is mainly responsible to provide overall functionality of the globe. It includes high-level globe and map constructs such as 3D globe or map, handling layer imageries from multiple sources, creation of geometries and materials, camera control and animation.
- Dynamic Scene: This is the top-most layer of Cesium, which provides dynamic visualization of the data with the help of its in-built language CZML. It allows to store the data in dynamic objects, loads and renders the dynamic objects altogether instead of rendering every frame.

Chaturvedi (2015) indicates that the most distinguished features of the Cesium are:

- It is most suitable for dynamic geospatial data visualization with the help of Cesium Language (CZML). CZML is a JSON based schema, which describes geospatial data along with their properties that vary over time.
- It can integrate layer imageries from different sources, such as OpenStreetMap, Bing Maps, ArcGIS MapServer and standard image files. Even, the external WMS and TMS

[1] Assembly is the .NET term for a compiled code library (i.e., an .exe or .dll file).

can be integrated. Each layer, then, can be visualized according to specific brightness, contrast or saturation.

- It includes extensive libraries which support 2D as well as 3D geometries. The user can draw polyline, polygon, ellipsoid, sphere, labels, billboards and sensors.
- It supports data imports from KML, ESRI Shapefiles and JSON.
- It includes handlers to control mouse/keyboard events, camera movements and zoom and pan the virtual globe.
- It supports extensive materials to describe the surface appearance of the objects. It also supports custom materials for the objects.
- It supports math libraries to support major reference frames such as World Geodetic System (WGS84) and International Celestial Reference Frame (ICRF). The libraries also support conversions of coordinates and Cartesians.

However, cesium does not provide direct support of CityGML data, although as we discussed before (see 2.2.4), CityGML is a geo-database schema for 3D city models rather than a common 3D visualization format e.g. Google's KML COLLADA, etc. Along with, Cesium does not support dynamic loading of portions from large 3D city models yet (Chaturvedi et al., 2015). Examining its features, it is clear that Cesium has most of the functionality of Google Earth (either desktop or plugin version). There are some drawbacks, however: in most cases, Google provides higher resolution imagery, and higher bandwidth to reach them. But these issues are not present when we are using our own data (Gede, 2015) .

3 Methodology

After discussing the legitimate, most current and qualified literature and comprehending the current borders of different aspects of our topic, this chapter explains the chosen methods and their compatibility regarding the questions and existing scientific methods. The main objective of the thesis research is to take advantage of the capabilities of the online WebGL globes (e.g. Cesium) for representing 3D maps to a broader range of audience for different applications; designing new mixtures of cartographic design variables in 3D environment and evaluating them regarding specific applications.

The main challenges of the thesis can be divided to the two main chapters:

1. Challenges regarding the technical implementation:
    - software shortcomings and bugs,
    - commercial softwares and their licensing issues,
    - 3D data formatting and conversion,
    - visualization on 3D online WebGL Globe.
2. Challenges regarding aesthetic aspects of 3D city models:
    - cartographic design variables and principle in 3D environment,
    - finding the optimum cartographic design variable for a specific application,
    - competence of conventional 2D design principles in 3D environment.

This chapter discusses the mentioned challenges and investigate solutions to overcome them regarding discussed theoretical basis. In the final part of this chapter a general overview of the theoretical workflow to obtain our research goals and evaluating our design products will be presented. Basic structure of our thesis research project can be classified under three general steps; each containing their particular technical and design challenges:

- Cartographic design
- Visualization
- Evaluation

3.1 Data Availability and Preferences / Data Requirements

3.1.1 Data Availability and Design Softwares Limitations

One of the main challenges in our research was finding the relevant, proper and free 3D data providing the best and most flexible environment for applying our cartographic design elements and principles. Discussing about the desired formats without considering the available softwares for visualizing and editing 3D data is irrelevant. In our research we utilized CityGML data of the Munich city in the LOD 2; provided by Landesamt für Digitalisierung, Breitband und Vermessung (LDBV). This dataset is not accessible by the public and only in exceptional cases e.g. researches, it can be granted after the concessionaire's permission.

Due to the novelty of 3D design concepts to the realm of cartography, despite one SketchUp commercial extension which has been designed to import and export CityGML data, there are no other softwares with capability of editing the appearance of the 3D models for responding cartographer's demands and expectations in a sophisticated level of design.

As mentioned in 2.2.4, "[…] CityGML, a multi-purpose and multi-scale representation, and as an interoperable syntactic and semantic schema for data storage and exchange in the 3D city models environment […]", CityGML provides a consistent and coherent schema for storage and exchange of 3D models and their semantic and geographical specifications, but inherently, CityGML is not an official and applicable format for visualization, as well as thematic manipulations and design purposes. Hence for utilizing and expanding CityGML data applications into the design context, steps for data conversion are unavoidable and indispensable. Accordingly, for taking advantage of CityGML data and applying our thematic designs to 3D model in SketchUp environment, we had to convert it to one of the SketchUp's processable formats.

3.1.2 3DCityDB Importer/Exporter

3DCityDB Importer/ Exporter[1] developed in collaboration of the Chair of Geoinformatics, Technische Universität München (TUMGI), *virtualcitySYSTEMS GmbH*, and *M.O.S.S. Computer Grafik System GmbH*, a free Open Source package consisting of a database schema and a set of software tools to import, manage, analyze, visualize, and export virtual 3D city models according to the CityGML standard, has been used in our research as a conversion bridge for preparing the suitable data for further cartographic designs in SketchUp.

The 3D City Database (3DCityDB) is a free Open Source package consisting of a database schema and a set of software tools to import, manage, analyze and export virtual 3D city models according to the CityGML standard. In order to understand 3D CityDB schema and its necessity for importing 3D models into the database, first the appropriateness of the CityGML's structure for organizing and classifying graphical, semantical and geometrical information should be investigated. As Gröger (2008) indicates CityGML is a rich standard both on the thematic and geometric-topological level of its data model. On its thematic level CityGML defines classes and relations for the most relevant topographic objects in cities and regional models comprising built structures, elevation, vegetation, water bodies and more. In addition to geometry and appearance content these thematic components allow to employ virtual 3D city models for sophisticated analysis tasks in different application domains like simulations, urban data mining, facility management, and thematic inquiries.

The Import/Export tool provides functionalities for the direct export of 3D visualization models to COLLADA format, which is a processable and importable format in SketchUp Pro environment. Although conversion the CityGML data to COLLADA format makes it thematically designable in an advanced 3D software environment. It also has some drawbacks; conversion costs us loosing consistent semantic data structure of CityGML database. To the other words, all the possible database queries and aggregations regarding CityGML semantic and also graphical data structure, will no longer available in COLLADA format environment.

One of the solutions for stated problem of losing such a database structure of a CityGML data, is to implement thematic and graphical designs directly to the CityGML data. At the moment

[1] http://www.3dcitydb.org/3dcitydb/3dcitydbhomepage/

the only software which has the capability of directly importing a CityGML data to its environment for further manipulations is SketchUp's commercial extension *CityEditor*. CityEditor allows us not only take over the geometry and materials of the source files, but also imports object attributes. Moreover, editing mechanisms for those attributes in the CityGML-specific attribute scheme as well as a feature for the export of edited models to CityGML are available.

Unfortunately, since CityGML extension is commercial, it was not possible for us to use it for our design procedure; so converting CityGML with 3D CityDB to COLLADA was our only option to make it readable for SketchUp.

3.1.3 Visualization on Virtual Globes

As we discussed in previous chapter, evolution of the OpenGL concept and the possibility to implement it on the web browsers through the concept of WebGL, has introduced the concept of virtual globes as a medium for representing 3D geospatial data to a broad range of audience. In addition to data representation, genesis of new infrastructures enables the novel users and analysts to interact and explore throughout the 3D models. Hence the web has not just broadened the circle of 3D data audiences, but it also offers a wide range of data interaction tools and filters. Virtual globes are so named because of their approach to visualizing the Earth as a three-dimensional globe that one can "fly" above (Schultz et al., 2008). Following we will briefly introduce some of the well-known virtual globes and compare their capabilities with CesiumJS.

Google Earth is the most popular virtual globe application currently available. It is a free but closed-source application; thus, unlike the CesiumJS, users cannot modify the software as per their specific needs. Google Earth is not a GIS, but because of its ease of use, its ability to easily incorporate data sets created by users, and Google's large presence in society, it has reached more people (Schultz et al., 2008). Online 3D WebGL globes have their own specifications in terms of data formats and visualizations, e.g. Google Earth *.KML/*.KMZ. KML which is a simple, open source language that is supported by many virtual globes. KML's unique strength is that geometry, styling, display behavior, and feature attribute information are all contained in a single file that can be compressed using standard zipping technologies (*.KMZ). The user can download KML that others have created, load them into Google Earth, and explore. Some of the major Google Earth's capabilities and specialties can be listed as:

- User-friendly interface,
- A lot of functionality including; navigation, search, drawing and measuring, place-marks and tours, 3D terrain and buildings, support of GPS data containing place-marks, vector data, raster overlays and 3D models, views of ocean, sky and planets, ability to adjust sunlight, print, share, overview map, fly through, historical aerial photography, street view, SketchUp, lots of vector data available.,
- Extra features include dynamic imagery source and date, dynamic coordinates and elevation at cursor, dynamic eye altitude,
- Ability to link to all other Google applications and accounts,
- Google Earth Pro has advanced measuring and drawing tools and the ability to create movies,
- Imagery loads seamlessly

Some of the minor deficiencies of this platform are:

- Cannot turn off imagery i.e. no other basemaps,
- Custom tile sets and custom terrain data are difficult to add.

NASA (National Aeronautics & Space Administration) World Wind is a free, open-source JavaScript software development kit (SDK) based cross-platform (i.e., based on OpenGL and Java), application that is part of NASA's Learning Technologies program. It was developed specifically as an educational and research tool to explore earth and it provides access to a wide range of NASA satellite imagery. Data can be imported through servers, Open Geospatial Consortium (OGC) Web Services, and there is limited support for KML. The open source nature of World Wind client browser in conjunction with a modular expandable systems architecture allows developers to easily customize the client through plug-ins (Bell et al., 2007). World Wind's focus is toward scientific users, so it has a more specialized community than that of Google Earth. World Wind's functionality can be enhanced by using one of many add-ons (small extensions) and plug-ins that are created by users (Rakshit & Ogneva-Himmelberger, 2008). Web World Wind provides an API that enables JavaScript programs to control every detail of visualization and interaction. Web World Wind provides a rich set of features for displaying and interacting with geographic information. Because Web World Wind is completely open-source and designed to be extensible, extending the API and functionality is simple and easy to do:

- Display high-resolution imagery, terrain and geographic data from any public or private source,
- JavaScript API for automating all aspects of interaction and visualization,
- Large collection of built-in high-resolution imagery and terrain,
- Simple to extend and modify, designed to be extensible, highly configurable and customizable,
- GIS formats support: Shapefile.

WebGL Earth 2 is an open source, Leaflet compatible API (Leaflet is a modern open-source JavaScript library for mobile-friendly interactive maps). It allows easy embedding of a 3D globe in websites, with out-of-the box user friendly features and support for mobile devices. WebGL Earth uses the CesiumJS project for the rendering of core globe data and the WebGL developers contribute code back into the Cesium project. It is free and there is no installation required, access is via a web browser. It can be easily used with little customization or users can choose to customize it to their own needs. The out-of-the box features and a focus on rendering custom map tiles with the Klokan MapTiler targets a slightly different audience to Cesium. It is a separate platform to Cesium despite using and contributing to Cesium, and it is also a visualization application, providing some base data.

As we discussed in 2.3.3, CesiumJS is an Open Source virtual globe JavaScript package supporting the presentation of 3D contents. It uses WebGL (web graphics library) for hardware-accelerated graphics, and is cross-platform, cross-browser, and aimed at dynamic-data visualizations. Cesium is under the Apache 2.0 license so is free for commercial and non-commercial use. It is different from a 3D globe like Google Earth because it is not a complete application targeted at end users. It requires programming to use and has a lot of potential for customization and user added content. Being a free-source means unlike Google Earth users can modify the software applications and interface appearance based on their preferences. Simplicity and ease of use for the novel users, flexibility and its growing application and popularity in private and public domains in different fields of applications are the main reasons for preferring CesiumJS as a virtual globe platform in our thesis over Google Earth and NASA World Wind. Nowadays a broad range of virtual globes have been built on the powerful CesiumJS library and extending its functionalities and capabilities and enhancing its performance based on their specific applications e.g. Skyline TerraExplorer[2] and Bhuvan-3D[3] (India) use Cesium in their National Map software, WebGL Earth 2 (see the previous paragraph).

CesiumJS detailed capabilities can be listed as:

- Visualize high-resolution global terrain and DTMs.
- Layer imagery from multiple sources, including WMS, TMS, WMTS, Bing Maps, Mapbox, Google Earth Enterprise, OpenStreetMap, ArcGIS MapServer, standard image files, and custom tiling schemes. Each layer can be alpha-blended with the layers below it, and its brightness, contrast, gamma, hue, and saturation can be dynamically changed.
- Industry standard vector formats, such as KML, GeoJSON, and TopoJSON, including terrain clamping.
- Draw 3D models using glTF with animations and skins. Clamp models to terrain. Convert COLLADA and OBJ to glTF using the online converter.
- Create data-driven time-dynamic scenes using CZML.
- Draw and style a wide range of geometries:
  - polylines
  - billboards
  - labels
  - points
  - draw, clamp to terrain, and extrude polygons, polygons with holes, rectangles, circles, and ellipses
  - boxes, spheres, ellipsoids, and cylinders
  - corridors, polyline volumes, and walls
- Shadows, including self-shadows and soft-shadows for terrain, 3D models, and geometries, based on the sun position.
- Draw the atmosphere, fog, sun, sun lighting, moon, stars, and water.
- Individual object picking.
- Camera navigation with mouse and touch handlers for rotate, zoom, pan with inertia, flights, free look, and terrain collision detection.
- Batching, culling, and JavaScript and GPU optimizations for performance.
- Precision handling for large view distances (avoiding z-fighting) and large world coordinates (avoiding jitter).
- A 3D globe, 2D map, and Columbus view (2.5D) with the same API.
- Cluster points, labels and billboards.

3.1.4 Binary glTF

In the thesis project, CesiumJS which is an online JavaScript library for visualizing 2D or 3D data models, chose as the virtual globe and online visualization platform. It takes advantage of the WebGL concept for online visualization of the 2D or 3D content. One of the cesium's specifications is its processable 3D format; Graphics Library Transmission Format (glTF),

---

[2] http://www.skylineglobe.com/SkylineGlobe/corporate/Default.aspx?
[3] http://bhuvan.nrsc.gov.in/globe/3d.php

which was initially designed and specified by the *Khronos Group* in October 2015, for the efficient transfer of 3D content over networks, to other words, glTF bridges the gap between various 3D models formats and GL based APIs (e.g. WebGL). glTF is a royalty-free specification for the efficient transmission and loading of 3D scenes and models by applications. glTF minimizes both the size of 3D assets, and the runtime processing needed to unpack and use those assets. glTF defines an extensible, common publishing format for 3D content tools and services that streamlines authoring workflows and enables interoperable use of content across the industry.

Streaming massive data sets to internet browsers and mobile applications involves transferring hundreds and thousands of data packets over HTTP. Due to technical aspects of the HTTP protocol, the number of concurrent connections per hostname is limited. Hence, bundling data to fewer files reduces the streaming workload. A common technique to include binary data such as JPEG images in XML or JSON files is Base64 encoding. Binary glTF addresses these issues. It is an extension to glTF and allows merging all glTF assets (JSON file, binary file, images and shaders) into a single file. The layout of a binary glTF file (see Figure 3-1) includes meta data, the scene description as JSON, and the binary data block containing all vertex, index, image, and shader data (Schilling et al., 2016).

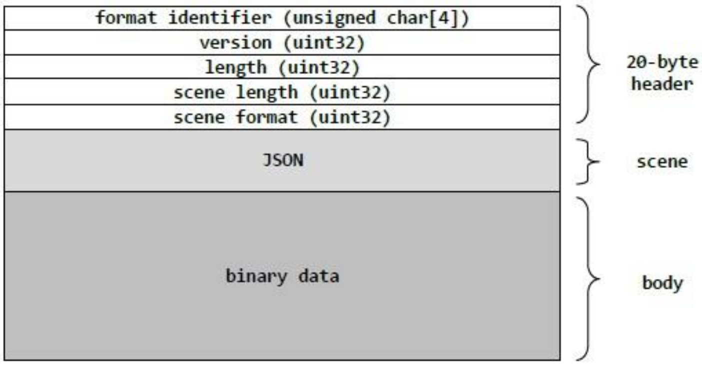


Figure 3-1: Layout of binary glTF file (Schilling et al., 2016).

The core of glTF is a JSON file that describes the structure and composition of a scene containing 3D models. The top-level elements of this file can be seen in the Figure 3-2.

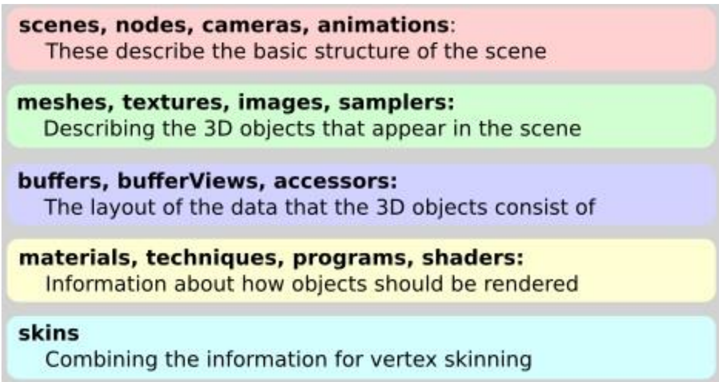


Figure 3-2: Top level elemnts of GL Transmission Format (Schilling et al., 2016).

These elements are given as dictionaries. References between the objects are stablished by using IDs to look up the objects in the dictionaries.

Hence for visualizing our designed 3D city models on the Cesium.js platform, we should initially perform another conversion to the glTF format. This task can be performed though Cesium's online converter, or for the large models (larger than 10MB), command line batches e.g. collada2gltf can be used. For embedding CityGML content in CesiumJS, all data must be made available as 3D Tiles layer and converted into glTF. Due to different concepts regarding spatial data representation and basic structuring, a series of processing steps must be performed that go way beyond a simple format conversion. Table 3-1 summarizes and compares the key features of CityGML, X3D, glTF and B3DM (Batched Three Dimensional Model) that are relevant in this context (Schilling et al., 2016).

	CityGML	**X3D**	**glTF**	**B3DM**
Coordinate Precision	double	single	single	As in glTF
Vertex data encoding	In UTF-8	In UTF-8	As little endian binary data (through binary glTF extension)	As in glTF
Inner Spatial Reference Frame	Using spatial reference systems from surveying, e.g. UTM	Local Cartesian coordinate system	Local Cartesian coordinate system	As in glTF
Georeferencing of Models	Not necessary	Through Geospatial component	Through CESIUM_RTC extension	As in glTF
Geometry types	3D polygon, Solid, Multi-Surface	Primitive types, Extrusion, ElevationGrid, IndexedFaceSet	Triangle mesh	As in glTF
Object/Feature Definition	CityObject XML node	Shape XML node	JSON node element	Batch
Image encoding	JPEG, PNG, TIFF	JPEG, PNG	JPEG, PNG	As in glTF
Attributes	Generic and feature-specific attributes	none	none	As batch table
Semantics	Full support	none	none	Can be partly mapped to attributes
Packaging	CityGML file plus texture image files	X3D file plus texture image files	Single file (through binary glTF extension)	Single file

Table 3-1: Comparison of key features between CityGML, X3D, glTF and B3DM (Schilling et al., 2016).

As you can see in the table 3-1, all attributes and semantic data will be neglected during the conversion process from CityGML to glTF and won't be modelled in glTF data structure, this problem can be partially solved by using B3DM format in parallel to the glTF format in visualization applications.

3.2 Cartographic Design of 3D City Models

In addition to the profound inventory of the cartographic design principles, and developed visual and graphical variables for the conventional 2D maps e.g. Bertin (1983), there are also number of researches dedicated to defining graphical variables in 3D design environment e.g. Häberling (2005). But cartographic visual variables in 3D environment has a lot of rooms for further improvements in future, because:

- The novelty of thematic design in 3D environment,
- The possibility of utilization of conventional principles in 3D designs,
- Absence of a professional 3D design software for city modeling,
- Lack of knowledge regarding the optimal visual variable for a specific application.

The last option defines the main objective of our thesis research. Some studies have proven (Zanola et al., 2009) that detailed photorealistic 3D model does not always provide the best results for all applications; to the other words, it should be always considered that the application of a 3D model determines the visual and graphical principles to be considered in the design procedure. In fact, cartographers have long been aware of the necessity of presenting an abstract view of the infinitely complex world in the form of 2D maps, and several authors from the geovisualization and cartography community have advocated the use of more abstract 3D views for more efficient inference making (Döllner, 2007). For example, in an empirical study Plesa and Cartwright (2008) found that users prefer non-photorealistic 3D maps for mobile navigation devices, because of their enhanced clarity and better usability. A detailed and fully photorealistic textured model does not necessarily lead to the best result for a specific application. This is the same concept which has been considered for designing conventional 2D maps; hence, for a successful design we should first focus on the map's application context.

Zanola (2009) describes the level of appropriateness of a specific model regarding user's knowledge toward the information visualized in the model. Zanola (2009) suggests that realistic depictions are the easiest form of visualization for novice users to communicate intended information embed in the 3D model illustrations, whereas more abstract depictions believed to be better suited for an expert audience with the appropriate domain of knowledge toward visualized information. Furthermore, photorealistic models might be seductive in term terms of accuracy. Meaning, photorealistic models can easily convince the audiences to be more precise and accurate rather than abstract ones, while there is no logical relationship among these two factors.

Despite the limited and sporadic attempts for defining novel design principles in 3D environment, e.g. progressive and degressive perspectives (Jobst & Döllner, 2008), communicating the reliability with the fusion of levels of realism: photorealism and wire frame models for archeological monuments (Zuk et al., 2005), there are no researches for linking these design novelties to the applications. Different applications of the 3D city models have been clarified in section 2.2.3; in this thesis research different design principles for two distinct applications will be examined; Visualization for communication of urban information to citizenry, and visualization for navigation purposes.

After converting the CityGML data to the COLLADA format, cartographic design variables can be assigned to the model in the SketchUp environment. SketchUp provides a wide range of graphical tools for manipulating the appearance of the model; color, size, texture, transparency, and etc. can be changed and customized. It is also possible to assign photorealistic textures to the surfaces; geo-referencing the model is also possible in SketchUp environment by adding the location which simply pins the model to our desired position on earth. After exporting the designed modes as COLLADA format, it will be converted to binary glTF structure for visualizing in the Cesium.js (see 3.1.3).

3.3 Evaluation

The final phase of the thesis is dedicated to evaluation of the 3D city designs. A questionnaire based evaluation for estimating appropriateness of the designed 3D models regarding different questioned applications will be conducted. In our evaluation we will identify the optimal visual variable for a specific application, or equivalently, we will connect best fitting graphical variable to the applications. After conducting the evaluation, the results will be statistically analyzed and discussed to draw conclusions for the thesis project.

The questionnaire based survey for evaluation of the 3D models was preferred over other options for evaluating 3D models e.g. evaluating the models by visualizing them in computers for participants, because of our limited infrastructures and time, as well as probable navigational issues for novel users in 3D environment which could lead to distraction and frustration in completing questioned tasks. Hence the aim of the evaluation mapped to the questions, and the results statistically analyzed and discussed to draw conclusions for the thesis project, and finally establishing new foundations for deriving new hypothesis for future researches and investigations. Detailed information regarding evaluation, the results of the questionnaire and analysis of the outcomes can be find in the chapter 5; in addition, the questionnaire is reachable in appendix C.

4 Practical Implementation

Chapter four is dedicated to document the detailed procedure which has been performed in order to reach the final results for further analysis in evaluation phase. Based on the available and accepted hypothesis and theories discussed in chapter two, and available technological infrastructures, hardware and software, a pipeline designed to produce meaningful 3D models; meaningful means models which meet both theoretical principles in cartographic design and aesthetic, and also consider the application context of the 3D models. As we discussed in previous chapters, designing maps and models regardless of their dimensions (2D, 2.5 perspective or 3D models), is a function of design and visual variables and their intended application domain; hence it is critical to identify models application parallel with design procedure.

Following this chapter, we will discuss our general designed pipeline for our research thesis. Furthermore, technical steps regarding data preparations and conversions will be presented in details. 3D CityDB, SketchUp Pro, CesiumJS are the main applications and technical backbone of the project. Despite these three major applications, we will utilize a number of complementary command line batch applications and database management softwares including PostgreSQL. In the final phase of technical implementation of our project, we will customize the CesiumJS interface for embedding our models and facilitate recalling them in the browsers environment.

4.1 General Workflow; 3D Data Conversion and Designs

CityGML model of Munich city center is considered as the initial dataset of the thesis. As we discussed in chapter 2.2.4, the semantics of CityGML, this 3D data format is not designed as a visualization carrier, hence the characteristics and structure of a visualization format has not foreseen in it. So the first task of our project was to convert CityGML data to a common and understandable format for graphical and visualization application softwares. Initial conversion of CityGML data will be performed with 3D CityDB software. After translating the data to a graphical format, it will be imported to the Google's SketchUp environment for assigning different sets of visual variables to the 3D city models considering their intended applications in different platforms e.g. desktop or mobile environment.

Visualizing the designed models in a browser and web environment is the next step. As discussed before CesiumJS as the most flexible and free-source virtual 3D globe has been considered as the core of the project. The initial focus of our project was to employ CesiumJS as the platform for representing cartographic products (2D or 3D in our project case) in the net environment. In order to be able to visualize our 3D models in CesiumJS, they should be converted to one of its processable formats: *.glb or *. glTF. Hence, converting the models to binary glTF (see 3.1.4); this task can be performed using command line converters or online converters. All the general steps shown in the figure 4-1.

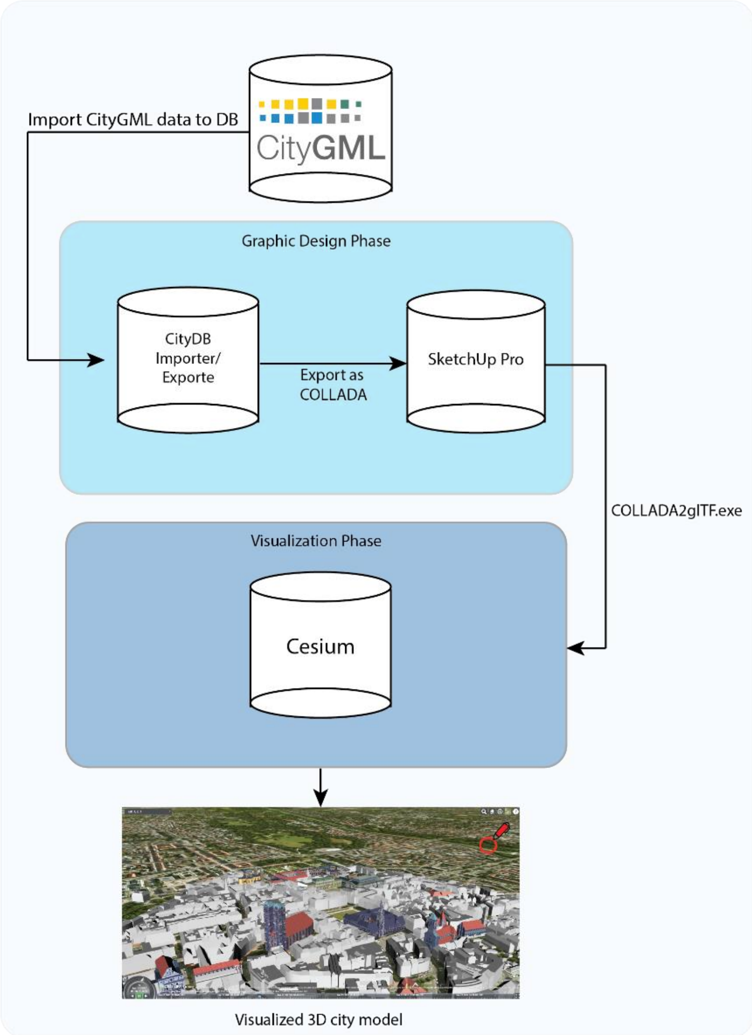


Figure 4-1: Thesis general workflow.

In the following subsections, all the steps will be profoundly discussed and the results of each phase will be reviewed in details.

4.2 SketchUp

Due to the novelty of the 3D city models, borders for the application domain and their relationship with 3D cartographic design concepts are blurry and fuzzy. Hence the thesis project faced two main challenges in the cartographic design phase; first identifying relevant applications for 3D city models, and second, applying the best fitting existing cartographic visual variables to the models.

3D city models different application and their logical classification has been outlined in section 2.2.3. Based on our analysis and detailed studies, we chose two distinct applications which cartographic design play a vital role to their appropriateness and effectiveness:

- Visualization for communication of urban information to citizenry,
- Visualization for navigation purposes.

After defining the applications, appropriate cartographic visual variables regarding the chosen applications selected. Application of different LODs; a mixture of photorealistic textures and solid surfaces, edge enhancement and color coding of the model's semantics levels, and size and transparency compound, drafted as the final design concepts for 3D cartographic visual enhancement concepts. Following we will review all the designed models and will specify their exclusive characteristics in terms of visualizing different aspects of the 3D city models regarding the applications.

4.2.1.1 Different LODs

Photorealistic displays are frequently lauded for their near-effortless comprehensibility and ease of use for non-experts (Zanola et al., 2009). Zanola (2009) suggests that photorealistic depictions are the easiest form of visualization for the public to associate with and understand and more abstract displays believed to be better suited for an expert audience. On the other hand, cartographers always have been aware of the necessity of presenting an abstract view of the infinite reality. The idea of initiating a 3D city model containing both photorealistic and abstract depictions originated from discussed ideas; for communicating urban information to citizenry applying the photorealistic textures to the questioned buildings provides the easiest and fastest perception procedure due to their effortless comprehensibility, and for the navigational purposes, photorealistic textures for the city's touristic and historical attractions for their ease of perception as navigational landmarks.

Based on the declared concepts, photorealistic and solid textures had to assigned to the buildings based on their applications and different semantics. Google's SketchUp Pro has been employed as the 3D city models designing software in our thesis. After importing the resulted COLLADA model acquired from the 3D CityDB Importer/Exporter (see 4.2) to the SketchUp environment, we are ready to assign different graphical variables to the 3D model. To be more specific, in our study case different LODs predicates to assigning photorealistic textures to the city's landmarks and solid textures to the other and less important urban structures (this selection is based on our application purposes of the 3D city models and they can change in different applications). Since our initial dataset did not accompany by any texture data (neither photorealistic nor graphical textures), we had to assign all the textures to the landmarks manually. SketchUp provide options to add photo textures to the different surfaces. Also the opacity and the size of the assigned photo textures are changeable.

More than hundred distinct texture images acquired from Google Earth to assign photorealistic textures to the touristic and historical landmarks of Munich's city center, whereas other buildings visualized as a bare and solid color geometries. The Figure 4-2 demonstrates the final product of the cartographic designed 3D city model which indicates to different LODs with variant texture types.

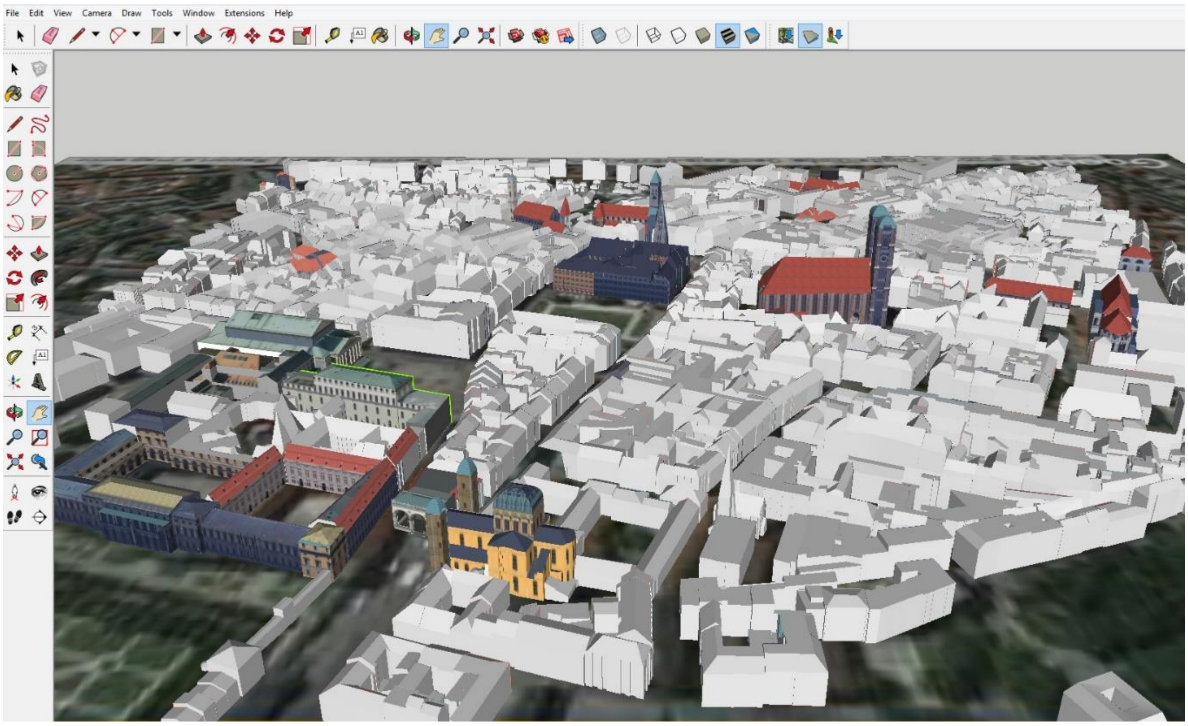


Figure 4-2: Different LODs; designed 3D model in Google's SketchUp environment.

Based on the concept of Level of Details described in section 2.2.4.2, detailed and photorealistic landmarks can be classified as models in LOD3, whereas other solid textured structures represent a model in LOD1. The most challenging part of designing such a models was to acquire texture images and assigning them to dominantly multiform surfaces of the buildings. Since during the conversion process form CityGML we lose all the sematic data and CityGML scheme structure of the dataset, designing a solution for automating the texturing procedure based on the database queries and modifications is not applicable. Hence all the texturing tasks had to be done manually with supervision of the operator.

4.2.1.2 Edge Enhancement and Color Coding

Edge enhancement has long been considered as one of the most efficient graphical variables in 3D design environment. Semmo (2012) indicates: "An edge enhancement of 3D objects highlights structural aspects of virtual 3D city models by emphasizing and separating features located in the background of an image". Despite highlighting the structures, edge modifications and sketchy stylization method has been used to communicate uncertainty or separating the actual existing building from buildings which has not gone beyond the initial planning stage. In an empirical study Schumann et al. (1996) could demonstrate that sketch-style, non-photorealistic, computer-based 3D views are significantly preferred over shaded displays or CAD-like renderings when architects present first drafts to clients, because they appear to be less finished (Zanola et al., 2009). For reliability purposes, Zuk (2005) claim that in archeological domain, sketch-style views were found to encourage more discussions about potential modifications to the plan than the other display types; they modified the level of realism as a visual cue for the degree of *temporal uncertainty* in 3D archeological reconstructions. They used photo-realistic 3D views and wire frame models to communicate the reliability of the dating method of a given archeological monument (Zanola et al., 2009).

Color as one the main visual variable in the realm of cartographic design has been investigated in several references. The colorization of geometric primitives adopts design principles known from cartography for visual abstraction and efficient communication of thematic geoinformation (Semmo, 2012). Brewer (1994) proposes conventions for using color in cartography; she demonstrates combination color schemes for which organization of the perceptual dimensions that can be matched with data organizations:

- qualitative/binary
  (for example, compare land uses inside and outside an aquifer recharge area)
- qualitative/sequential
  (compare general vegetation types to precipitation patterns)
- sequential/sequential
  (compare educational attainment to crime rates)
- diverging/sequential
  (compare cancer rates above and below a mean rate to air pollution levels)
- diverging/diverging
  (compare areas above and below the poverty line in 1960 and 1990)
- diverging/binary
  (compare income levels above and below a median to representation by republicans or democrats)

(Brewer, 1994).

In combination with the edge enhancement, color schemes can improve the perception of city structures (Semmo, 2012) (Figure 4-3). In the figure 4-7 touristic attractions of the city colored in blue, and the hotels are colored in orange. In addition, the edge of these colored building has been enhanced for a clear and better represent of their structure.

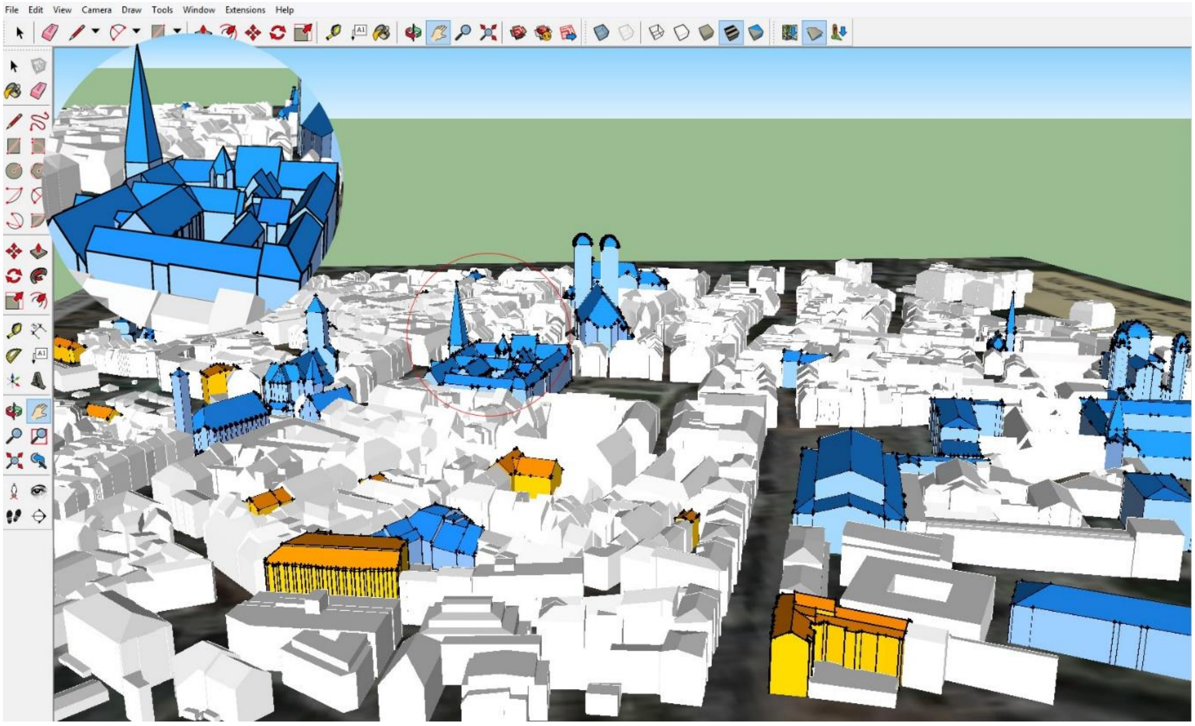


Figure 4-3: Edge enhancement and color coding ; designed 3D model in Google's SketchUp environment.

The only possible way to enhance edges in SketchUp environment is to assign pipes along edges; for this purpose, all the individual edges in the desired building models one by one selected and the pipes signed to them subsequently. In our models we took advantage of the Brewer's qualitative/binary color scheme for coding two major classification of building types; the blue color for touristic landmarks and orange color for visualizing hotels located in the city center area. In addition of considering different qualitative colors for different types of buildings, a series of sequential color tones were initiated for separating building's roofs from their lateral surface. This initiative boosts user's perception regarding buildings structure and highlights the three dimensionality of the model in distinct lighting and shadowing scenarios in different visualization environments. In addition, in the bird eye view of the 3D models, majority of the visualized phenomenon in the scene have been formed by the building's rooftop geometries, hence considering rooftops as an important visual index for recognizing the buildings during design workflow is necessary. This is a contradictory notion regarding some conventional 3D cartographic design concepts which considers the rooftops as the least important section of the building when discussing about cartographic visual design of the buildings. Different colors and color tones assigned to the surfaces through *Material* dialogue option in SketchUp environment.

4.2.1.3 Size and Transparency

As we discussed in section 3.2, for photorealistic 3D city models it is easier to convince the novice users that the underlying 3D data are more accurate and closer to reality rather than non-photorealistic depictions. On the other hand, visual variable *size* can be considered as an advantageous index in cartographic design pipeline when the actual geometrical precision of the individual buildings does not have a significant importance considering intended context of the 3D model's application (which is the case in this thesis project). Since neither communication of urban information to citizenry, nor Visualization for navigation purposes are susceptible to the dimensional and positional precision and perfection.

Accordingly, size and photorealistic texturing can be perfect complements for each other in terms of visual design; while photorealistic textures can conceptually trick the novel users about being precise and legitimate, the size index can be applied to enhance the visibility of the landmarks in the compact and fully occupied city environments for presentational and navigational purposes of the 3D city models. Transparency is a well-known variable in cartographic design, Semmo (2012) indicates: "*Transparency effects are a well-known method to improve comprehension and visibility of occluded model entities in 3D-space*". Also transparency has been employed to highlight construction states of buildings models, Further, transparency effects are used in web services and systems, such as Google Maps, to aid the perception of complex structures or the architecture of 3D building models (Semmo, 2012).

Expanding the actual physical size of the landmarks reinforces their visibility and recognition process in the 3D environment; accompanied with assigned transparency to the less-relevant urban structures, accelerates process of acquiring and conceptual understanding of the location and structure of them in occluded and dense city environments (see Figure 4-4).

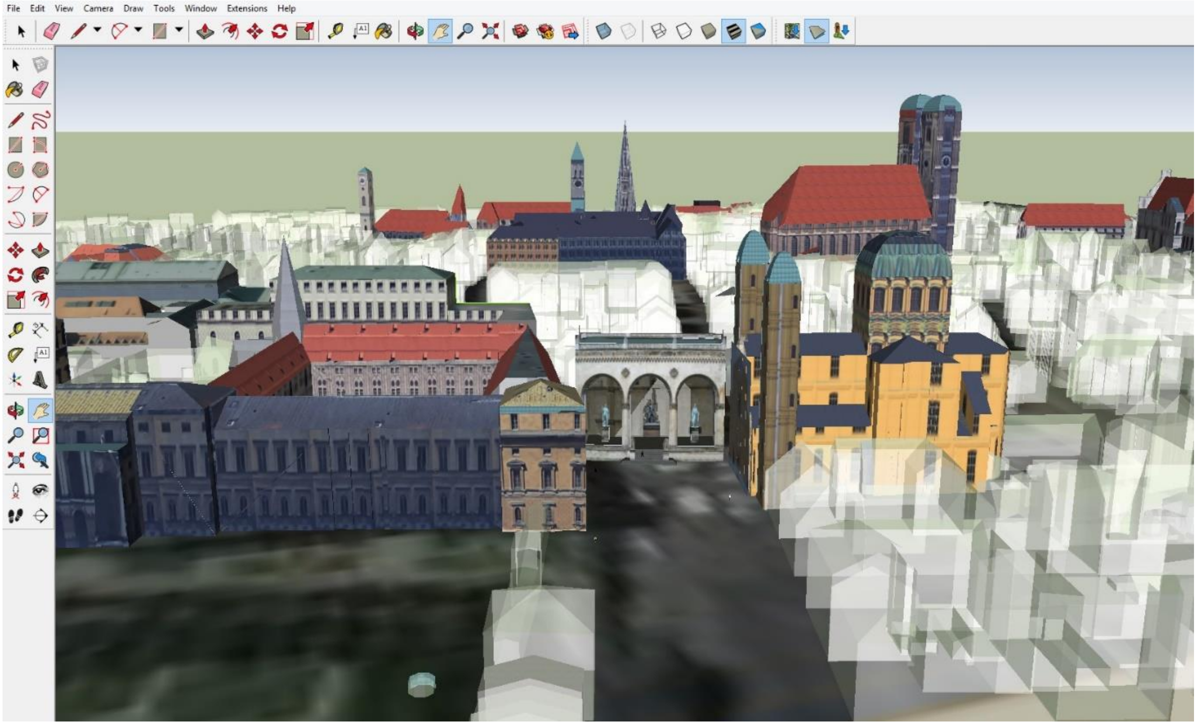


Figure 4-4: Size, photorealistic texture and transparency assigned to the 3D model in Google's SketchUp environment.

In our model distinct buildings have been enlarged with the factor of 1.7 to 2.2. Simultaneous with enlargement we were extremely cautious to preserve distinct structures size ratios regarding their surrounding buildings in an admissible and acceptable range. In addition to providing clarity in crowded city environments, transparency accompanied with photorealistic texturing unconsciously derives the attention toward the city landmarks; which is our optimal goal in presentation and navigational applications.

Prepared 3D models in SketchUp Pro will be exported to the COLLADA format for further proceeding towards visualization in CesiumJS virtual globe; which has been profoundly discussed in next sections.

4.3 Conversion to glTF

Processable format by CesiumJS virtual globe is *. glTF (graphics library Transmission Format); whereas exported 3D city models from SketchUp pro have been formatted in COLLADA. Hence for visualizing the 3D models in CesiumJS, they should first be converted to the glTF format. There are several online and windows command line applications offering free conversion and transformation from COLLADA to glTF schema. CesiumJS web portal provides an online converter (https://cesiumjs.org/convertmodel.html) which converts COLLADA (*.dae) or OBJ models to glTF. User can also indicate to either download binary glTF or quantized and oct-encode glTF format.

All the 3D COLLADA models exceed the volume limit of 10MB for online conversion. Hence *COLLADA2glTF*[1] which is a command-line conversion tool was chosen as the converter. Developed by *KhronosGroup*, *COLLADA2glTF* is a flexible converter which provides us variety of different options to achieve our desired glTF model. The binary glTF datasets resulted from conversion tool are ready for visualizing in CesiumJS which later on can be published and distributed to the public sectors and individuals on the world wide web.

4.4 Visualization in CesiumJS

The final task of our thesis's practical implementation is to visualize the exported glTF models in CesiumJS. CesiumJS provides a flexible interface which is easily customizable with few lines of JavaScript coding. After downloading the latest version of CesiumJS (current version 1.30), the first step in order to run the Cesium applications on our computer is to setup a local web server to host our directories; for the project *Node.js* has been used for providing a local web server.

In thesis project the simplest HTML template was customized for embedding the 3D models to the virtual 3D globe. The initial *Cesium Viewer* interface contains a number of attractive widgets and plugins providing functionalities like switching between viewing modes (2D, 2.5D Columbus view and 3D) and handling imagery and terrain layers. Following outlines some of the viewer's components and their functionalities (Figure 4-9):

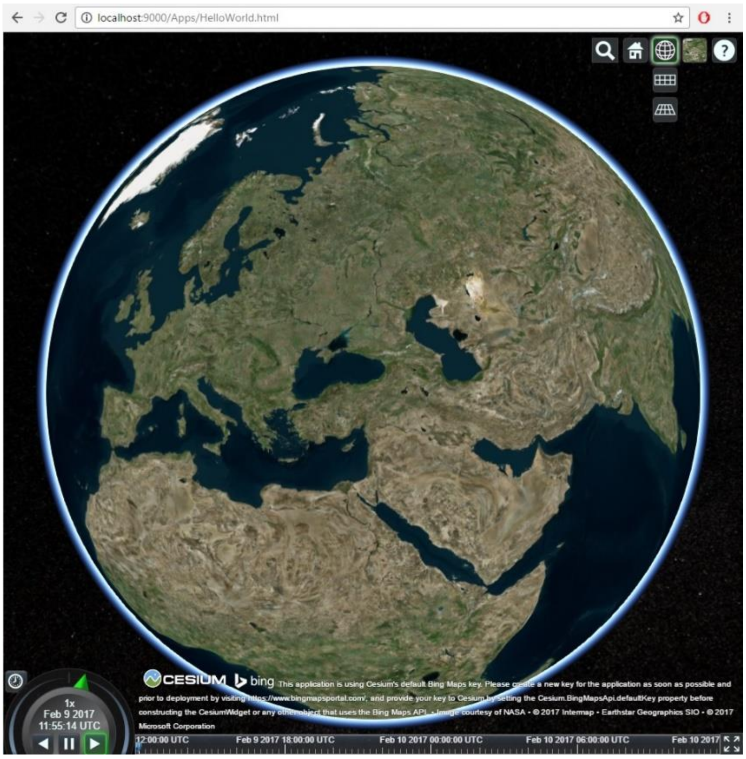


Figure 4-2: Cesium viewer graphical interface[2].

[1] https://github.com/KhronosGroup/COLLADA2GLTF/
[2] http://cesiumjs.org/

- 3D virtual globe: Cesium widget that allows user to navigate through the globe by panning, moving, tiling, and rotating the camera perspective.
- Navigational component: contains panning and zooming functions.
- Cesium toolkit widget: containing geocoder, home button, scene mode picker, base layer picker and help button.
- Time scale: demonstrates the actual date and time on a scale bar.
-

Cesium viewer does not include predefined functions for visualizing multiple 3D city models; hence we had to define a custom functions to automatically call the models from the defined directory and visualize them on cesium virtual globe. To do so we defined some simple functions in JavaScript and introduced our 3D model's directories to the function. Our script adds a simple widget to the cesium viewer's which enables users to choose their desired 3D model from a drop down menu. Appendix D contains the written JavaScript syntax in the thesis which forms the backbone of the CesiumJS virtual globe.

Executing our embedded JavaScript syntax on the running local server through HTML request, leads us to the final phase of our thesis's practical implementation; visualizing cartographically designed 3D models in CesiumJS 3D virtual globe environment (Figure 4-5).

Figure 4-5: Visualizing designed models in Cesium viewer. The user can switch the models through the drop-down menu in top-left corner.

We also enabled shadowing in our models for more realistic signature; after successfully visualizing the models in Cesium viewer we have evaluate the appropriateness of designed models regarding the intended applications, which is the subject of our next chapter; evaluation and discussion.

5 Evaluation of Results and Discussion

In order to evaluate the appropriateness of the designed models regarding the intended applications (visualization for communication of urban information to citizenry, and visualization for navigation purposes), a paper based questionnaire designed. The questionnaire consists total number of 24 multiple-choice questions to facilitate and accelerate answering procedure, accompanied with snapshots of the questioned 3D models. After conducting the evaluation, analyzing the results can provide us a clear overview regarding user preferences in 3D city modelling in different contexts. In the following subsections, we will discuss questionnaire structure and explain the deigned questions intention towards the final conclusions; and analyze the results, discuss and criticize them towards the existing hypothesis.

5.1 Developing the Evaluation Questionnaire

the main aim for developing a questionnaire in this thesis research was to evaluate the quality and the appropriateness of the designed 3D models towards the intended applications. Questionnaire evaluation helps us to formulate the qualitative criteria and facilitates the analysis procedure for further detailed investigations. Hence the questionnaire survey was designed with the main focus of the thesis in mind: 3D city modeling.

The initial questions have designed as a user modeling phase. Information acquired in this step create a general profile of the questioned user regarding his/her age, gender, education and his/her current occupation. Furthermore, these basic data will be used as catalyzer or filter in analysis phase of the evaluation to provide more insights regarding the obtained results and statistics.

Second stage of our designed questionnaire evaluates the appropriateness of the designed models regarding the landmarks recognition. After introducing some of the Munich's well-known historical and touristic landmarks, users were asked to mark the represented structure in order to evaluate 3D models efficiency in landmarks recognition. In addition to asking the users to actually mark the landmark which can be evaluated towards correctness, they were asked about how convenient was for them to find that particular questioned landmark in the model. Accompanied with two more questions to evaluate user's general satisfaction toward visualized model and their impression about touristic landmark's design quality, forms the second stage of the evaluation.

Landmarks recognition is an important factor for evaluating 3D city models efficiency towards communication and navigational purposes. In navigation, well-known city landmarks work as an exemplification for navigational signs and benchmarks, hence in the 3D environment a well-designed landmark or building can facilitate the navigation through dense city center environment. In addition to navigational purposes, a correct and accurate 3D city model is always preferred over an inaccurate one for communicating the city attractions to citizen. Three distinct sets of questionnaire compiled in order to evaluate recognition of three distinct landmarks in three different models to preserve the principle of randomness in our evaluation for further statistical analysis. It is worth to mention that an accurate and correct model in this case does not always indicates to a photorealistic and fully detailed model; the cartography community have advocated the use of more abstract 3D views for more efficient inference making. Also, Plesa and Cartwright (2008) found that users prefer non-photorealistic 3D maps for mobile navigation devices, because of their enhanced clarity and better usability.

The third stage is dedicated to evaluate the models towards navigation purposes; three snap-shots of three models in street view, indicating the same scene brought to the questioned users and they were asked about their preferred model for navigation (Figure 5-1).

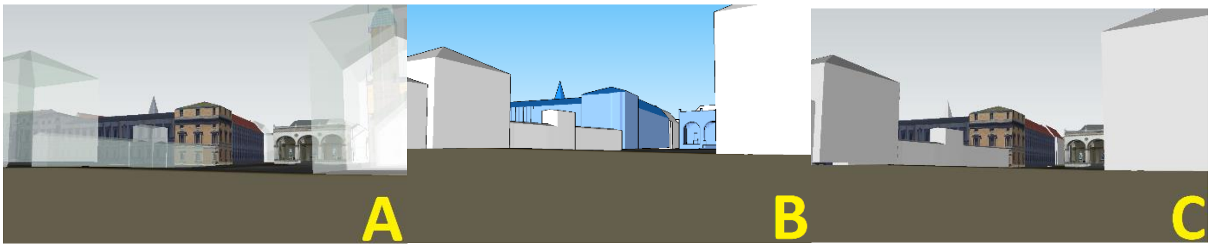


Figure 5-1: Evaluation of 3D models for navigational purposes.

In addition, the questionnaire asks the users about their preference in using 2D or either 3D models for navigation and more general; about their general desire in using 3D depictions for navigation purposes.

Final phase of the questionnaire investigates the appropriateness of visual variable size, for navigation and visual communication (Figure 5-2).

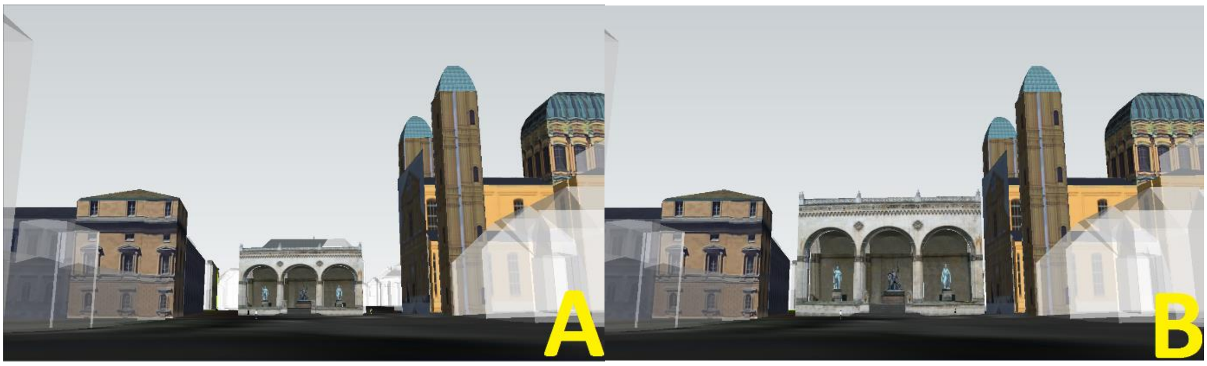


Figure 5-2: Evaluation of variable size, in 3D city modeling.

The first question pursuits questioned user's general impression regarding scaling a landmark building in the scene; the second question evaluates their preference regarding using a 3D model for navigational purposes; and the final query measure the importance of landmark's size accuracy for users in navigations.

5.2 Evaluation of the Results

Total number of twenty-seven people including thirteen females and fourteen males were asked to participate in the evaluation questionnaire. The students in different educational levels (Bachelors, Masters and PhDs) form the backbone of the total participants. In addition, the majority of the questioned participants age between twenty-three to thirty-four which indicates rather a young population sample in analytical-statistical language. 55.56 percent of participants claim to be familiar 3D city models as a distinct domain in the realm of geovisualization and cartography and 81.48 percent have used (at least once) the 3D city models for the following purposes (see Figure 5-3):

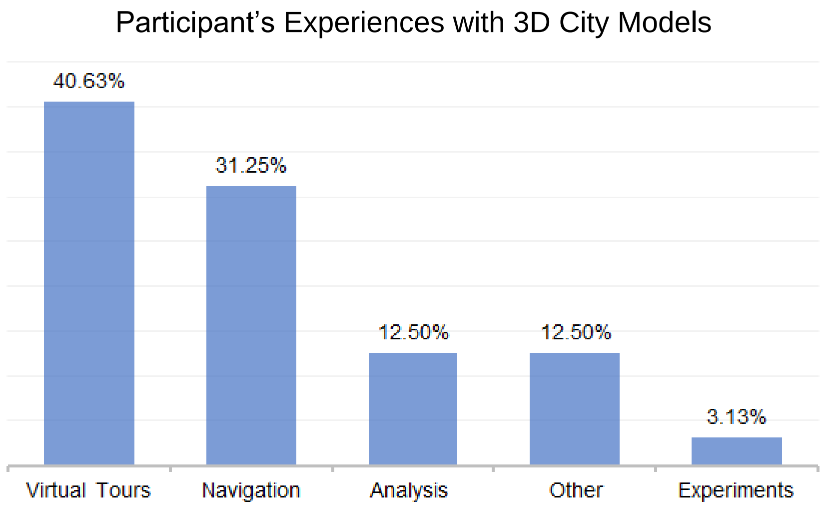


Figure 5-3: Participant's experiences with 3D city models.

As you can see in 5-3, the majority of 3D city models applications regarding the user's experiences is limited to 3D city virtual tours and 3D navigational applications; which are the two main applications we are investigating in our thesis project as 3D city models applications. It indicates that our chosen applications have decent overlap regarding the user's experience which can lead to more reliable and legitimate results for our questionnaire evaluation.

Figure 5-4 elaborates outcomes of the object recognition task in all three distinct models in a comparable fashion. Our strategy in design of visualization tools was, from the outset, to assist in both detecting and confirming proper matches and rejecting those which are irrelevant and incorrect.

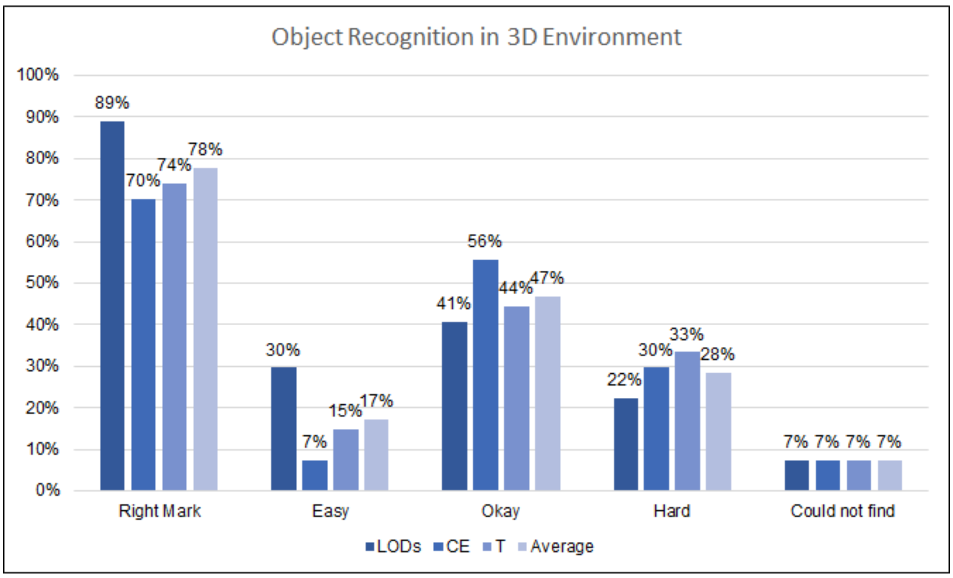


Figure 5-4: Object recognition results in 3D model environment.

Along indicating the correctness of the marks regarding the questioned landmarks in each model, an index of ease of finding requested landmarks according to the participant's experiences has been included in the figure. '*LODs'* indicates to our first model which contained photorealistic textures for landmarks and solid texture for the rest of the structures, '*CE'* represents the color-coded and edge enhanced model and '*T'* is related to the third model which introduces photorealistic textured landmarks accompanied with transparent irrelevant buildings; for simplicity reasons we will refer to the models in this chapter as 'model 1', 'model 2' and 'model 3' respectively. Model 1 can be introduced as the best model for land mark recognition, followed by model 3 with 74% and model 2 with 70% accuracy in the landmark recognition task. The only common visual element between model 1 and model 3 is photorealistic landmarks texturing which can be concluded as an important index for facilitating recognition process for the users. To the other words, it seems that solely edge enhancement and color-coding can more easily lead the users to the wrong recognitions regarding the questioned landmarks.

Following the mentioned conclusion, highest percentage in ease of finding landmarks as well as lowest percentage in adversity index is dedicated to the model 1, based on the participant's experiences with our designed models. Surprisingly model 3 has been introduced as the hardest 3D environment to allocate the questioned landmarks; this results have challenged the initial ideas in mixing the transparency effect with photorealistic texturing to guide the user's attention towards the intended landmarks and induction of the sense of ignorance regarding less important city structures.

Participants requested to elaborate their standpoints towards touristic landmarks visual appropriateness in terms of visualization aesthetics (see Figure 5-5).

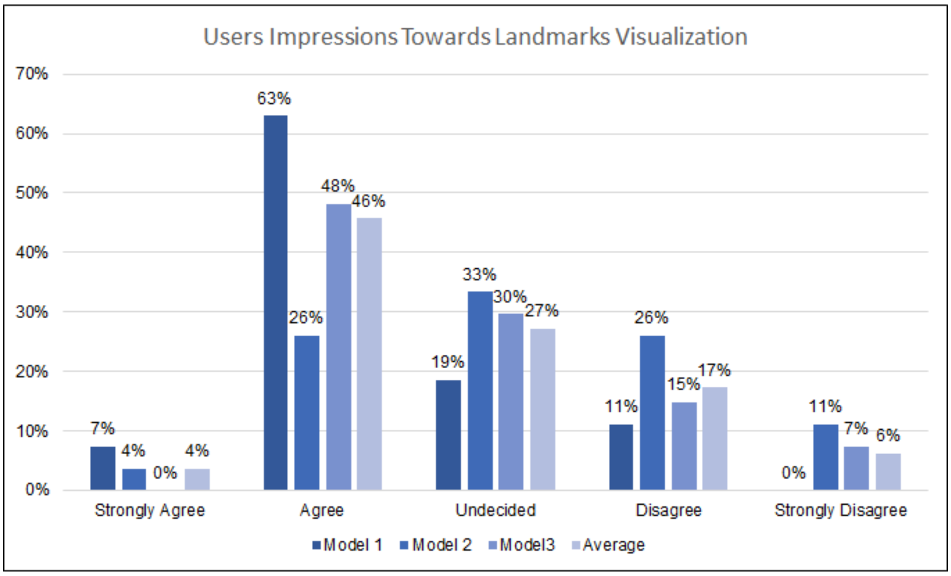


Figure 5-5: Users impressions towards landmarks visualization in models.

In confirmation to the results of the previous questions, users prefer the models 1 in terms of visualizing city landmarks. On the other hand, model 2 has entitled as the least appropriate model in terms of visualization aesthetics. This results once more emphasize the necessity of applying photorealistic textures for communicating urban information to citizenry. This indicates that enhancing the edges and applying abstract visual themes for visualizing geospatial data which has always been supported by cartography community does not always necessarily provides the best results in all applications.

User's satisfaction regarding the general appearance of the model has been illustrated in Figure 5-6.

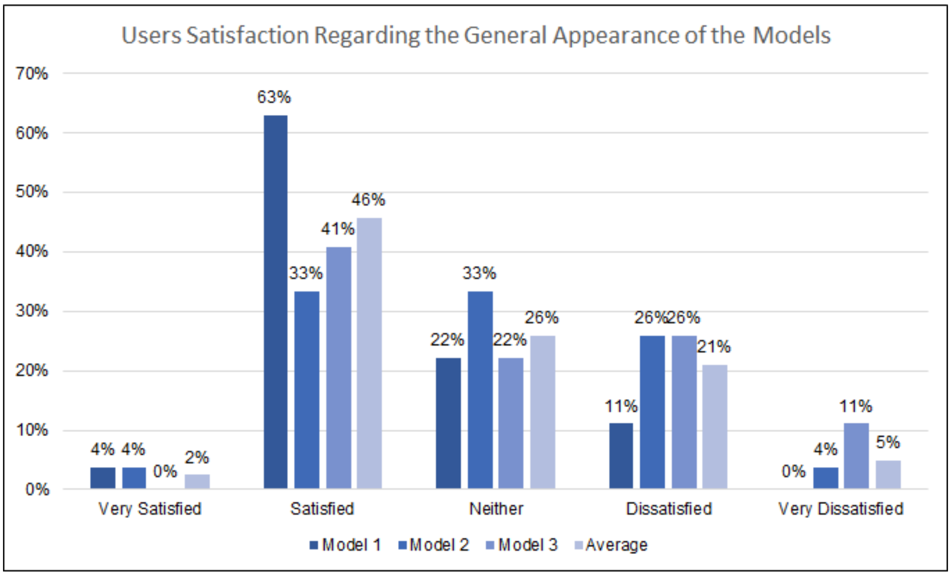


Figure 5-6: User satisfaction index regarding the general appearance of the models.

Similar to the user's impression regarding landmarks detection and their appearance, the preference for choosing the best model in terms of general satisfaction leans towards model 1. Models 2 and 3 have nearly same satisfaction levels; also the results indicate that model 3 has been slightly more successful for communicating the visual 3D information with participants.

Results of evaluating the designed 3D city models for navigational purposes have been visually elaborated in Figure 5-7.

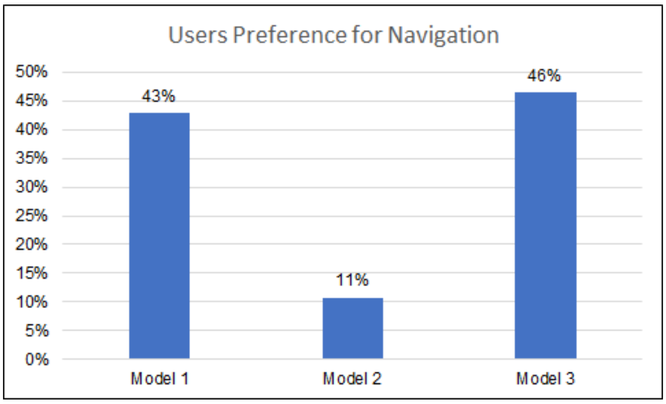


Figure 5-7: User preference over 3D models for navigation.

In contrast of being one of the least favorite models for communicating 3D landmarks with citizenry, model 3 has been introduced as the participant's favorite model for navigational purposes. Based on our findings through this questionnaire, photorealistic texturing of the landmarks has a tremendous effect on user's preferences for choosing the models; model 1 and model 3 by far are have been preferred over model 2 which emphasizes on color-coding and edge enhancement and more general, represents more abstracted model of the reality.

Participant's perspectives regarding utilization of 3D models and their preference in using 3D models over conventional (paper) 2D maps for navigational purposes has been visualized in Figure 5-8.

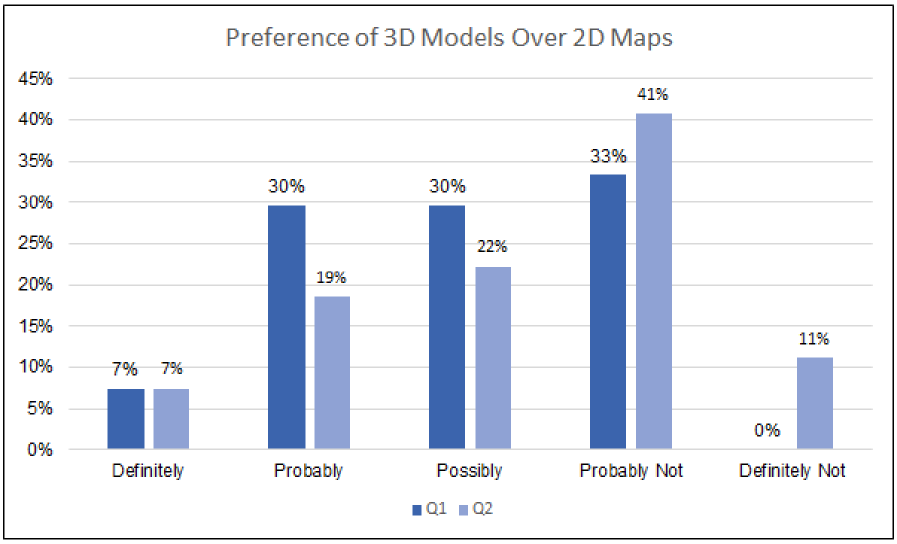


Figure 5-8: User preference regarding 2D maps or 3D models for navigation.

Q1 refers to question number 20: "*Will you (if available as an option) use a 3D map for navigation?*" and Q2 investigates: "*Will you prefer using a 3D map over a conventional 2D map for navigation?*". Roughly around 70 percent of participants indicate that they will at least possibly use 3Dmodels for navigation. There is a clear preference for using 2D maps for navigation when the both (2D maps and 3D models) option are available; More than half of the questioned users indicate that they will probably not prefer 3D models over 2D maps for navigational purpose.

Figure 5-9 indicates the appropriateness of using the variable *size*, in our intended 3D city modelling applications.

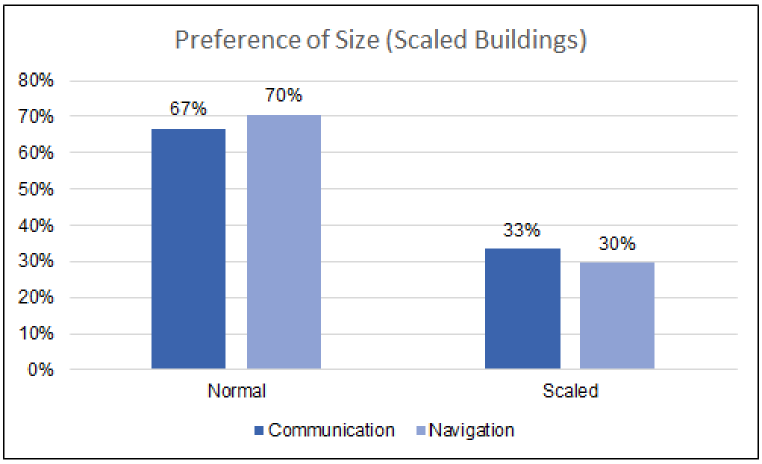


Figure 5-9: Evaluation of variable ‘*size*’ for urban information communication and navigational purposes.

In contrast to our initial hypothesis for applying the variable ‘size’ to emphasize the landmarks for a better communication of urban structures and to facilitate the navigation, users prefer the models in their actual size; roughly 70 percent of participants prefer the normal sized model over sized enhanced one for both applications. A part of these results is explainable with last question’s results (see Figure 5-10):

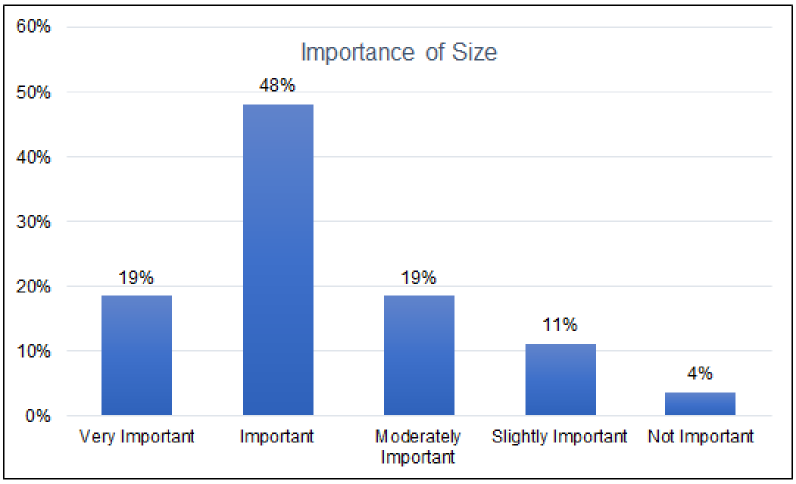


Figure 5-10: Importance of variable ‘size’ for questioned participants.

67 percent of questioned participants claim that the size accuracy has a significant importance for them when using 3D models. These results indicate the user’s preference in having a model with actual and reliable sizes rather than a size enhanced model which emphasizes landmarks with imaginary scales.

5.3 Discussion

Our results challenge some of the existing belief and findings in cartographic and geovisualization community indicating user’s preferences for using non-photorealistic 3D maps for navigational purposes (see (Plesa & Cartwright, 2008)) and more general, the idea to expand the conventional design principle in 2D environment which is based on the necessity of presenting an abstract view of the infinitely complex world to the 3D domain. To other words, based on our findings the common idea of using an abstract 3D view does not necessarily always guarantee more efficient inference making procedure by the user. Application domain and the intended platform for visualizing the 3D data are two major indexes which should be considered as a catalyzer while designing in 3D environment for refinement of the existing cartographic design principles.

On the other hand, user's tendency towards photorealistic models can be explained by their novelty to the 3D concept and restricted ability to handle 3D spatial data. One of the most important indexes for novice users in 3D data investigations is the concept of certainty; novel users prefer the models which better support the resemblance of their cognitive understanding of the 3D models to the reality. For a novice user photorealism denotes as higher data quality and as a result, provides higher degree of reliability and certainty rather than abstracted models. In addition, Schilling (2005) approved that photorealistic and abstract models are perceived differently by the user's cognition mechanisms; despite the user's novelty with 3D models, perception channel for processing the photorealistic and abstract models are distinct and independent.

The most common navigational strategy for general users is to recognize landmarks from the 3D model, but our results approves that participants prefer to have the landmarks in their actual size rather than magnified ones for navigation (70 percent); although scaled landmarks are easier to identify and follow in the dense urban environments. A part of this preference can be explained by the importance of size-accuracy for the users; more than 85 percent of the questioned participants stated that landmarks in their actual size is important (from very important to moderately important domain) for them while using 3D models for navigation. More generally, users indicate that the model with landmarks in their actual size is preferred (67 percent) over the scaled one; certainty and reliability once more can be a determining factor for explaining this preference. Models in actual size can provide more realistic and reliable depictions rather than their scaled counterpart hence are preferable.

Despite the technological developments in designing advance hardware and softwares, users still prefer conventional 2D maps over 3D city models and even 2.5D perspective maps for different applications e.g. navigation. More than half of the questionnaire participants (52 percent) straightly stated that they won't use 3D models for navigation in the case of having the both options available, 2D conventional maps and 3D models. Accompanied with 22 percent of the questioned users which are not sure about their neither of the options, up to possible maximum 74 percent of the users won't prefer 3D models over 2D maps. As we indicated in results section, shortcomings for providing a smooth and responsive 3D environment, wide data bandwidth requirement for downloading 3D data in comparison to 2D data, and insufficient proficiency and knowledge for interacting with 3D data can be outlined as some of the main reasons for the participant's preference in using conventional navigation medium. Some studies confirm finding some evidences which indicate that navigation strategies vary between males and females (Schilling et al., 2005); females, who might also less accustomed to using maps in general comparing to the males, would have found 3D maps with landmarks more useful and attractive.

In addition to providing more certainty, photorealistic texturing is the user's favorite visual variable in terms of using 3D models for navigation. The majority (89 percent) of the users prefer using models for navigation which include photorealistic textures over abstract model which contains edge enhanced and color-coded landmarks. Not only do novice users tend to favor realistic displays for various inference making tasks, but also more realistic looking displays seem to instill greater confidence in data quality (Zanola et al., 2009). Hence two distinct concepts, confidence in data quality and user certainty which will dramatically increase by using the visual variable, photorealism.

In general, nearly half (48 percent) of the participants are satisfied with general appearance of our models, by adding the undecided users, this number will be increased to 76 percent in the most optimistic case, which indicates our relative success in providing suitable models for first application; communicating the city landmarks with citizenry. 37 percent of questioned users state that they will use 3D models for navigational purposes which in the best condition it can be increased to the maximum 67 percent assuming that all undecided participants will prefer 3D models for navigation. This result indicates that 3D models have a long way to go to be accepted as the desired medium for navigational purposes and can be explained by different reasons; existing hardware and software shortcomings for providing a smooth and responsive 3D environment, wide data bandwidth requirement for downloading 3D data in comparison to 2D data, and insufficient proficiency and knowledge for interacting with 3D data can be outlined as some of the main reasons for the participant's preference in using conventional navigation medium.

In addition, a part of these results also can be explained by accessibility and availability; while conventional 2D maps are omnipresent and easily accessible in digital and paper formats, 3D models are not accessible in the paper format or any other similar formation (yet today, except the expensive Virtual Reality technology), and hardware limitations for visualizing and live tracking in 3D environment. On the other hand, based on our results, 3D city models have a good potential to be considered as the mainstream for communicating and visualizing the city features with the citizenry.

6 Conclusion and Outlook

The main aim of the thesis was to identify and examine design variables, and evaluate the possibility of applying conventional cartographic design variables in 3D environment. This research was an attempt to address graphical variables in 3D city designs and evaluate their appropriateness regarding different applications. The results of the evaluations challenged some of the accepted conventional concepts regarding the necessity of abstraction in visualizing real world phenomenon, indicates that all standard design principles in the 2D domain does not necessarily have the same impact in 3D environment; which again authenticates to the necessity of introducing new design concepts based on 3D environment specialties and reconcile the existing principles.

Technical phase of the thesis was mostly concentrated on taking advantage of the WebGL technology for visualizing the 3D cartographic production in Cesium virtual globe. Due to the novelty of using WebGL technology for visualizing 3D models on web browsers for the realm of cartography, there is no predefined coherent pipeline embracing the initial cartographic design steps to the final phases of embedding and visualizing the cartographic products on virtual globes. A comprehensive workflow which explains required softwares and converters for each distinct step to accomplish the intended strategies both in design and visualization was designed. Due to the novelty of 3D design in cartography there was a continuous effort to test the different available softwares, consider their supported formats and evaluate and identify the potential data losses during the conversion procedures from one format to the other. These shortcomings can be minimized with a closer interaction between cartographers and software developers for designing coherent procedures in 3D city designs, visualizations and interactions.

In general, this thesis took advantage of using the concept of virtual 3D globes for cartographic purposes and identified the possible challenges and perquisites to embed 3D models on the web environment. Virtual globes by providing dynamic interaction with 3D models, can promote the application of 3D models from just a communication tool to a gear which provides the required contexts for further analysis into the geospatial dataset; establishing new insights towards potential applications of 3D city models.

We tried to identify and apply meaningful variables for designing 3D models and evaluate their appropriateness regarding different applications. As we discussed before, there are limited number of literatures dedicated to identify, classify and evaluate graphical variables in 3D models environment; this area definitely need to be improved through future investigations. Cesium's processable format, glTF does not the ability to store and embed the semantic metadata; hence it is restricted exclusively to thematic information. One of the solutions for embedding CityGML database content to cesium is taking advantage of different data formats e.g. B3DM; But also this format partially supports sematic CityGML data. Future investigations can be concentrated on finding new methods for embedding semantic data to the glTF format through either internal or external formats and plugins. Developing such an infrastructure can promote the position of the 3D models to a sophisticated GIS analyzing tool for experts and aid the planners in decision making procedure.

Some experts suggest that navigational strategies differ between genders; in addition, conducting the evaluation in the actual 3D environment might result to different conclusions. Hence considering the participants gender in the evaluation phase can lead us to more reliable and authentic results. It is also recommended to conduct the 3D models evaluation in the actual computer 3D environment.

Further researches can investigate the optimal graphical variables for different applications of 3D city models and map them to the different applications accordingly; which can tremendously facilitate and accelerate the design phase of the city models.

References

Batty, M., Chapman, D., Evans, S., Haklay, M., Kueppers, S., Shiode, N.,Torrens, P. M. (2000). *Visualizing the city: communicating urban design to planners and decision-makers*.

Bell, D. G., Kuehnel, F., Maxwell, C., Kim, R., Kasraie, K., Gaskins, T., Coughlan, J. (2007). *NASA World Wind: Opensource GIS for mission operations.* Paper presented at the Aerospace Conference, 2007 IEEE.

Bertin, J. (1983). Semiology of graphics: Diagrams, networks, maps. Madison, Wis: University of Wisconsin Press..

Biljecki, F., Stoter, J., Ledoux, H., Zlatanova, S., & Çöltekin, A. (2015). Applications of 3D city models: State of the art review. *ISPRS International Journal of Geo-Information, 4*(4), 2842-2889.

Bollmann, J., & Koch, W. G. (2006). Lexical knowledge sources for cartography and GIS–development, current status and outlook. *Geodesy and Cartography*, 65(2), 259-270.

Bonham-Carter, G. Geographical information systems for geoscientists: modelling with GIS. New York, 1996. 398p. *Computer Methods in Geosciences, 13*.

Booch, G. (2005). *The unified modeling language user guide*: Pearson Education India.

Brewer, C. A. (1994). Guidelines for use of the perceptual dimensions of color for mapping and visualization. Paper presented at the IS&T/SPIE 1994 *International Symposium on Electronic Imaging: Science and Technology*.

Chaturvedi, K., Yao, Z., & Kolbe, T. (2015). Web-based Exploration of and Interaction with Large and Deeply Structured Semantic 3D City Models using HTML5 and WebGL. Paper presented at the *Wissenschaftlich-Technische Jahrestagung der DGPF und Workshop on Laser Scanning Applications*.

Cox, S., Cuthbert, A., Daisey, P., Davidson, J., Johnson, S., Keighan, E., Martell, R. (2002). OpenGIS® Geography Markup Language (GML) Implementation Specification, version.

Cozzi, P., & Bagnell, D. (2013). A webgl globe rendering pipeline *GPU Pro 4: Advanced Rendering Techniques* (pp. 39-48): AK Peters/CRC Press.

Cozzi, P., & Riccio, C. (2012). *OpenGL Insights*: AK Peters/CRC Press.

Cozzi, P., & Ring, K. (2011). *3D engine design for virtual globes*: CRC Press.

Crampton, J. W. (2001). Maps as social constructions: power, communication and visualization. *Progress in Human Geography, 25*(2), 235-252.

DiBiase, D. (1990). Visualization in the earth sciences. *Earth and Mineral Sciences, 59*(2), 13-18.

Döllner, J. (2001). Dynamic digital 3D maps. Paper presented at the *International Cartographic Conference ICC*, Bejing, China.

Döllner, J. (2007). Non-Photorealistic 3D Geovisualization. In W. Cartwright, M. P. Peterson, & G. Gartner (Eds.), *Multimedia Cartography* (pp. 229-240). Berlin, Heidelberg: Springer Berlin Heidelberg.

Döllner, J., Baumann, K., & Buchholz, H. (2006). Virtual 3D city models as foundation of complex urban information spaces. Reviewed paper. University of Potsdam, Hasso-Plattner-Institut.

Döllner, J., Kolbe, T. H., Liecke, F., Sgouros, T., & Teichmann, K. (2006). The virtual 3d city model of berlin-managing, integrating, and communicating complex urban information. Paper presented at the Proceedings of the *25th Urban Data Management Symposium UDMS (Urban Data Management Society)*.

Fulton, S., & Fulton, J. (2013). *HTML5 canvas*: " O'Reilly Media, Inc.".

Gede, M. (2015). Thematic Maps on Virtual Globes. Proceedings of *ICC2015, Rio de Janeiro.* http://www.icc2015.org/trabalhos/31/502/gede_thematic_maps_on_virtual_globes_icc2015. pdf.

Gröger, G., Kolbe, T. H., Czerwinski, A., & Nagel, C. (2008). OpenGIS city geography markup language (CityGML) encoding standard. *Open Geospatial Consortium Inc*, 1-234.

Häberling, C. (2003). Topografische 3D-Karten: Thesen für kartografische Gestaltungsgrundsätze: Swiss Federal Institute of Technology.

Häberling, C. (2005). Cartographic design principles for 3D maps–A contribution to cartographic theory*.* Paper presented at the Proceedings of *ICA Congress Mapping Approaches into a Changing World*.

Hallisey, E. J. (2005). Cartographic visualization: An assessment and epistemological review. *The Professional Geographer, 57*(3), 350-364.

Hughes, J. F., Van Dam, A., Foley, J. D., & Feiner, S. K. (2014). *Computer graphics: principles and practice*: Pearson Education.

Jobst, M., & Döllner, J. (2008). 3D city model visualization with cartography-oriented design. Reviewed paper.

Kolbe, T., Gröger, G., & Plümer, L. (2005). CityGML: Interoperable access to 3D city models *Geo-information for disaster management* (pp. 883-899): Springer.

Kolbe, T., Gröger, G., & Plümer, L. (2008). CityGML–3D city models and their potential for emergency response. *Geospatial information technology for emergency response, 257*.

Kraak, M.-J. (1988). Computer-assisted cartographical three-dimensional imaging techniques. TU Delft, Delft University of Technology.

Kraak, M.-J. (1999). Visualising spatial distributions. In: Longley, P.A. and et al. (2005) *Geographical information systems : principles, techniques, management and applications : abridged.* Hoboken : Wiley & Sons, 2005. 358 p. ISBN: 978-0-471-73545-8. pp. 157-173.

Kraak, M.-J. (2003). Geovisualization illustrated. *ISPRS journal of photogrammetry and remote sensing, 57*(5), 390-399.

Kraak, M.-J., & Ormeling, F. (2011). *Cartography: visualization of spatial data*: Guilford Press.

Longley, P. (2005). *Geographic information systems and science*: John Wiley & Sons.

Longley, P. A., Goodchild, M. F., Maguire, D. J., & Rhind, D. W. (2001). Geographic information system and Science. *England: John Wiley & Sons, Ltd*, 327-329.

MacEachren, A. M. (1994). Visualization in modern cartography: setting the agenda. *Visualization in modern cartography, 28*(1), 1-12.

MacEachren, A. M. (1995). *How maps work: representation, visualization, and design*: Guilford Press.

MacEachren, A. M., & Ganter, J. H. (1990). A pattern identification approach to cartographic visualization. *Cartographica: The International Journal for Geographic Information and Geovisualization, 27*(2), 64-81.

MacEachren, A. M., & Kraak, M.-J. (1997). Exploratory cartographic visualization: advancing the agenda. *Computers & Geosciences, 23*(4), 335-343.

MacEachren, A. M., & Kraak, M.-J. (2001). Research challenges in geovisualization. *Cartography and Geographic Information Science, 28*(1), 3-12.

MacEachren, A. M., & Taylor, D. R. F. (2013). *Visualization in modern cartography* (Vol. 2): Elsevier.

Peterson, M. P. (1995). *Interactive and animated cartography*: Prentice Hall.

Plesa, M. A., & Cartwright, W. (2008). Evaluating the effectiveness of non-realistic 3D maps for navigation with mobile devices. In Meng, L., & Zipf, A. (2005). *Map-based mobile services*. T. Reichenbacher (Ed.). p. 80-104. Berlin: Springer.

Rakshit, R., & Ogneva-Himmelberger, Y. (2008). Application of virtual globes in education. *Geography Compass, 2*(6), 1995-2010.

Ross, D.-I. L. (2010). Virtual 3D City Models in Urban Land Management. PhD Dissertation. *Technischen Universitat, Berlin*.

Schilling, A., Bolling, J., & Nagel, C. (2016). *Using glTF for streaming CityGML 3D city models.* Paper presented at the Proceedings of the 21st International Conference on Web3D Technology.

Schilling, A., Coors, V., & Laakso, K. (2005). Dynamic 3D maps for mobile tourism applications *Map-based Mobile Services* (pp. 227-239): Springer.

Schultz, R. B., Kerski, J. J., & Patterson, T. C. (2008). The use of virtual globes as a spatial teaching tool with suggestions for metadata standards. *Journal of Geography, 107*(1), 27-34.

Semmo, A. H., Dieter; Trapp, Matthias; Döllner, Jürgen. (2012). Concepts for Cartography-Oriented Visualization of Virtual 3D City Models. *Photogrammetrie, Fernerkundung, Geoinformation, 2014*(4), 0455-0466 (0411). doi:10.1127/1432-8364/2012/0131

Shreiner, D. (2009). OpenGL programming guide: the official guide to learning OpenGL, *versions 3.0 and 3.1*. Pearson Education.

Wood, M., & Brodlie, K. (1994). Visc And GIS: Some Fundamental Considerations in Visualization in Geographical Information Systems. HM Heamshaw and DJ Unwin (eds) John Wiley and Sons Chichester UK 3-8 Xiong D 1992 Traffic Flow Modeling with Visualization Tools. *In Proceedings GIS/LIS* (Vol. 92, No. 2, p. 844853).

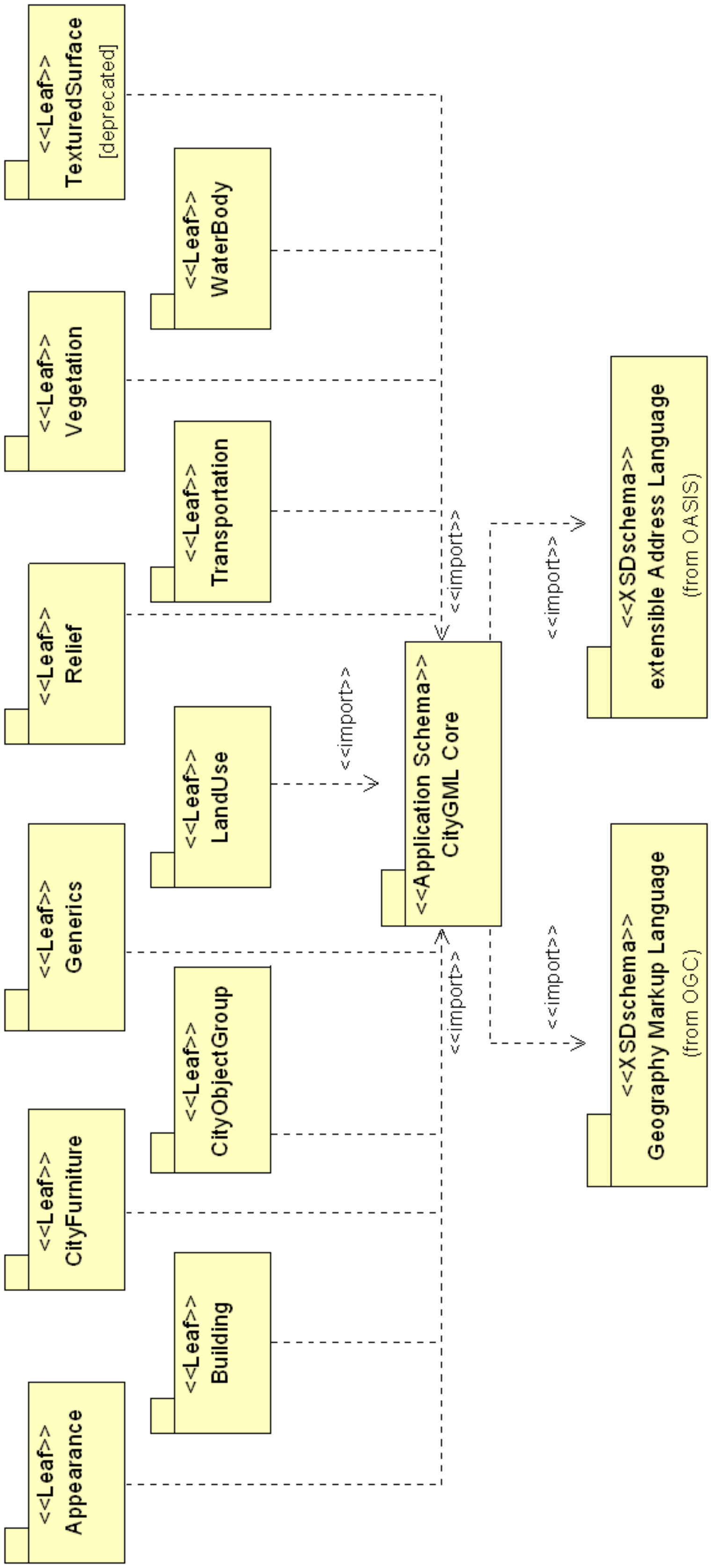
Zanola, S., Fabrikant, S. I., & Çöltekin, A. (2009). *The effect of realism on the confidence in spatial data quality in stereoscopic 3D displays.* Paper presented at the Proceedings of the 24th international cartography conference (ICC 2009), Santiago, Chile.

Zhu, Q., Hu, M., Zhang, Y., & Du, Z. (2009). Research and practice in three-dimensional city modeling. *Geo-spatial Information Science, 12*(1), 18-24.

Zlatanova, S. (2000). 3D GIS for urban development: International Inst. for Aerospace Survey and Earth Sciences (ITC).

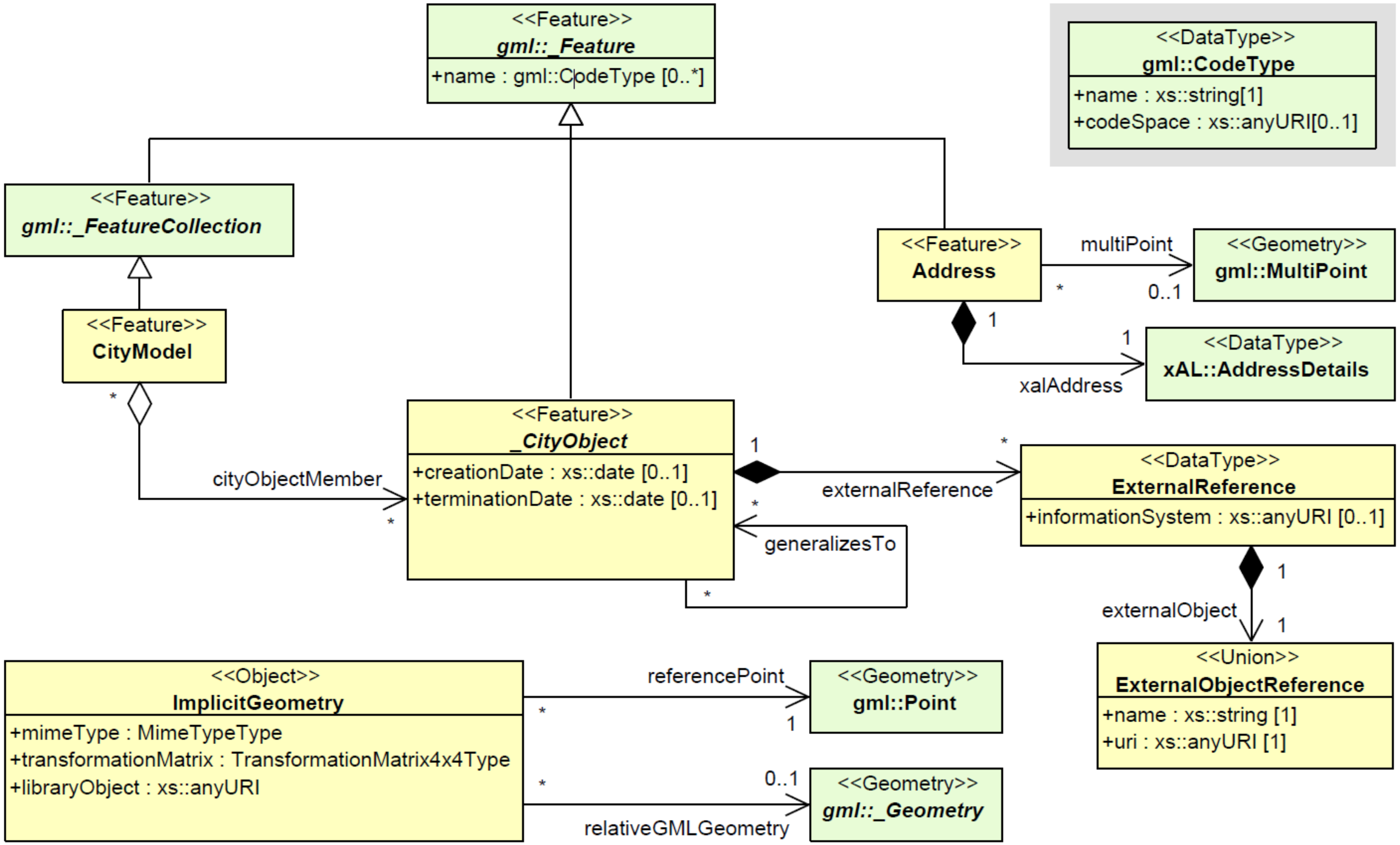
Zuk, T., Carpendale, M. S. T., & Glanzman, W. D. (2005). Visualizing Temporal Uncertainty in 3D Virtual Reconstructions. Paper presented at the *VAST (Visual Analytics Science and Technology)*.

Appendix A: UML package diagram of CityGML and their schema dependencies



<<Leaf>>
TexturedSurface
[deprecated]
<<Leaf>>
WaterBody
<<Leaf>>
Vegetation
<<Leaf>>
Transportation
<<Leaf>>
Relief
<<Leaf>>
LandUse
<<Leaf>>
Generics
<<Leaf>>
CityObjectGroup
<<Leaf>>
CityFurniture
<<Leaf>>
Building
<<Leaf>>
Appearance
<<import>>
<<import>>
<<import>>
<<Application Schema>>
CityGML Core
<<import>>
<<import>>
<<XSDschema>>
extensible Address Language
(from OASIS)
<<XSDschema>>
Geography Markup Language
(from OGC)

Appendix B: UML diagram of CityGML's core module



<<Feature>>
gml::_Feature
+name : gml::CodeType [0..*]
<<DataType>>
gml::CodeType
+name : xs::string[1]
+codeSpace : xs::anyURI[0..1]
<<Feature>>
gml::_FeatureCollection
<<Feature>>
CityModel
<<Feature>>
Address
multiPoint
<<Geometry>>
gml::MultiPoint
0..1
<<DataType>>
xAL::AddressDetails
xalAddress
<<Feature>>
_CityObject
+creationDate : xs::date [0..1]
+terminationDate : xs::date [0..1]
cityObjectMember
externalReference
generalizesTo
<<DataType>>
ExternalReference
+informationSystem : xs::anyURI [0..1]
externalObject
<<Union>>
ExternalObjectReference
+name : xs::string [1]
+uri : xs::anyURI [1]
<<Object>>
ImplicitGeometry
+mimeType : MimeTypeType
+transformationMatrix : TransformationMatrix4x4Type
+libraryObject : xs::anyURI
referencePoint
<<Geometry>>
gml::Point
relativeGMLGeometry
<<Geometry>>
gml::_Geometry

Appendix C: Evaluation Questionnaire Type I

Hello and welcome to the survey!

Thank you for agreeing to take part in this survey. As a critical part of my master thesis, this question based survey has been conducted for evaluating different cartographic design concepts in 3D city modeling. Concepts like size, texture, transparency, color and edge enhancement will be evaluated to identify the optimal design options in 3D city model visualizations which suits best to the users' expectations regarding their intended application. This survey should take only 5-6 minutes to complete. Be assure that all provided information by you will be kept in the strictest confidentiality and will be used only for further statistical analysis.

---

1. What is your gender?

☐ Male ☐ Female

2. How old are you?

☐ Under 16 ☐ 16-22 ☐ 23-28 ☐ 29-34 ☐ 35-40 ☐ above 40

3. What is your highest education level?

☐ High school/less ☐ Bachelors ☐ Masters ☐ PhD ☐ Higher

4. What is your occupation?

☐ Student ☐ Graduated ☐ Employee ☐ Retired ☐ Jobless ☐ Other

5. Are you familiar with 3D city models?

☐ Yes ☐ No

6-A. Have you ever used a 3D map?

☐ Yes ☐ No

6-B. IF yes please tell us for which purpose?

☐ Virtual city tours ☐ Navigation ☐ Analysis ☐ Experiments ☐ Other, …………..

Here is the picture of “Staatsoper München”:

Which is included in the following 3D model; In this model, touristic thumbnails of Munich are textured and other buildings are solid white blocks:

7. Please mark the “Staatsoper München” with a *circle* in the following model:

8. How convenient is for you to find the “Staatsoper München” in model above?

☐ Easy ☐ Okay ☐ Hard ☐ Couldn’t find it

---

9. How satisfied are you with this model?

☐ Very satisfied ☐ Satisfied ☐ Neither ☐ Dissatisfied ☐ Very dissatis-fied

---

10. Touristic thumbnails have well illustrated in this model?

☐ Strongly agree ☐ Agree ☐ Undecided ☐ Disagree ☐ Strongly Disagree

---

Here is the picture of “St. Peter's Church”:

Which is included in the following 3D model; In this model touristic attractions have blue colors and edge enhanced while hotels are orange:

11. Please mark the “St. Peter's Church” with a *circle* in the following model:

12. How convenient is for you to find the “St. Peter's Church” in model above?

☐ Easy ☐ Okay ☐ Hard ☐ Couldn’t find it

---

13. Touristic thumbnails has well illustrated in this model?

☐ Strongly agree ☐ Agree ☐ Undecided ☐ Disagree ☐ Strongly Disagree

---

14. How satisfied are you with this model?

☐ Very satisfied ☐ Satisfied ☐ Neither ☐ Dissatisfied ☐ Very dissatisfied

---

Here is the picture of “Theatinerkirche”:

Which is included in the following 3D model which touristic attractions are textured and other buildings are transparent:

15. Please mark the “Theatinerkirche” with a *circle* in the following model:

16. How convenient is for you to find the “Theatinerkirche München” in model above?

☐ Easy ☐ Okay ☐ Hard ☐ Couldn’t find it

---

17. Touristic thumbnails has well illustrated in this model?

☐ Strongly agree ☐ Agree ☐ Undecided ☐ Disagree ☐ Strongly Disagree

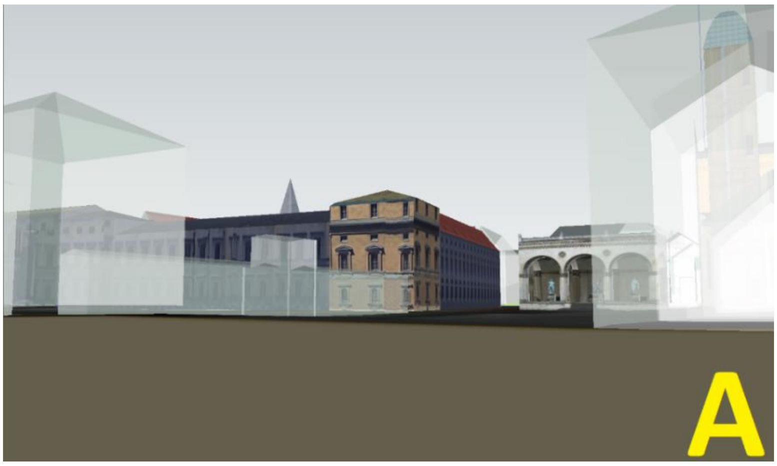
---

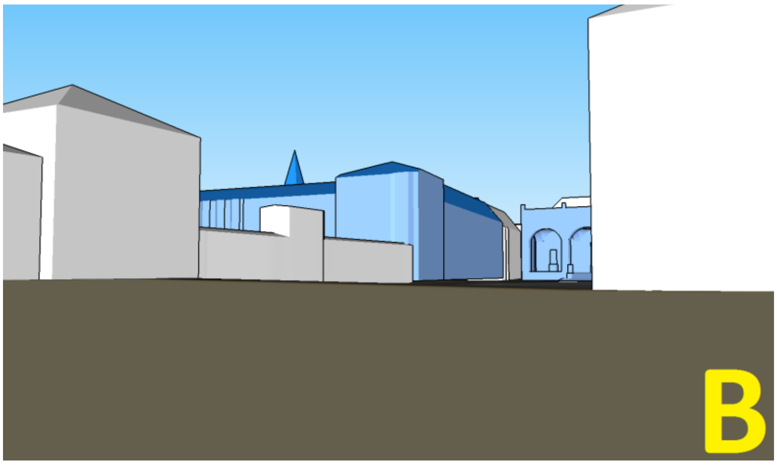
18. How satisfied are you with this model?

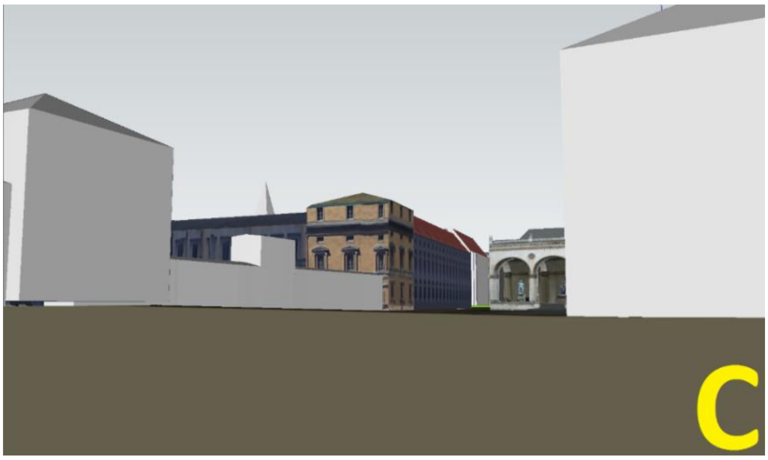
☐ Very satisfied ☐ Satisfied ☐ Neither ☐ Dissatisfied ☐ Very dissatisfied

---

19. Imagine you are using a 3D map in your mobile for navigation; which model do you prefer for navigation?







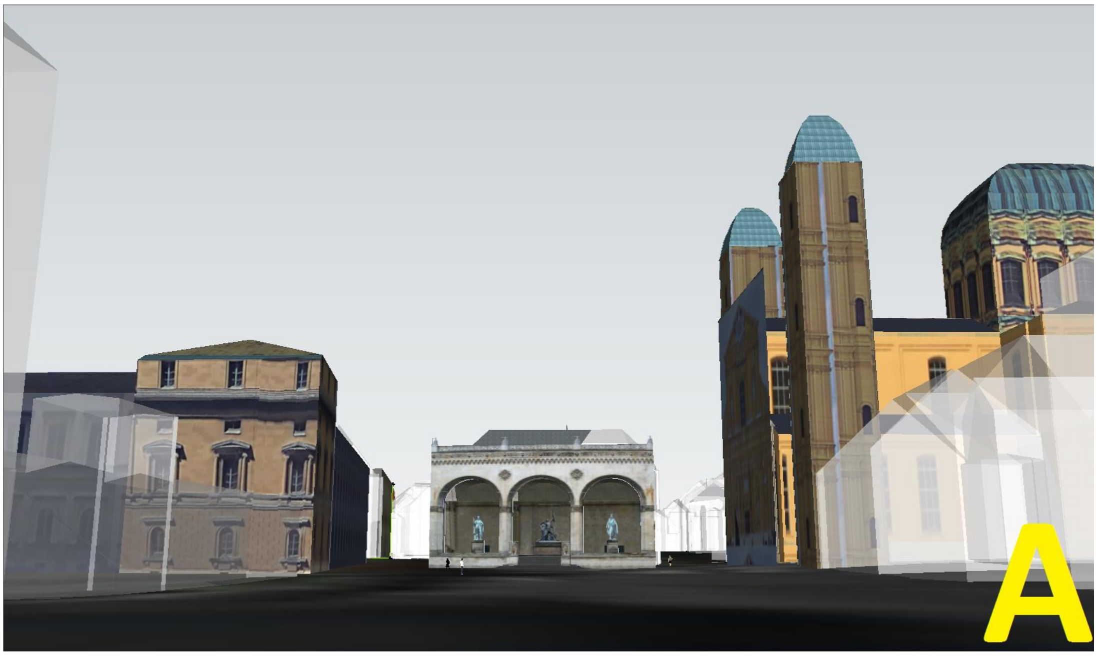
☐ A ☐ B ☐ C

20. Will you (if available as an option) use a 3D map for navigation?

☐ Definitely ☐ Probably ☐ Possibly ☐ Probably Not ☐ Definitely Not

21. Will you prefer using a 3D map over a conventional 2D map for navigation?

☐ Definitely ☐ Probably ☐ Possibly ☐ Probably Not ☐ Definitely Not



A contains “Feldherrnhalle” (gate shape monument in the center of picture with three statues) in actual size, but in the ***B*** size of the landmark has increased.

22. Which model looks better in your opinion?

☐ A ☐ B

---

23. Which model do you prefer for navigation?

☐ A ☐ B

---

24. How important is size-accuracy of buildings for you while using a 3D model for navigation?

☐ Very Important ☐ Important ☐ Moderately Important ☐ Slightly Important ☐ Not Important

---

Appendix D: CesiumJS Virtual Globe JavaScript syntax

```
function startup(Cesium) {
    'use strict';
//Sandcastle_Begin
var viewer = new Cesium.Viewer('cesiumContainer', {
    infoBox : false,
    selectionIndicator : false,
    shadows : true
});
function createModel(url, height) {
    viewer.entities.removeAll();
    var position = Cesium.Cartesian3.fromDegrees(11.576006, 48.139096, height);
    var heading = Cesium.Math.toRadians(135);
    var pitch = 0;
    var roll = 0;
    var entity = viewer.entities.add({
        name : url,
        position : position,
        model : {
            uri : url
        } });
    viewer.trackedEntity = entity;}
var options = [{
    text : 'LOD 1, 2, 3',
    onselect : function() {
        createModel('http://localhost:9000/Apps/SampleData/models/CesiumGround/1.gltf', 0);
    }}, {
    text : 'Color and Edge',
    onselect : function() {
        createModel('http://localhost:9000/Apps/SampleData/models/CesiumGround/4.gltf', 0);
    }}, {
    text : 'Texture, Size and transparency',
    onselect : function() {
        createModel('http://localhost:9000/Apps/SampleData/models/CesiumGround/6.gltf', 0);
    }}];
Sandcastle.addToolbarMenu(options);
//Sandcastle_End
    Sandcastle.finishedLoading();}
```