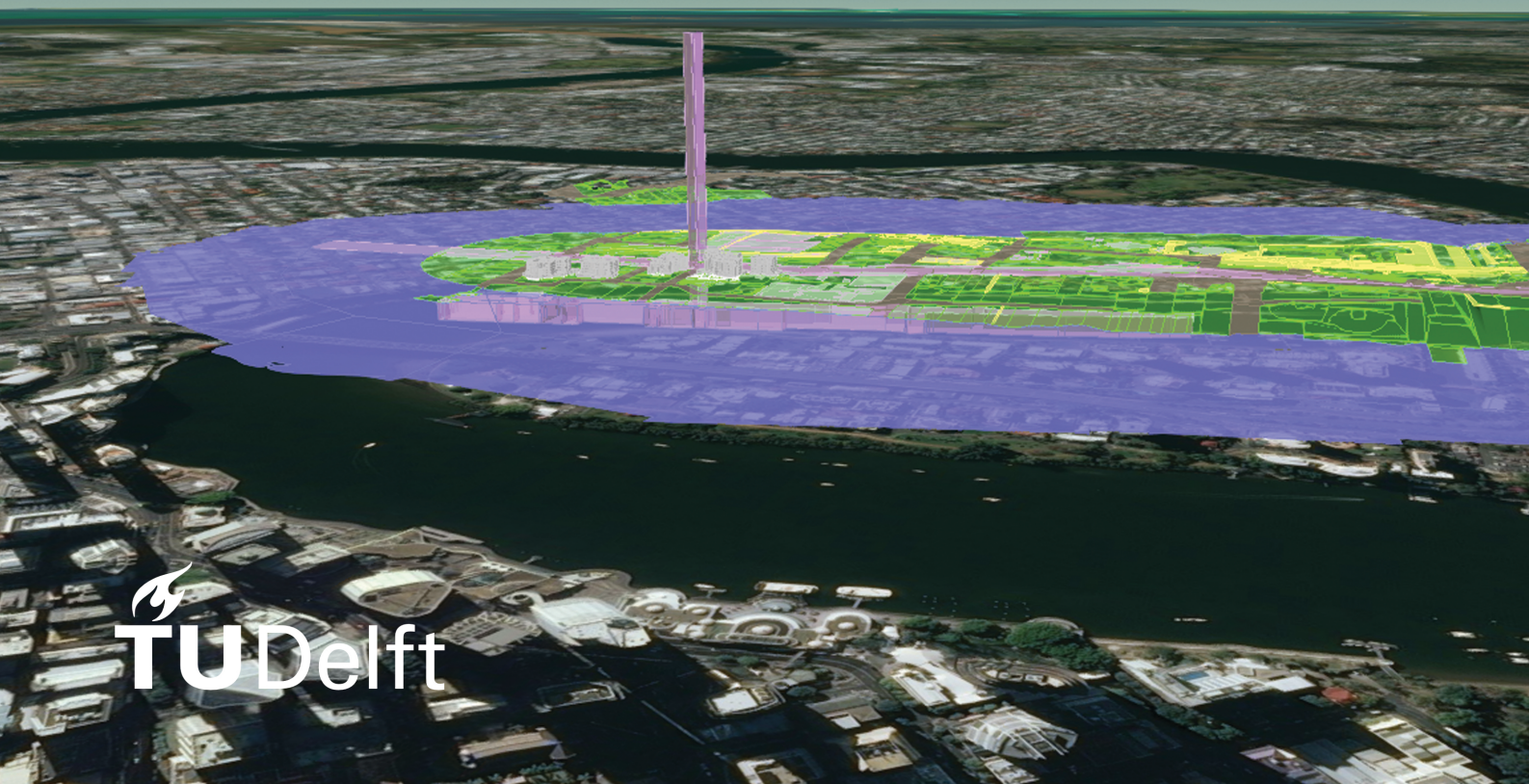


MSc thesis in Geomatics for the Built Environment

Web-based visualization of 3D cadastre

Barbara Cemellini
2018



WEB-BASED VISUALIZATION OF 3D CADASTRE

A thesis submitted to the Delft University of Technology in partial fulfillment
of the requirements for the degree of

Master of Science in Geomatics for the Built Environment

by

Barbara Cemellini

July 2018

Barbara Cemellini: *Web-based visualization of 3D cadastre* (2018)

© This work is licensed under a Creative Commons Attribution 4.0 International License. To view a copy of this license, visit

<http://creativecommons.org/licenses/by/4.0/>.

ISBN 999-99-9999-999-9

The work in this thesis was made in the:



Section GIS-technology
Department of the OTB
Faculty of Architecture & the Built Environment
Delft University of Technology

Supervisors: Prof. dr. ir. Peter van Oosterom
Drs. Marianne de Vries
Dr. Rodney J. Thompson
Co-reader: Prof. mr. dr. Hendrik Ploeger

ABSTRACT

Many countries in the world are extending their cadastral visualization systems in the third dimension. The reason of this, stems from the increasing complexity of contemporary cities, the growing 3D approach in other fields (including 3D spatial data acquisition, spatial data processing and visualization) which made 3D cadastre technologically feasible, and the need to overcome the issues of 2D visualization. The visualization of cadastral parcels in 3D is a huge challenge, since legal boundaries are, in some cases, invisible in the real world. Several countries built their own 3D cadastre (partial) systems/prototypes which are still lacking some crucial functionalities, therefore this research will try to fill some of the gaps of the existing cadastral systems with a special focus on visualization and dissemination aspects.

The goal of this research is to solve issues of occlusion, representation of unbounded volumes and ambiguous perception (in terms of position, size and shape) of objects in the context of 3D cadastre visualization. Additionally, the combination of topography and cadastral parcels is a double-edged sword; on one hand, it is useful for orientation purposes but, on the other hand, the growing complexity and increasing occlusion can make the visualization more challenging. The exploration of specific interaction techniques is fundamental to overcome these issues.

This document provides guidance on the system design choices to implement a 3D cadastre prototype. The study starts with a theoretical and a technological research carried out to investigate the state of the art in 3D cadastre visualization and to explore the existing WebGL platforms on which to build a successful prototype. Along with these phases, two lists of requirements are created; the first one related to 3D visualization issues and the second one related to the web viewer functionalities. The implementation of the prototype is carried out following (part of) the requirements listed, leaving the rest for future work. The development phase is the core of the research and includes the selection of the datasets and of the area of interest located in the city centre of Brisbane, Australia. In addition to that, the definition of the LADM compliant storage schema and the process of data encoding, to transform the data from survey plans to a cadastral database, are described. Although these steps are not in the main scope, they are crucial for the development of the prototype. After that, different visualization functionalities have been implemented, each of them presenting drawbacks and difficulties mainly related to the data format chosen.

In order to assess the usability and user-friendliness of the 3D cadastre prototype, a questionnaire has been handed out to potential users of the application. Their feedback is fundamental for this research and will provide insights on how to improve the actual design of the prototype.

ACKNOWLEDGEMENTS

I would like to take the opportunity to express my gratitude to a few people who played an important role in the development of this research.

First of all, I would like to thank my three supervisors for making me feel like part of a team, brainstorming together and working on different aspects of the same big project was very stimulating and made me very enthusiastic about implementing the prototype.

In particular, I would like to thank *Peter van Oosterom* for introducing me to this challenging thesis topic and for his invaluable advice throughout the whole academic year. Then, a big thanks goes to *Marianne de Vries* for all the technical support concerning the Cesium JS implementation. She was always ready to help me when things were going wrong. Next, I would like to sincerely thank *Rod Thompson* who worked on the data encoding software and joined our meetings from the other side of the world, despite the time difference.

Moreover, I would like to thank *Hendrik Ploeger*, for his critical and constructive advice as the external reader of this research and *Eftychia Kalogianni* for the precious suggestions on how to improve this thesis.

In addition, a big thanks to *Martin Kobbe* and *Pawel Michalak* from Fugro for the provision the digital terrain model of the region in question.

Finally, sincere thanks to my *family* and *friends* for their love and support during these two years abroad.

CONTENTS

1	INTRODUCTION	1
1.1	Motivation and problem statement	1
1.2	Basic principles of Land Administration	2
1.3	Objectives and research questions	3
1.3.1	Scope and framework of the research	4
1.4	Thesis outline	4
2	METHODOLOGY	7
2.1	The "Design Science Research" Approach	7
2.2	Research start-up	7
2.3	Literature and technology Research	7
2.4	Prototype development	9
2.5	Usability test	10
3	RELATED WORK	11
3.1	The Russian Prototype	11
3.2	The Queensland cadastre	12
3.3	The Dutch Cadastre	13
3.4	Other solutions found in the literature	14
4	REQUIREMENTS	17
4.1	Requirements related to 3D visualization challenges	17
4.1.1	Navigation tools and view controls	17
4.1.2	Tooltip	17
4.1.3	Integrating topography and reference objects	17
4.1.4	Transparency	18
4.1.5	Object selection	19
4.1.6	Object search	20
4.1.7	Wireframe display	20
4.1.8	Explode view	21
4.1.9	Sliding	21
4.1.10	Cross-section view	22
4.1.11	Visualization cues	23
4.1.12	3D measurement tools	23
4.1.13	3D buffer	24
4.1.14	Display partly unbounded objects and "complex geometries"	24
4.2	Requirements related to the web-based viewer as a visualization platform	25
4.2.1	Platform and browser independence	25
4.2.2	Handling massive data and caching/tiling between server and client	25
4.2.3	Layers control	26
4.2.4	Database support	26
4.2.5	Support different models (vector/polyhedral, raster/voxel, point clouds)	26
4.2.6	Support of 3D topographic visualization	26
4.2.7	Support for geo-referencing	26
4.2.8	Ensure spatial validity (3D vector topology)	27
4.2.9	Underground View	27
4.2.10	Open source platform	27
4.2.11	Possibility for the platform to be extended	27

4.2.12	2D overview map (orientation)	28
4.3	Implementation priorities	28
5	THE WEB BASED VIEWER	31
5.1	Analysis of the existing web viewers	31
5.1.1	iTowns	32
5.1.2	Cesium JS	32
5.1.3	OSM Buildings	33
5.1.4	WebGL Earth	33
5.1.5	GeoBrowser 3D	34
5.1.6	ESRI CityEngine Web Viewer	34
5.2	Comparison of the 3D Web viewers	35
5.3	Selection of the platform to develop the prototype	37
6	DEVELOPMENT OF THE 3D CADASTRE PROTOTYPE	39
6.1	Server-client system architecture	39
6.2	The datasets	40
6.3	The storage schema	41
6.3.1	The LADM compliant storage schema	42
6.4	The data encoding process	42
6.4.1	Encoding <i>building format</i> spatial units	42
6.4.2	Encoding <i>volumetric format</i> spatial units	44
6.4.3	Outcome of the encoding process	45
6.5	Selection of the DTM	46
6.5.1	The STK World Terrain	47
6.5.2	The DTM provided by Fugro	48
6.5.3	Conversion of the DTM	49
6.6	Display of 3D objects	49
6.6.1	About KML	50
6.6.2	Exploring the possibilities of KML (and other formats) in Cesium JS	52
6.7	Layers control	54
6.8	Camera start-up position	54
6.9	Subsurface visualization	54
6.9.1	Three alternatives to overcome the problem of subsurface visualization	55
6.9.2	Development of a new requirement: the dynamic elevation tool	58
6.10	Feature selection and highlight	59
6.10.1	Limitations of KML in Cesium JS	60
6.11	Tooltip functionality	61
6.11.1	Integration of fake RRRs and persons/parties in the database	63
6.11.2	Retrieval of administrative information from the database	63
6.12	Object search on administrative data	64
6.13	Summary of the requirements included in the prototype	65
7	USABILITY TEST	67
7.1	About usability and usability testing	67
7.2	Four steps to conduct usability testing	68
7.2.1	Step 1: Recruiting users	68
7.2.2	Step 2: Define goals and tasks to be performed	69
7.2.3	Step 3: Create a questionnaire	70
7.2.4	Step 4: Process results and obtain feedback	71
8	CONCLUSIONS AND FUTURE WORK	75
8.1	Conclusions	75
8.1.1	Research questions	75
8.1.2	Contribution in the field of Geomatics	80

8.1.3	Reflection and discussion	80
8.2	Future work	81
8.2.1	Refinement of the dynamic elevation tool	82
8.2.2	Refinement of the highlight functionality	82
8.2.3	Refinement of the object selection concept: the active set	82
8.2.4	Integration of administrative data (RRRs and persons/parties)	83
8.2.5	Towards XML structured 3D survey plans	83
8.2.6	Formal usability test	83
8.2.7	Virtual Reality and Augmented Reality	83
A	TECHNICAL SCHEMA	91
B	A VOLUMETRIC FORMAT SURVEY PLAN OF THE QUEENSLAND CADASTRE	97
C	JAVASCRIPT CODE	101
D	USABILITY QUESTIONNAIRE	113
E	REFLECTION	123

LIST OF FIGURES

Figure 1.1	The three main keywords of the thesis research	1
Figure 1.2	Advantages of a 3D, rather than a 2D cadastre representation [Zare Zardini and Hakimpour, 2015].	2
Figure 2.1	Design Science Research approach applied to 3D cadastre visualization	8
Figure 3.1	User interface of the Prototype for the Russian Federation [Elizarova et al., 2012]	12
Figure 3.2	Example of registration of 3D multi level property using 3D PDF. Starting from the top left corner and going in a clockwise direction we can see: the artistic impression of the property, the 3D view of legal spaces, floor plan of the 1st floor and a vertical cross section view. [Stoter et al., 2012a].	14
Figure 3.3	Interface of the ePlan prototype [Olfat et al., 2016]	15
Figure 4.1	Example of tooltip functionality showing attribute data of a cadastral parcel [Shojaei et al., 2013]	18
Figure 4.2	Use of transparency to enhance the difference between physical and legal objects [Pouliot et al., 2018]	19
Figure 4.3	Use of dynamic transparency to improve object discovery [Pouliot et al., 2018]	19
Figure 4.4	Example of object search as developed in the ePlan prototype for the state Victoria, Australia. The user can type the lot name, in this case "Lot Go3" and the system will highlight and zoom to the indicated lot (i.e. parcel) [Olfat et al., 2016].	20
Figure 4.5	Example of wireframe display [Shojaei, 2014].	20
Figure 4.6	Explode view example [Ying et al., 2016]	21
Figure 4.7	Sliding tool used to compare floor plans [Shojaei, 2014]	21
Figure 4.8	Sliding tool in the 3D cadastral prototype of the Russian Federation [Vandysheva et al., 2012]	22
Figure 4.9	Cross section view obtained by means of a vertical plane slicing the building in a vertical way. In this way, the interior of the apartment building can be better visualized [Shojaei et al., 2013].	22
Figure 4.10	Combination of <i>explode view</i> (3D displacement) and <i>cross-section</i> functionalities [Pouliot et al., 2018]	23
Figure 4.11	Adding visualization cues to a pipeline representation can help understanding depth and height of the object with respect to the earth surface. The red "sticks" represent the distance between the ground surface and the pipeline [Pouliot et al., 2018]	23
Figure 4.12	A spatial measurement tool calculating the distance between points as implemented in the ePlan prototype by the Land Use Victorian Government [Olfat et al., 2016]	24
Figure 4.13	Example of a 3D buffer around a pipeline object [Shojaei, 2014]	24
Figure 4.14	Example of underground view [Shojaei, 2014]	27
Figure 5.1	<i>WFS extruded on globe</i> : an application developed with iTowns [iTowns, 2017].	32
Figure 5.2	An application built on top of Cesium JS showing the buildings in New York city coloured according to their height value [Cesium, 2018].	33
Figure 5.3	Example of OSM Buildings [OSM buildings, 2017].	33

Figure 5.4	An example of WebGL Earth application [WebGL Earth, 2017].	34
Figure 5.5	An example application built on top of GeoBrowser 3D showing a wind turbine and the information related to it [Geobrowser 3D, 2017].	34
Figure 5.6	An example application using ESRI CityEngine Web Viewer [ESRI ArcGis, 2017].	35
Figure 6.1	Server-client system architecture diagram of the 3D cadastre prototype.s	40
Figure 6.2	Bounding box of the Brisbane City centre including the Story Bridge and the Kangaroo Point areas.	41
Figure 6.3	Database schema of the 3D cadastre prototype. Original tables from the Queensland Digital Cadastral DataBase (DCDB) are shown in light orange. New tables are shown in red, while the "views" are in blue.	43
Figure 6.4	Excerpt from a building format spatial unit [Thompson et al., 2018]	44
Figure 6.5	Excerpt from a 3D survey plan (SP192737)- Lot 822 [Thompson et al., 2018].	45
Figure 6.6	Excel sheet used for the encoding of Lot 822 (SP192737). The manual data entry is highlighted in yellow [Thompson et al., 2018].	45
Figure 6.7	Diagram showing the main phases carried out for the encoding process.	45
Figure 6.8	Building and volumetric format units imported in Google Maps [Thompson et al., 2018]. Note that, due to the impossibility of visualizing underground objects in Google Earth, all parcels have been raised 50 meters above ground.	46
Figure 6.9	Pop-up window containing the options that can be chosen to extract the KML files from the database.	47
Figure 6.10	STK World Terrain loaded in Cesium JS.	47
Figure 6.11	Elevation data of the Brisbane City centre in raster format (courtesy of Fugro).	48
Figure 6.12	Structure of the ESRI ASCII grid containing the DTM information.	49
Figure 6.13	Structure of the ASCII XYZ file containing the DTM information.	50
Figure 6.14	Example structure of KML encoding.	51
Figure 6.15	KML files imported in Google Earth.	52
Figure 6.16	Structure of a B3DM file [Schilling et al., 2016]	53
Figure 6.17	The 2D parcels are switched off, so that the 3D ones can be better visualized. Note: the "Switch 2D parcels on/off" checkbox is unchecked.	54
Figure 6.18	Excerpt from the JavaScript code used to set the camera initial position on Brisbane, Australia.	55
Figure 6.19	The underground parcels loaded in Cesium JS can be seen "from above" as if the ground surface was in transparent mode, but there are some limitations while zooming too close to the objects.	55
Figure 6.20	Cesium JS Ground Push Plug-in [Cesium, 2018]	56
Figure 6.21	Translucent terrain in Cesium JS [Cesium, 2018]	56
Figure 6.22	Using a fake surface for the visualization of subsurface earthquakes [Cesium, 2018]	57
Figure 6.23	Combination of fake surface and transparent terrain in the early stage of the development of the prototype.	57

Figure 6.24	Dynamic elevation tool: initial position of the parcels (before using the tool) and parcels shifted to 100 meters above the ground (after using the tool).	59
Figure 6.25	Zoom to underground parcels is possible thanks to the dynamic elevation tool.	59
Figure 6.26	Outcome of the highlight functionality on 2D parcels using the <code>scene.Pick</code> method.	60
Figure 6.27	Outcome of the highlight functionality on 3D parcels using the <code>scene.Pick</code> method.	61
Figure 6.28	Second attempt to implement the highlight functionality using the <code>scene.drillPick</code> method.	61
Figure 6.29	First implementation of the tooltip functionality using the default Cesium JS style.	62
Figure 6.30	This implementation of the tooltip functionality can be inconvenient in some cases.	62
Figure 6.31	Second implementation of the tooltip functionality using the highlight feature functionality.	63
Figure 6.32	Example of WFS response converted into a table, as implemented in the prototype.	63
Figure 6.33	WFS response to get the owner of the selected parcel (and some additional information).	64
Figure 6.34	Example of object search carried out on administrative data. The owner "Clem 7" has been entered in the search box and all the parcels owned by Clem 7 are now highlighted in red. Note that also the parcels underground are highlighted.	65
Figure 6.35	WFS response to get all parcels owned by a certain person/-party.	65
Figure 7.1	Percentage of usability problems found according to the number of test users. Plot of the curve for $L = 31\%$, where L is the proportion of usability problems determined from the testing of a single user [Nielsen, 2000].	68
Figure 7.2	Pie chart showing the percentage of representatives of the users groups who carried out the test.	72
Figure 7.3	Pie chart showing the knowledge of Cesium JS by the different users.	72
Figure 7.4	Bar chart showing the total points distribution.	74
Figure A.1	"Parcel" table definition with administrative and spatial indexes.	92
Figure A.2	"Polyhedron face" table definition with administrative and spatial indexes.	93
Figure A.3	"qc right" table definition with administrative index.	94
Figure A.4	"qc party" table definition with administrative index.	95
Figure A.5	"Owner of parcel" view definition	95
Figure A.6	"Parcel owned by" view definition	96
Figure B.1	Complete paper showing a survey plan of a volumetric format parcel - front page.	98
Figure B.2	Complete paper showing a survey plan of a volumetric format parcel - page 1/2.	99
Figure B.3	Complete paper showing a survey plan of a volumetric format parcel - page 2/2.	100
Figure D.1	Questionnaire for the assessment of the usability of the 3D cadastre prototype - page 1/7	114
Figure D.2	Questionnaire for the assessment of the usability of the 3D cadastre prototype - page 2/7	115
Figure D.3	Questionnaire for the assessment of the usability of the 3D cadastre prototype - page 3/7	116

Figure D.4	Questionnaire for the assessment of the usability of the 3D cadastral prototype - page 4/7	117
Figure D.5	Questionnaire for the assessment of the usability of the 3D cadastral prototype - page 5/7	118
Figure D.6	Questionnaire for the assessment of the usability of the 3D cadastral prototype - page 6/7	119
Figure D.7	Questionnaire for the assessment of the usability of the 3D cadastral prototype - page 7/7	120
Figure D.8	Responses of Question n.1 - Navigate to the Brisbane Airport and check where it is located with respect to the river. You can do this in two ways, if you know where the airport is approximately located just pan and zoom to the location, otherwise click on the magnifier icon and type "Brisbane Airport, Australia".	121
Figure D.9	Responses of Question n.2 - Which layers can be toggled on/off?	121
Figure D.10	Responses of Question n.3 - Suppose you want to take a look at the boundaries of the underground tunnel in our 3D Cadastral test area near the Brisbane city centre. In order to do that, you have to shift the ground surface and navigate around the parcels to visualise them in detail. What is the lowest z-value? (i.e. deepest point of the lowest 3D parcel below the surface)? Note: if you shifted the parcels you have to take in to account the amount of the shift.	121
Figure D.11	Responses of Question n.4 - Who is/are the owner(s) of the underground 3D tunnel parcels?	122
Figure D.12	Responses of Question n.5 - Search for owner "Annie" (note fake names in database for privacy). Which is the highlight colour used to show the parcels owned by Annie?	122

LIST OF TABLES

Table 4.1	Requirements priorities for 3D visualization	29
Table 4.2	Requirements priorities for the web-viewer	29
Table 5.1	Requirements check in the different WebGL platforms (3D visualization)	36
Table 5.2	Requirements check in the different WebGL platforms (Web viewer)	36

ACRONYMS

API	Application Programming Interface	32
ASCII	American Standard Code for Information Interchange	49
AR	Augmented Reality	83
B ₃ DM	Batched 3D Model	53
BIM	Building Information Model	42
CRS	Coordinate Reference System	52
DB	database	2
DBMS	DataBase Management System	12
DBMSs	DataBase Management Systems	26
DCDB	Digital Cadastral DataBase	xiv
DCDBs	Digital Cadastral DataBases	26
DEM	Digital Elevation Model	12
DTM	Digital Terrain Model	18
DTMs	Digital Terrain Models	18
FIG	International Federation of Surveyors	74
GeoTIFF	Geospatial Tagged Image File Format	49
gITF	GL Transmission Format	39
GUI	Graphical User Interface	58
IGN	Institut National de l'Information Géographique et Forestière	32
ISO	International Organization for Standardization	13
JSON	JavaScript Object Notation	39
KML	Keyhole Markup Language	14
LADM	Land Administration Domain Model	39
LAS	LASer File Format	48
OBJ	Relocatable Object Module Format	53
OGC	Open Geospatial Consortium, Inc.	34
PDF	Portable Document Format	9
RRRs	registration of Rights, Restrictions and Responsibilities	26
SQL	Structured Query Language	91
TIFF	Tagged Image File Format	12
TIN	Triangulated Irregular Network	41
UML	Unified Modeling Language	91
VR	Virtual Reality	83
WebGL	Web Graphics Library	9
WFS	Web Feature Service	39
x ₃ D	Extensible 3D Graphics	11
XLS	Microsoft Excel spreadsheet file	44
XML	Extensible Markup Language	11

1

INTRODUCTION

“Web-based visualization of 3D Cadastre” can seem a rather concise title for such an ambitious project. However, it conceals a number of keywords that describe the whole research. As shown in [Figure 1.1](#) the three main keywords are: *3D Cadastre*, *Web* and *Visualization*.

3D Cadastre combines the two main objects of this research, the Cadastre and its 3-dimensional representation. The word *Web* refers to the data dissemination aspect that this research is aiming to achieve. The last one is *visualization*, the main focus of the project.

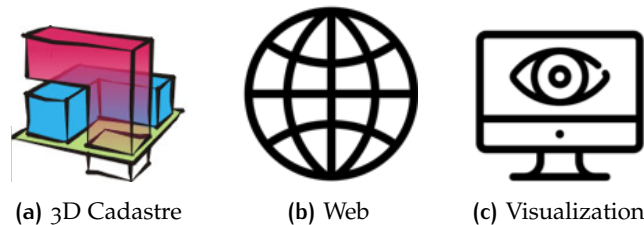


Figure 1.1: The three main keywords of the thesis research

1.1 MOTIVATION AND PROBLEM STATEMENT

In recent years, many countries have or are considering to extend their cadastral registration systems into the third dimension. The reason of this choice is given by three main factors: (1) the increase in private property values, (2) the higher density and raising complexity of buildings over the last 50 years (i.e. buildings above roads/railways, buildings leaning on top of each other, etc.) and (3) the growing 3D approach in other fields which made 3D cadastre technologically feasible [[Stoter and van Oosterom, 2006](#)].

The aim of cadastre is to make information available to everyone, thus sharing and visualizing it on the web would help to fulfill this purpose. An interactive 3D visualization system representing the third geometric dimension in a flexible way would give the user the possibility to explore the complexity of the 3D world and overcome most of the issues of 2D techniques [[Pouliot et al., 2018](#)].

The advantages of a 3D representation of cadastral data over a 2D representation are summarized in [Figure 1.2](#). The first image ([Figure 1.2a](#)) shows the difficulty to associate different owners to a unique 2D parcel, while the problem can be easily solved with a 3D representation. The second image ([Figure 1.2b](#)) presents a complex situation in which a road passes through a building creating an interference in the land rights. This can be solved with a 3D representation showing the situation in more detail. Finally, the third image ([Figure 1.2c](#)) represents a building and an underground infrastructure, in a 2-dimensional representation the underground objects would be hidden by the earth surface, while in a 3-dimensional one they can be easily visualized [[Zare Zardini and Hakimpour, 2015](#)].

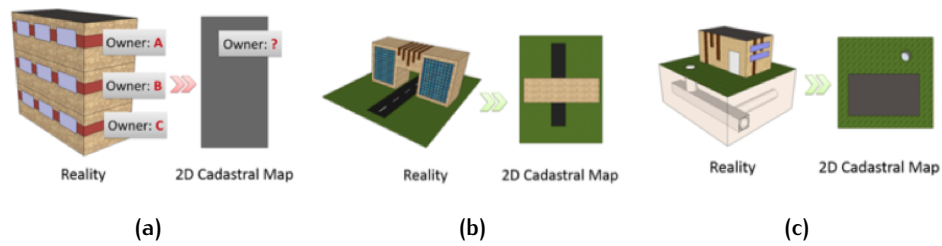


Figure 1.2: Advantages of a 3D, rather than a 2D cadastre representation [Zare Zardini and Hakimpour, 2015].

Often, cadastral parcels cannot be seen in the real world, they are defined by legal boundaries measured by surveyors and stored in some sort of database (DB), so the question is: *how can we properly represent something that is invisible to our eyes?* Undoubtedly, the creation of a proper 3D visualization of cadastral data presents several challenges.

The typical issues of 3D visualization such as occlusion and difficulties in the perception of position, size and shape of objects are combined with the issues related to 3D cadastre, like underground or unbounded volumes representing certain parcels. In addition, combining topography (i.e. earth surface and reference objects) and legal boundaries (i.e. cadastral parcels) contributes to create a more familiar view of the world. Although, this makes the visualization even more challenging due to the growing number of objects in the viewer.

1.2 BASIC PRINCIPLES OF LAND ADMINISTRATION

The concept of cadastre fits into the broader framework of land administration, which combines the cadastral and land registration domains.

Generally, a land administration system supports fiscal, juridical or mixed (both fiscal and juridical) activities. Therefore, it provides a framework for land taxation, it supports land registry and resolves ownership conflicts, and it is also used as a basis for land use planning [Henssen, 1995]. In addition, the primary purpose of land administration is to collect, store, retrieve, disseminate and use land related information [UN-ECE, 1996].

In general, four foundation principles related to land administration can be identified:

- The **booking or register principle** suggests that a change in land rights is not effective until it is booked or registered.
- The **consent principle** implies that the entitled person registered in the book must give a consent for a change in the land register.
- The **principle of publicity** entails that legal registers must be open for public inspection and they are (roughly) considered correct.
- The **principle of specialty** implies that the persons and the properties contained in land registration documents must be unambiguously identified.

The 3D cadastre prototype that will be developed for this research is, in the first place, connected with the last two principles. On one hand, the principle of specialty states that the land must be unambiguously identified, thus the use of 3D cadastre, instead of the 2-dimensional one, would help to eliminate uncertainties regarding ownership rights, etc. On the other hand, the principle of publicity implies that the information must be disseminated, and therefore a web based solution would be ideal for reaching the target users [Henssen, 1995].

Among the target users we can find researchers in the field of cadastre, professionals (such as architects, engineers, notaries, and so on), managers in the government and in the public administration, public and private entities, and companies operating in different fields. Nevertheless, also non-specialized users should be able to easily use the application; among them we can list the general public and citizens who just want to take a look at their own or other properties.

1.3 OBJECTIVES AND RESEARCH QUESTIONS

The objective of this MSc research is to overcome the challenges described in [Section 1.1](#), keeping as main goal the visualization and dissemination aspects of 3D cadastre, through the development of a web-based prototype. Therefore, the main research question for this thesis is:

What is a good system design to obtain a proper, clear and not misleading visualization of 3D cadastral parcels through a geo-web viewer?

A *good* system design is meant to solve the main challenges of 3D cadastre visualization by meeting specific requirements. Furthermore, a good cadastral system must be able to properly support land administration and, more specifically, cadastral activities by providing a straightforward and user-friendly interface. The system should assist professionals in performing the fiscal and juridical tasks typical of the cadastre domain. In addition, since the prototype has also the goal of disseminating the information, the system should address the needs not only of the professional users such as researchers, managers in the public administration and so on, but also of the ordinary citizens in order to ensure transparency in the land administration procedures.

In order to answer the main question, five different subquestions have been identified, each of them addressing a specific issue:

1. *How to tackle issues such as occlusion and ambiguous perception (in terms of position, size and shape) of objects while interacting with 3D Cadastre?*
This subquestion aims at investigating issues affecting 3D visualization in general.
2. *What is the best way to combine the visualization of legal boundaries (cadastral parcels) and reference objects (topography)?*
For orientation purposes, it is important to include reference objects to help us understand the location of the parcels in the real world. Although, adding more objects to the viewer makes the visualization more challenging.
3. *Which interaction techniques should be used in order to enhance user experience in 3D cadastre applications?*
The focus of this subquestion is on the interaction with the platform from a user specific point of view.
4. *What are the options to store and transfer the data? (DBMS or encodings such as XML, JSON, glTF, etc.)*
Many cadastral prototypes are still struggling with this issue, therefore evaluating the possibilities will help to choose the most effective option.
5. *What is a good system architecture for 3D cadastre visualization?*
The system architecture includes: communication protocols, encodings, shipping, caching and data storage. In particular, at the server side it consists of the database and web components and at the client side the visualization and interaction components.
Therefore, a good system architecture is meant in terms of user experience,

quality of the data (i.e. availability of up to date information), use of **DB** or files according to the situation, flexibility in changes in the data structure, ability to handle both small and large datasets.

It is important to keep in mind that, due to time constraints, not all the sub-questions will be answered with an in-depth analysis and concrete results, some of them will remain purely theoretical and will be used as a starting point for further research in the future. In particular, part of subquestion n.4, regarding the use of a database to store the data, and subquestion n.5, regarding the system architecture, will be explored more theoretically.

1.3.1 Scope and framework of the research

The scope of the MSc thesis is to implement a prototype for the visualization and dissemination of 3D cadastre through a web-based viewer. The focus will be mainly on visualization, in particular, on how to solve the visualization issues of 3D parcels. Other concerns related to 3D Cadastre such as, how to solve topology issues (i.e. overlap, intersection, etc.), efficient storage of the data, speed and performance of the whole system are out of scope.

Moreover, this research fits into a wider framework that goes beyond visualization and dissemination of 3D cadastral data and sees the collaboration of a team including Dr. Rodney J. Thompson, Drs. Marian de Vries and Prof. dr. ir. Peter van Oosterom. Dr. Rodney J. Thompson provided the data from the Queensland cadastre, he also worked on a Java code to convert the survey plans from paper to digital format and together with Peter van Oosterom and Marian de Vries developed an LADM compliant schema for the Queensland cadastre. In addition, Marian de Vries contributed to the implementation the prototype thanks to her developer skills.

1.4 THESIS OUTLINE

Now that the reader has an idea of the topic and motivation behind this thesis research, it is time to present a short overview of the content of the following chapters. The document is organized as follows:

Chapter 2 contains the theoretical **METHODOLOGY** that will be used to carry out the research and the development of the prototype.

Chapter 3 makes an overview of the **RELATED WORK** in order to present the state of the art in the field of 3D Cadastre visualization in different parts of the world. The strengths and gaps are taken as an inspiration for this research.

Chapter 4 contains the identified **REQUIREMENTS**, both related to 3D visualization and to the web viewer implementation. Each of them is explained in detail and some examples from existing prototypes and literature are shown.

Chapter 5 focuses on **THE WEB BASED VIEWER**. The available options based on WebGL technology are described and compared in order to find the optimal solution for the implementation of the prototype.

Chapter 6 is the core of the research, or rather the **DEVELOPMENT OF THE 3D CADASTRE PROTOTYPE**. The chapter documents the main decisions taken and the consequent results. The focus is mainly on the issues and on the technological limitations that interfere with the implementation.

Chapter 7 introduces the **USABILITY TEST** carried out to obtain design feedback on the 3D cadastre prototype.

Chapter 8 closes the document with **CONCLUSIONS AND FUTURE WORK** by answering the research questions, discussing the contribution in the field of Geomatics and giving some inputs for the future implementation.

Furthermore, the following appendixes aim at integrating the description with additional details.

Appendix A incorporates the technical schema in combination with the conceptual schema described in the document.

Appendix B contains the full document of a volumetric format survey plan of the Queensland cadastre.

Appendix C contains the whole JavaScript code written to develop the prototype.

Appendix D includes the full transcript of the questionnaire used for the usability testing.

Appendix E incorporates the mandatory reflection.

2 | METHODOLOGY

This chapter outlines the theoretical approach designed to solve the 3D visualization and dissemination issues of 3D cadastre. Having a clear and defined methodology is crucial for conducting the research, for the correct implementation of the prototype and to understand the workflow.

[Section 2.1](#) introduces the "Design Science Research" approach used to define the methodology of this MSc research. After that, [Section 2.2](#), [2.3](#), [2.4](#) and [2.5](#) explain in detail the different phases of the method chosen.

2.1 THE "DESIGN SCIENCE RESEARCH" APPROACH

The research scope and objectives have been defined in [Chapter 1](#), therefore the next step is to outline a clear methodology for the thesis research. The methodology will address all the research questions and it will give an overview of the phases of the project. The approach followed is based on development and performance of designed artifacts, the so-called "Design Science Research" approach as described by [Hevner and Samir \[2010\]](#). [Figure 2.1](#) displays a flowchart of the overall process of visualization and dissemination of 3D cadastre. The main phases are:

- Research start-up
- Literature and technology research
- Prototype development
- (Usability test)

A detailed description of each phase will be given in the following sections.

2.2 RESEARCH START-UP

The research start-up phase includes the definition of the graduation topic along with the supervisors, the definition of the research questions and related subquestions, and a preliminary draft of the research to be carried out.

Once the main goal has been defined, the research narrows down to more specific issues.

2.3 LITERATURE AND TECHNOLOGY RESEARCH

The literature study focuses on existing 3D cadastre applications which have a similar purpose than this MSc thesis research. Many countries all over the world are moving towards the implementation of a three dimensional cadastre. The aim of this research is to investigate on what has been done and on what are the gaps that need to be filled in the future. This will give insights about which directions the research should follow.

Therefore, the literature research phase is divided into three main steps as shown in [Figure 2.1](#):

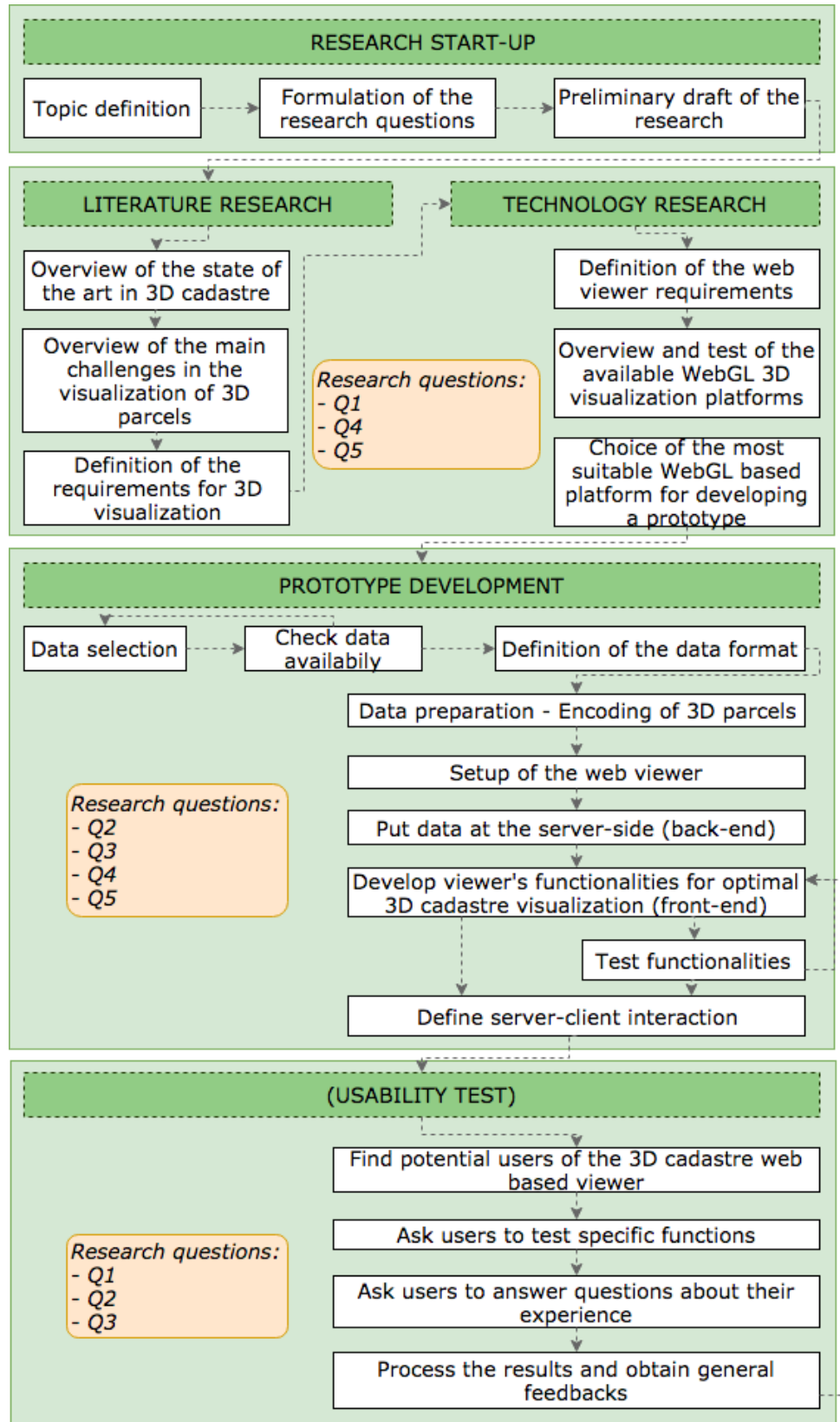


Figure 2.1: Design Science Research approach applied to 3D cadastre visualization

- An overview of the state of the art in 3D cadastre visualization with a focus on the 3D cadastre prototype for the Russian Federation, the Queensland cadastre, and the Dutch cadastre, as described in [Chapter 3](#).
- An overview of the gaps and challenges that need to be solved to obtain a proper visualization of 3D cadastral parcels, also contained in [Chapter 3](#).
- The definition of a set of requirements that need to be met to overcome the visualization issues, described in detail in [Section 4.1](#).

On the other hand, the technology research focuses on the available Web Graphics Library ([WebGL](#)) 3D visualization platforms that can be used as a framework for the 3D cadastre prototype, as explained in [Chapter 5](#). The choice of WebGL stems primarily from its plug-in free interface which runs on the most common web browsers, making it an efficient and cutting edge technology. [Shojaei \[2014\]](#) states that a good knowledge of existing platforms and their capabilities can help building a successful cadastral prototype. Accordingly, the following steps will be carried out:

- First, a list of requirements related to the web viewer functionality is produced. The detailed description of the requirements is in [Section 4.2](#).
- Then, an overview, test and comparison of the available [WebGL](#) platforms is performed to find out which one is the most suitable for the implementation of the prototype. A description of this can be found in [Section 5.1](#) and [Section 5.2](#).
- Finally, the selection of the most suitable platform is made, according to the predefined requirements. The purpose of the study is to find the platform with the highest number of requirements already implemented and/or the possibility to modify its source code in order to implement additional functionalities. The process of selection is described in [Section 5.3](#).

As shown in [Figure 2.1](#), the literature and technology research aim at answering three of the research subquestions: *Q1*, *Q4* and *Q5* listed in [Section 1.3](#).

2.4 PROTOTYPE DEVELOPMENT

The core of the MSc research is the development of the 3D cadastre visualization prototype taking into account the tailor-made list of requirements. The workflow can be subdivided in several stages as shown in [Figure 2.1](#):

- The choice of the data is crucial to build a reality-like prototype. In this case, the dataset chosen is part of the Queensland [DCDB](#), in particular, the area of the Brisbane city centre. Generally, the availability of the data is one of the main issues, thus several iterations could be necessary before the data is selected. For an overview of the datasets chosen see [Section 6.2](#). Once the data is chosen, a data format compatible with the 3D visualization platform should be determined.
- Since the data obtained is not always matching the desired data format, it can be necessary to convert or encode the information in a different format before using it. The 3D survey plans of the Queensland [DCDB](#) are only available in Portable Document Format ([PDF](#)), therefore the data needs to be encoded in a digital format to be stored in a spatial database (see [Section 6.4](#) for a detailed description of the encoding process). It is implicit that, although it is not the main goal of the MSc research, the data preparation/encoding phase is necessary for the realization of the prototype.

- After the data has been prepared, the selected 3D visualization platform (client) needs to be set up on a local machine, so it can run on the web browser.
- The prepared data is stored in a [DB](#) at the back-end, so that it can be retrieved by the client during interaction.
- The implementation of the functionalities will follow, according to the two types of requirements previously identified. Most likely, only a part of the requirements will be implemented due to time constraints. The functionalities will then be tested making sure that they solve common 3D visualization problems, such as occlusion, distortion and difficulties in the perception of position, size and shape of objects. For a more detailed description of the prototype development see [Chapter 6](#).
- The last step is the definition of the client-server interaction to ensure a wide dissemination of the cadastre application. In particular, this phase will investigate on which is the best system architecture (i.e. communication protocols, encodings, shipping caching and data storage) for this purpose.

As can be seen in [Figure 2.1](#), the prototype development phase is investigating the topics related to most of the subquestions: *Q2*, *Q3*, *Q4* and *Q5* listed in [Section 1.3](#).

2.5 USABILITY TEST

The usability test phase is shown between brackets at the bottom of [Figure 2.1](#), since its implementation is uncertain due to time constraints.

The idea is that potential users of the 3D cadastre prototype are identified and they are asked to perform specific tasks, in order to assess the user friendliness and usability of the web viewer. The user groups need to be as diverse as possible in order to assess the functionalities in different fields.

The test users will have to fill in a questionnaire about their experience with the prototype and the results will be processed to obtain feedback for the further development of the 3D cadastre prototype.

In order to fasten the process, a possibility would be to conduct “informal” usability tests on a couple of target groups: people working in the Geomatics domain at TU Delft (including Geomatics students) and general public/citizens. These groups would try out the newly built prototype and give their opinion about it. The limitation of this approach is that the users are not covering all the user groups to which the prototype is addressed to. For a description of the usability test, see [Chapter 7](#).

The usability test phase is investigated in the following subquestions: *Q1*, *Q2* and *Q3* listed in [Section 1.3](#).

3

RELATED WORK

Several 3D cadastre prototypes have been implemented in different countries [Pouliot et al., 2018], however there are still big challenges with respect to interaction, visualization and dissemination.

In this chapter the relevant literature related and linked to the project is presented. The goal is to have a general overview of the state of the art in 3D cadastre, with a focus on visualization. The related work will be described highlighting the gaps that the thesis research is aiming to fill.

Section 3.1 analyses [The Russian Prototype](#) for the visualization of 3D Cadastre, based on the Extensible 3D Graphics (X3D) standard format. Section 3.2 describes [The Queensland cadastre](#) from its early days until today. Section 3.3 introduces the 3D PDF solution used by the Dutch cadastre. Finally, Section 3.4 concludes the chapter with a number of similar prototypes developed worldwide.

3.1 THE RUSSIAN PROTOTYPE

The 3D cadastre prototype for the Russian Federation was developed to display both 3D objects and their legal boundaries. The prototype is supported by a web browser based solution running on Internet Explorer and Firefox, after the installation of the *BS Contact plug-in* from Bitmanagement [Vandysheva et al., 2012], which allows the visualization and interaction with the 3D objects.

The 3D geometry is stored in static X3D files, while the corresponding administrative information is stored in additional Extensible Markup Language (XML) files [Elizarova et al., 2012].

The user interface is composed of three main parts: the *3D Viewer*, the *Select* pane and the *Selection Results* pane (Figure 3.1). The *3D Viewer* is where the interaction takes place; the user can rotate, zoom, pan, switch features on and off and use some special tools for viewing more details. The *Select* pane contains different options for the selection and visualization of 3D parcels. The *Selection Results* pane shows information about selected objects [Elizarova et al., 2012].

The prototype supports a wide variety of functions for 3D cadastre visualization, among them we can list:

- Display and interact with 3D objects (rotate, pan, zoom, select, etc.);
- Selection based on different attributes such as, owner name, id of cadastral object, address, etc.;
- Possibility to toggle the visibility of layers;
- Possibility to show/hide privacy data based on user's right of access;
- Possibility to toggle the visibility of elements in the 3D model (walls, floors plants, etc.);
- Switch the language of the interface between Russian and English.

One of the main challenges during the development of the prototype was the link between the 3D objects in the viewer window (X3D files) and the corresponding

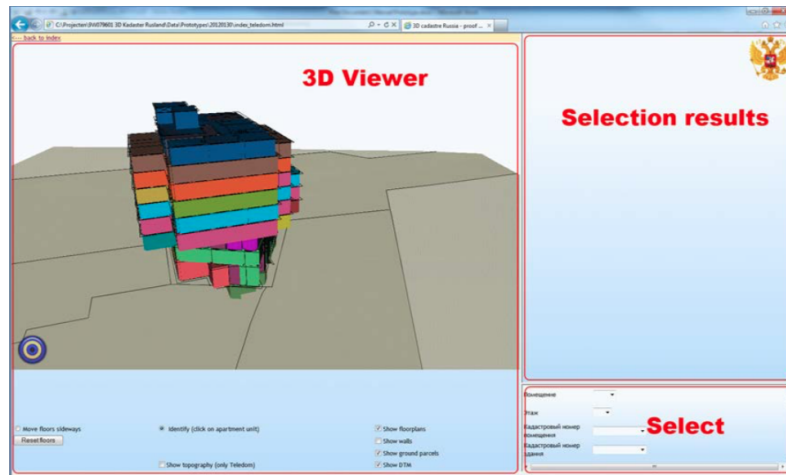


Figure 3.1: User interface of the Prototype for the Russian Federation [Elizarova et al., 2012]

legal cadastral data in the non-spatial part of the prototype (XML files) [Vandyshcheva et al., 2012]. Another challenge was to match the 3D models with the 2.5D GIS reference data containing ground parcels and a Digital Elevation Model (DEM) with the terrain elevation [Elizarova et al., 2012].

After the implementation of the prototype, the involved parties decided to make a questionnaire to assess the prototype functionalities and determine future directions. The questionnaire was composed of 44 questions both open and multiple-choice, aimed at the evaluation of the testing made by the participants. The outcome of the questionnaire showed a positive attitude to the future introduction of a 3D cadastre in the Russian Federation [Elizarova et al., 2012].

Two main gaps need to be filled:

- First, the Russian Prototype requires the installation of a plug-in in the web browser to read the X3D files. This makes the dissemination of the application more challenging, since some users could have difficulties with its installation.
- Second, the information is not stored in a unique place, i.e. a DataBase Management System (DBMS), but in separate XML and X3D files. Having the information subdivided into different files makes the data update and management more challenging and error-prone. Storing the information in a database will prevent this to happen and would ensure the data to be always up to date and available to different users.

3.2 THE QUEENSLAND CADASTRE

In the past, the state of Queensland, Australia was maintaining a land administration system based on Torrens System of registration [Queensland Government, 2009]. The early Cadastre was just a simple register of areas of land but, since 1997 3D unit rights, timeshare and 3D volumes has been included. Nowadays, the database maintained by the state, the so called Digital Cadastral DataBase (DCDB) contains records of all spatial units, although only the footprints of the 3D parcels are stored. The actual 3D survey plans are stored on scanned PDF files or Tagged Image File Format (TIFF) [Thompson et al., 2018].

Moreover, the cadastral geographic dataset of Queensland has a *base layer* which consist of a non-overlapping coverage containing parcels, roads, railways, water bodies and intersection parcels [Stoter, 2004].

Even though the Queensland legislation is already oriented to the implementation of 3D cadastral parcels, there are still some gaps that need to be filled:

- First of all, the fact that the 3D parcels are laid down on paper (or on scanned PDF/TIFF documents) is a mayor shortcoming, since the data cannot be interactively viewed.
- The 3D survey plans only show the boundaries of the parcels, without any integration with the surroundings.
- It is not possible to check the validity of the 3D objects and their topological relationships. The parcels are described without the use of 3D primitives, but only by metes and bounds on the drawings.
- Currently, the data model of the Queensland DCDB is not described according to the International Organization for Standardization (ISO) standard Land Administration Domain Model (ISO 19152) and it does not support 3D parcels and reference objects [Thompson et al., 2018].
- The last issue is related to the fact that the 3D information is not integrated with the 2D cadastral map, but it is stored in separate survey plans. Therefore, it would be good to combine this information with the cadastral registration system which is already well implemented [Stoter, 2004].

3.3 THE DUTCH CADASTRE

The Dutch cadastre proposes a solution based on two distinct phases: the first one fits within the current cadastral and legal framework, the second one proposes a more long term solution [Stoter et al., 2012a,b].

The first phase of the 3D cadastre implementation in the Netherlands includes a separate level with spatial units combined with a 3D drawing. This solution has been implemented within a short timeframe because it did not foresee significant changes and fitted in the current cadastral and legal framework [Stoter et al., 2017]. In the current implementation of this first phase, it is no more allowed to subdivide a 2D parcel due to the projection of the 3D volumes on the cadastral map, because this would lead to an unclear visualization. The original parcel will be maintained and a 3D detailed representation will be created as a 3D PDF drawing of the parcels, which is submitted via ELAN system. A 3D PDF is a simple PDF containing 3D geometry [Adobe Acrobat, 2018]. It should be noted that, this is only possible for the newly built constructions of which digital drawings are available.

The advantage of the 3D PDF is that it provides helpful tools to understand 3-dimensional situations; it allows interaction such as zooming, rotating, sliding, and the possibility to query the data [Stoter et al., 2017]. An example of registration of a 3D unit using 3D PDF is shown in Figure 3.2.

Despite the advantages mentioned, this approach has some drawbacks that need to be solved:

- This solution focuses primarily on visualization, therefore the 3D data is not registered.
- A consequence of the previous point is that, the 3D geometries cannot be topologically validated; thus overlapping situations and not closed (i.e. invalid) volumes can occur.
- Data cannot be altered easily, so it might be necessary to recollect the data with risk of discrepancies.
- Only the parcels in the deed are shown, it is not possible to visualize the cadastral database with neighboring parcels.

On the other hand, the second phase is addressed to a long-term 3D registration and it is still at a conceptual level. This phase foresees the implementation of a full 3D cadastre solution including traditional parcels, unbounded parcels, multi-level properties [Stoter et al., 2012b].

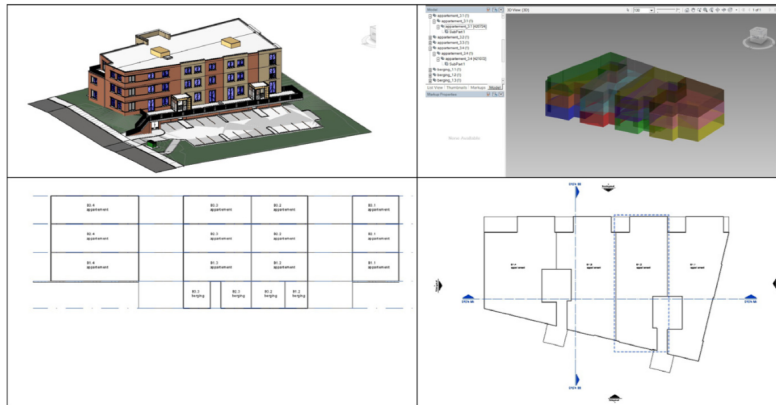


Figure 3.2: Example of registration of 3D multi level property using 3D PDF. Starting from the top left corner and going in a clockwise direction we can see: the artistic impression of the property, the 3D view of legal spaces, floor plan of the 1st floor and a vertical cross section view. [Stoter et al., 2012a].

3.4 OTHER SOLUTIONS FOUND IN THE LITERATURE

Besides the examples described above, which are used as a starting point for this thesis research, other open source and proprietary prototypes were built for 3D cadastral systems [Pouliot et al., 2018], both web and desktop based:

The Victorian digital cadastre

The Digital Cadastre Protocol (ePlan) was introduced in 2011 in the state of Victoria, Australia. The web-based prototype of the 3D ePlan was developed by the Land Use Victorian Government to replace paper and PDF plans with digital files. The main purpose of this prototype is to enable the users to interactively visualize a 3D property on the web. The ePlan/LandXML files can be visualized after being converted in Keyhole Markup Language (KML) format to be input in Google Earth.

The main advantage of this prototype is the ability to create very clear and understandable views for the users, while the main shortcoming lies in the fact that the underground visualization is still an issue [Olfat et al., 2016]. Figure 3.3 shows the interface of the prototype.

Two 3D cadastre web map prototypes for land administration and spatial planning in Indonesia

Aditya et al. [2011] developed the 3D Cadastre web maps for space management based on KML with Google Earth and for disaster management activities based on X3D with ArcGIS online. Two innovative solutions that still present some issues. The space management prototype manifests some problems in the integration between the web database and the 3D visualization components, returning misleading information. On the other hand, the disaster management prototype is based on X3D, so it requires the installation of a plug-in in the web browser [Aditya et al., 2011].

The 3D cadastre of Shenzhen

Differently from the previous examples in which web-based solutions have



Figure 3.3: Interface of the ePlan prototype [Olfat et al., 2016]

been presented, Guo et al. [2013] developed a desktop-based 3D cadastre addressed to the administration of urban land use in Shenzhen, China. The prototype has been implemented on a desktop version of Google Earth. Although this is a rather innovative solution, it does not seem ideal for dissemination purposes, since it is meant to be used on desktop and not on the web [Guo et al., 2013].

4 | REQUIREMENTS

The **REQUIREMENTS** chapter contains two different groups of requirements: the functional/platform independent requirements related to the challenges of 3D visualization (in [Section 4.1](#)) and the requirements related to the web viewer (in [Section 4.2](#)). The first group refers to the functionalities aimed at enhancing 3D visualization to overcome issues such as, occlusion and ambiguity in the perception of position size and shape of objects. The second group concerns the characteristics that the platform chosen should have in order to support visualization of 3D cadastre. Note that, some requirements are closely related to each other.

Finally, since the list is rather long, some implementation priorities will be defined in [Section 4.3](#).

The content of this chapter is part of a previous paper by [Cemellini et al. \[2018\]](#) written in occasion of the *FIG Congress 2018* held in Istanbul, Turkey.

4.1 REQUIREMENTS RELATED TO 3D VISUALIZATION CHALLENGES

In order to define what is the best way to visualize cadastral data in 3-dimensions, it is important to analyze the existing literature and define a list of requirements.

In this section, the requirements concerning 3D visualization of cadastre will be listed and described with examples taken from existing prototypes and related work in the field.

4.1.1 Navigation tools and view controls

One of the main requirements for a proper 3D visualization of cadastral data is interaction by means of navigation tools and view controls. Among them, we can list traditional functionalities of 3D viewers: zoom in and out, pan, rotate and tilt. These features help the user to better understand the 3D environment by interacting with it [[Shojaei et al., 2013](#)].

4.1.2 Tooltip

The tooltip functionality, which shows the attribute data linked to an object, could be convenient in cadastral systems. Administrative information about legal objects is usually in the form of text, so it can be efficiently shown using a tooltip [[Shojaei et al., 2013](#)]. [Figure 4.1](#) presents an example of tooltip to display attribute data of a cadastral parcel.

4.1.3 Integrating topography and reference objects

Visualizing legal boundaries without reference objects, buildings and ground surface would result in an abstract representation; this could lead to misunderstandings in the perception of the actual boundaries. Physical objects give reference to interpret legal objects in terms of location and size [[Kalogianni, 2016](#)], therefore it is good practice to combine them together in cadastral representations. Nevertheless,

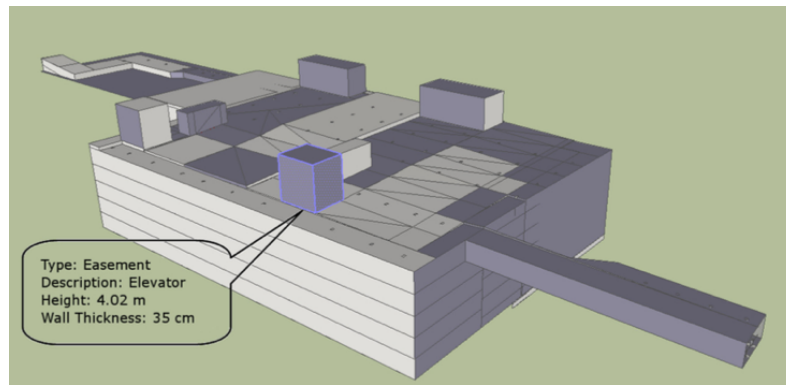


Figure 4.1: Example of tooltip functionality showing attribute data of a cadastral parcel [Shojaei et al., 2013]

there is the danger that the visualization will result in further complications due to the higher number of objects [Pouliot et al., 2018].

Among these reference objects or surfaces we can list Digital Terrain Models (DTMs), 3D building models, urban utility networks, road networks, railways and rivers.

DTMs are needed to represent the different elevations of the earth surface (i.e. topography). For the purpose of 3D cadastre, parcels can be placed above, below or on the ground surface, therefore the Digital Terrain Model (DTM) helps defining a reference for the correct visualization of the parcels. It is very common to combine a DTM with satellite or aerial images to have a more detailed view of the world. The accuracy of the DTM and of the parcel vertices must be checked to avoid situations in which parcels are “flying” above ground or are partially hidden by the earth surface (when this should not be the case).

3D building models represent the structure of a building, together with its associated components. For 3D cadastral purposes, both the external and internal geometry are required to represent the ownership of parcels.

Urban utility networks include pipelines, cables, sewage systems, storm-water drainage and so on. They are important from a legal point of view to facilitate the understanding of ownership boundaries and from a physical point of view for the maintenance of infrastructures [Shojaei, 2014].

Vandyshva et al. [2012] described the 3D cadastre prototype for the Russian Federation as a combination of 3D cadastral parcels, a DTM, 2D parcels, 2D and 3D reference objects (topographic data) and digital images.

Furthermore, the possibility of switching on and off layers containing different types of reference objects gives more flexibility to the visualization. This functionality is closely related to the layer control functionality, described in Section 4.2.3.

4.1.4 Transparency

Another requirement is transparency, which can be interpreted as opacity/transparency at different levels. Transparency can be applied in order to show the interior of a 3D model. This feature is also convenient to visualize and distinguish between physical and legal objects [Vandyshva et al., 2012], for example, in Figure 4.2 two levels of transparency have been used to enhance the difference between legal boundaries and buildings.

Similar techniques have also been explored, for instance, *dynamic* transparency is able to improve object discovery behind opaque surfaces (Figure 4.3).

Additionally, Wang et al. [2016] made a research project to assess the usability of transparency in 3D cadastre visualization. They used three different degrees of

transparency to represent three categories of boundaries: physical, administrative and linked boundary (both physical and administrative). The result of the research showed that three levels of transparency were enough to demarcate legal parcels and their physical counterparts.

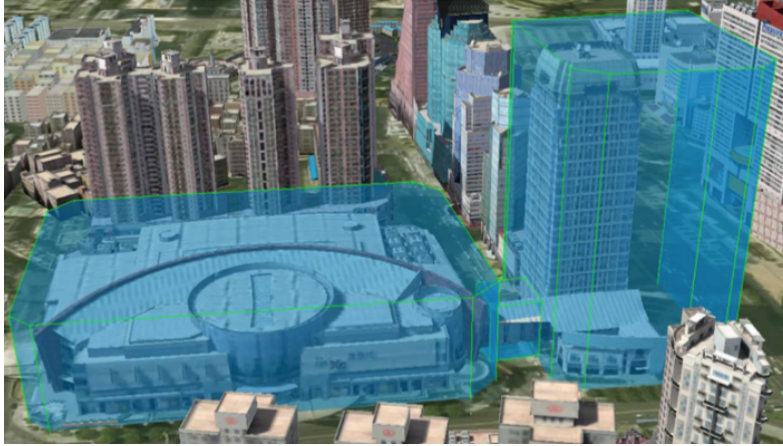


Figure 4.2: Use of transparency to enhance the difference between physical and legal objects [Pouliot et al., 2018]

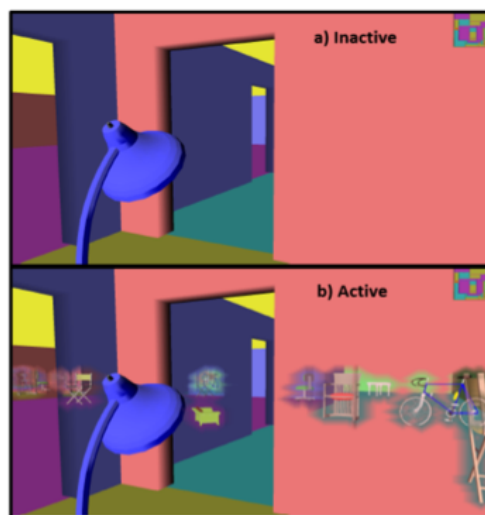


Figure 4.3: Use of dynamic transparency to improve object discovery [Pouliot et al., 2018]

4.1.5 Object selection

The selection of objects is a very important interaction technique for the success of any 3D visualization tool [Pouliot et al., 2016]. Clicking on an object with the mouse, or touching the object (if we are using a touch device), is referred to as object selection. Selection can also be performed on administrative data, for example, it is possible to show or select only the parcels owned by a specific person.

A direct consequence of the selection is the highlighting of the feature, usually by changing its colour to make it better visible [Shojaei, 2014].

An example of selection can be found in the 3D cadastre prototype for the Russian Federation described by Vandyshva et al. [2012]. The object selection was used to highlight the available 3D objects starting from a 2D representation. In this way, the user has a clear overview of the available 3D parcels and he/she only has to click on the 2D parcel to visualize it in 3-dimensions [Vandyshva et al., 2012].

4.1.6 Object search

A step beyond *object selection* there is a more advanced functionality, the so-called object search. This search allows the user to find required information in a quick and easy way. It can be carried out both on spatial and non-spatial data, and it can be based on address, name, geocode, coordinates, etc. The user only needs to type the input, as we can see in [Figure 4.4](#), and the system will zoom to the desired object or will highlight the object [[Shojaei, 2014](#)].

A variant of object search is the *active set search* [[van Oosterom et al., 2002](#)]. The concept of active set refers to a group of objects which has been selected and saved for further processing [[van Oosterom et al., 2000](#)]. Thanks to this functionality, a group of parcels with the same characteristics can be easily found.

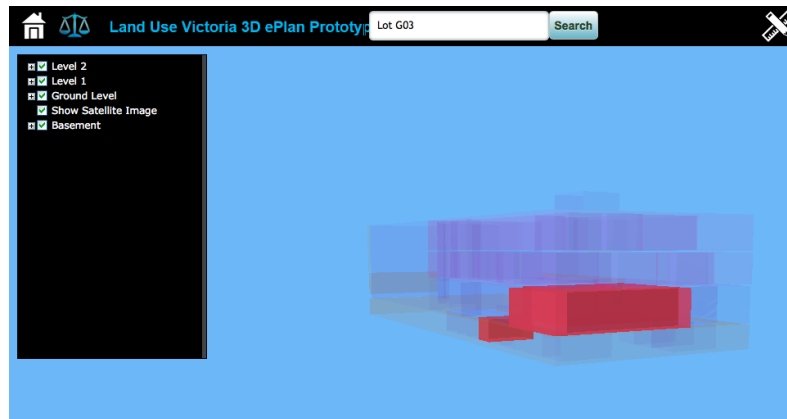


Figure 4.4: Example of object search as developed in the ePlan prototype for the state Victoria, Australia. The user can type the lot name, in this case "Lot G03" and the system will highlight and zoom to the indicated lot (i.e. parcel) [[Olfat et al., 2016](#)].

4.1.7 Wireframe display

A wireframe view is a representation of a 3D object exclusively by means of its edges, which can be used to enhance the rendering speed when a large amount of data is loaded.

In cadastral applications, the wireframe view can also help the user to see through objects and overcome occlusion issues that obstacle the correct perception of parcel boundaries. [Figure 4.5](#) shows an example of wireframe display.

In addition, wireframe display can be applied to non-selected parcels (or vice versa) to provide context and distinguish them from the selected ones [[Shojaei, 2014](#)].

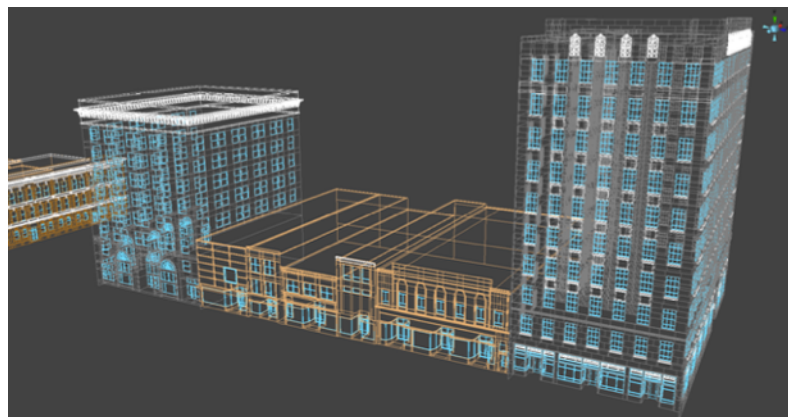


Figure 4.5: Example of wireframe display [[Shojaei, 2014](#)].

4.1.8 Explode view

Explode view is used in 3D cadastre applications to show the internal components of complex units (for example, apartment buildings). This functionality displaces the components in space separated by a given distance, therefore all the components of an entity are slightly shifted from their original location [Shojaei, 2014]. Among others, Ying et al. [2016] explored the explode view functionality; they started from a connected set (left of Figure 4.6) and using displacement equations they created the exploded view (right of Figure 4.6) to have a better visualization of the inner units [Ying et al., 2016].

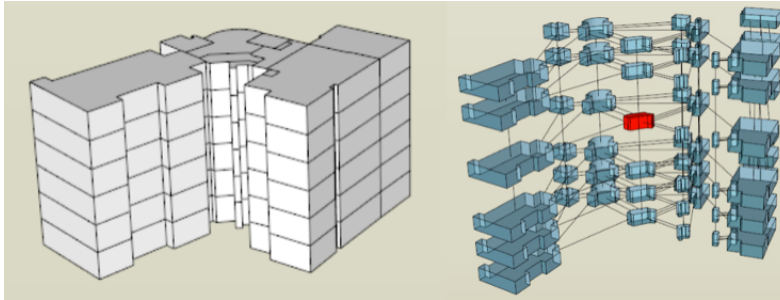


Figure 4.6: Explode view example [Ying et al., 2016]

4.1.9 Sliding

The sliding tool is similar to explode view since it also has the function to better show the internal geometry of a 3D object, which is otherwise occluded by the surrounding features. Usually this tool moves an object horizontally in a specified direction, i.e. the floor of an apartment building, to reveal its internal structure. By looking at the interior geometry of the parcels the user can better understand the subdivision of the space [Shojaei, 2014]. This functionality can also be used to slide the floors of a building on top of each other to make comparisons (Figure 4.7). The sliding tool is one of the essential functionalities of the 3D cadastre prototype of the Russian Federation. By selecting the *move-floor-mode*, floors can be dragged out of high rise buildings, revealing the parcels subdivision in each floor [Vandysheva et al., 2012] as shown in Figure 4.8.

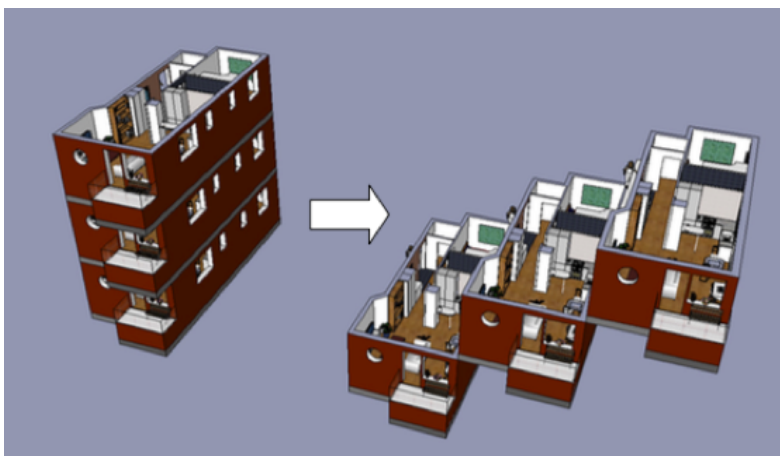


Figure 4.7: Sliding tool used to compare floor plans [Shojaei, 2014]

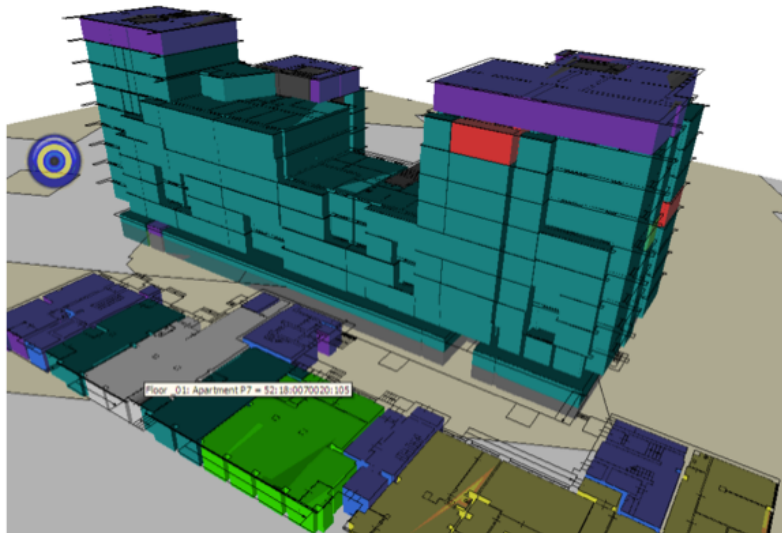


Figure 4.8: Sliding tool in the 3D cadastral prototype of the Russian Federation [Vandysheva et al., 2012]

4.1.10 Cross-section view

The slicing or cross section view is a useful functionality in 3D cadastral visualization because it allows the user to see the internal geometry of the building or parcel [Shojaei, 2014]. A cross section is a view revealed by slicing an object by means of a plane that it is usually vertical, but can also be oriented in a different way [Shojaei et al., 2013]. In addition, allowing the user to slide the slice plane, would enable introspective views of the spaces.

The cross-section tool is currently used in Australia in building subdivision plans to represent the relative position of cadastral objects. In particular, this visualization tool is useful in complex buildings to facilitate the understanding of the ownership information [Shojaei et al., 2013]. In Figure 4.9 we can see an example of a cross-section view.

Other researchers combined 3D displacement (i.e. *explode view* described above) and cross-section to have an internal view of the building/parcel [Pouliot et al., 2018] as shown in Figure 4.10.

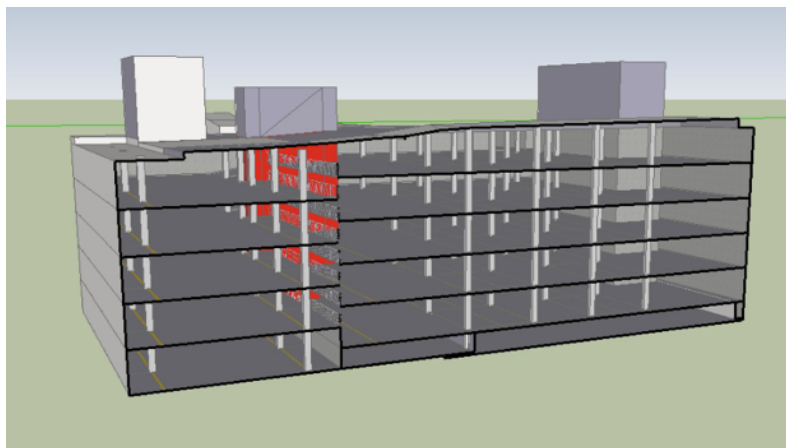


Figure 4.9: Cross section view obtained by means of a vertical plane slicing the building in a vertical way. In this way, the interior of the apartment building can be better visualized [Shojaei et al., 2013].

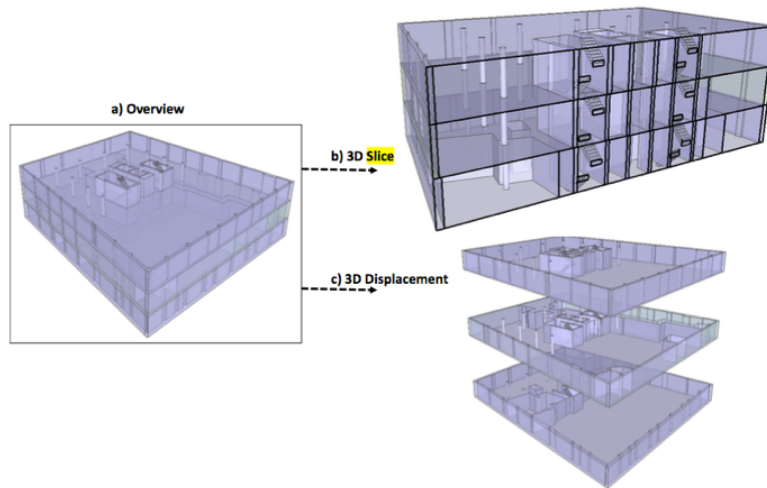


Figure 4.10: Combination of *explode view* (3D displacement) and *cross-section* functionalities [Pouliot et al., 2018]

4.1.11 Visualization cues

Switching on and off visualization cues can help the user to estimate the relative position of an object. Figure 4.11 shows a pipeline (coloured in purple) which is partly above and partly below ground; to better represent this, vertical red “sticks” have been added to the representation to show the position and distance between the pipeline and the earth surface [Pouliot et al., 2018].

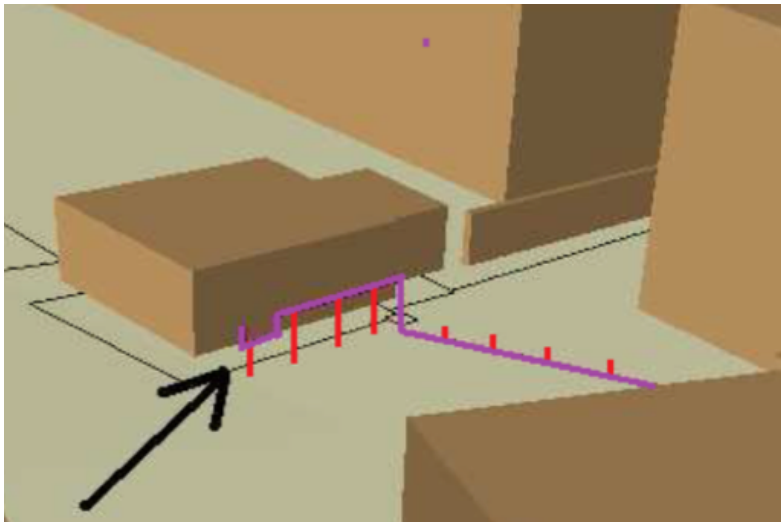


Figure 4.11: Adding visualization cues to a pipeline representation can help understanding depth and height of the object with respect to the earth surface. The red “sticks” represent the distance between the ground surface and the pipeline [Pouliot et al., 2018]

4.1.12 3D measurement tools

In cadastral applications, a 3D measurement tool can be used to calculate the area or volume of a parcel to quickly estimate its size. The user must be able to perform measurements according to specific needs [Shojaei, 2014].

Among the possible calculations there are: areas, volumes, distance between points

(as shown in [Figure 4.12](#)), etc. Furthermore, to perform these operations, object snapping should be implemented to allow the user to make accurate measurements [[Shojaei et al., 2013](#)].

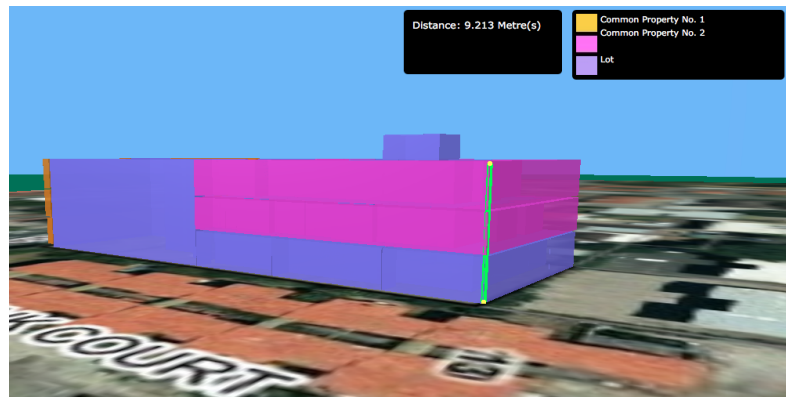


Figure 4.12: A spatial measurement tool calculating the distance between points as implemented in the ePlan prototype by the Land Use Victorian Government [[Olfat et al., 2016](#)]

4.1.13 3D buffer

The 3D buffer functionality creates a (buffer) area around an object. This feature can be suitable for querying and selecting neighboring objects to a 3D object; for instance, the assessment of constraints in a specific area and the creation of an "active set" of objects. The 3D buffer functionality must be possible both for cadastral parcels and for physical objects in order to support various analyses. The features within the buffer can be then selected for further processing [[Shojaei, 2014](#)]. [Figure 4.13](#) shows a 3D buffer around of a pipeline.

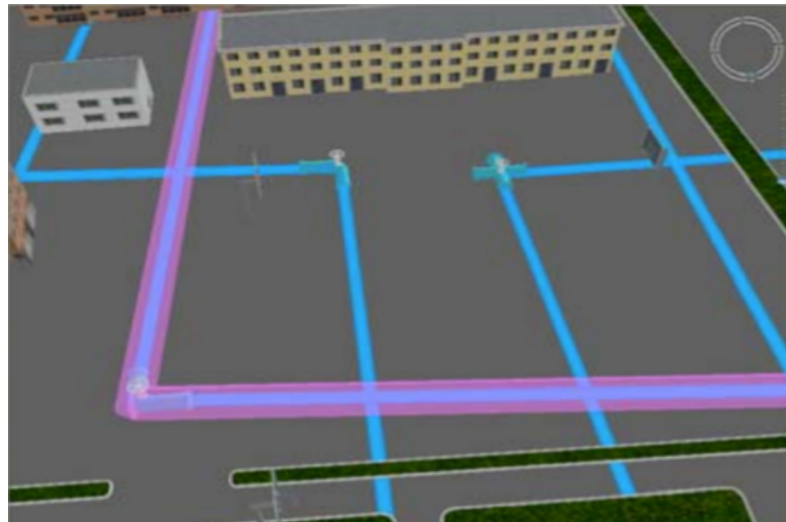


Figure 4.13: Example of a 3D buffer around a pipeline object [[Shojaei, 2014](#)]

4.1.14 Display partly unbounded objects and "complex geometries"

Legal objects can be both bounded and unbounded. Bounded objects have a defined volume, so they might seem relatively easy to represent. Nevertheless, they could become rather complex; examples are, non-2-manifold geometries and curved

boundary surfaces. The support for these “complex geometries” is crucial to represent the different shapes of buildings in modern cities. Curved surfaces boundaries are difficult to represent; therefore, a solution could be the approximation of those surfaces.

In addition, unbounded objects are challenging for 3D cadastre visualization since their volume is undefined. As an example, these unbounded objects are parcels that continue infinitely towards the sky or below ground level. A 3D cadastre application should be able to show them in a non-misleading way. Some researchers suggested their representation in two dimensions as the most efficient method [Shojaei et al., 2013].

4.2 REQUIREMENTS RELATED TO THE WEB-BASED VIEWER AS A VISUALIZATION PLATFORM

In addition to the functionalities related to 3D visualization, the web viewer requirements must be considered too. One of the main aims of cadastre is to make information available to everybody and the web is a powerful tool which can make this happen. Consequently, the choice of the most appropriate web platform is crucial. Having a good 3D visualization would be useless if the platform in which the data is visualized is not suitable for that purpose. Choosing the proper web based platform for the dissemination of 3D cadastre is not an easy task. Most of the following functionalities should be present.

4.2.1 Platform and browser independence

A wide variety of platforms is present nowadays, they are either based on computer workstations or on mobile/tablet devices. Among the most popular ones we can list Windows, Mac OS, Linux, Android and iOS. Some applications only work in specific operating systems, but this strongly limits their dissemination to users of a different platform [Shojaei, 2014]. For example, if an application is supported only by Windows, Mac users will have difficulties in using the same application or will not be able to use it at all.

Cadastral systems serve a wide variety of users and a platform and browser independent 3D visualization system is a prerequisite for achieving a high level of dissemination. In this way, the user does not have to concern about the underlying technology or incompatibility problems for using the application [Shojaei et al., 2013]. The cutting-edge Web Graphics Library (WebGL) technology to visualize 3D data on the web is supported by all major platforms and browsers, which makes it a good candidate for implementing a 3D cadastre web-based solution.

4.2.2 Handling massive data and caching/tiling between server and client

A 3D cadastral application on the web must be able to handle many users at the same time, while maintaining the performance objectives. The amount of data obtained by combining physical and legal objects is huge, therefore the application performance should not be affected while dealing with big data [Shojaei, 2014].

In addition, since the client does not need to visualize all the data at the same time, an application supporting tiling and caching can provide a subset of the server data when needed, in order to make the visualization faster and smoother.

4.2.3 Layers control

As already mentioned in [Section 4.1](#), an important requirement for a web-based 3D cadastral viewer is the support for 3D visualization basics. In particular, the possibility to add and remove layers (i.e. layers containing reference objects), switch on and off their visibility and modify their transparency enables the user to customize the visualization.

Moreover, the capability of adding layers to the web viewer implies the support of a variety of data sources [[Shojaei, 2014](#)].

4.2.4 Database support

A database allows users to have all the information in a unique place and to maintain it up to date. It is good practice to store 2D cadastral data into databases, the so called Digital Cadastral DataBases (DCDBs). In the same way, 3D cadastral data should also be stored in DCDBs. For this reason, 3D cadastre visualization applications must support and connect to databases to retrieve data and submit to the server the changes made in the front-end [[Shojaei, 2014](#)].

Nowadays, a number of DataBase Management Systems (DBMSs) are available to store and manage the data, but only a few of them, such as PostgreSQL/PostGIS and Oracle, are suitable for storing 3D data. In addition, the database should also contain persons/parties data and the legal/administrative data, such as registration of Rights, Restrictions and Responsibilities (RRRs), which are crucial for a 3D cadastre application.

4.2.5 Support different models (vector/polyhedral, raster/voxel, point clouds)

As already discussed above, the combination of physical and legal objects further complicates the visualization issues.

Topographic objects used as reference are buildings, pipelines, roads, bridges, railways, waterways, etc. They can be represented in different ways such as, raster or voxel representations, vector or polyhedral surfaces and even smart point clouds [[Pouliot et al., 2018](#)]. Hence, a viewer supporting a wide variety of models allows diverse datasets to be used in the same environment.

4.2.6 Support of 3D topographic visualization

A 3D web viewer should support 3D topographic visualization both for the content or information model, and for the encoding, which is the actual exchange format. Many data formats and specifications currently exist to store, deliver and exchange 3D data. The most popular ones are: CityGML, KML, X3D, GeoVRML, GML, DWG, 3DS, shapefile, Collada, 3D PDF, IFC and LandXML. For a more detailed explanation about the formats, see Davood Shojaei PhD thesis [[Shojaei, 2014](#)].

4.2.7 Support for geo-referencing

A 3D cadastral system contains geo-data. Geo-data is information about a geographic location and it is produced in different coordinate systems. For this reason, a good 3D cadastre visualization platform should support various horizontal and vertical coordinate systems [[Stoter et al., 2012a](#)].

In addition, real world coordinates (with an origin on the surface of the earth) need to be translated into screen coordinates (with origin on the top left part of the screen) to be visualized on a computer screen [[Pouliot et al., 2018](#)].

4.2.8 Ensure spatial validity (3D vector topology)

Integrating topology in 3D cadastral applications can be of great benefit for two main reasons: (1) efficiently render 3D data and (2) carry out various types of analyses. An application lacking topological support may have information redundancy and therefore it will require more resources to render the data.

Topological structures enable spatial analyses such as, identify adjacency, intersection, connectivity, containment and disjunction between objects, which can be useful for users of 3D cadastre. Checking spatial validity makes sure that all the parcels are closed and valid both in 2D and 3D. For example, complex 3D cadastral objects, such as apartment buildings, are composed by parcels sharing many faces which should not overlap or intersect, but align with each other to ensure a valid subdivision of the space. On the other hand, in case of unbounded volumes, using vector topology can give invalid results [Shojaei, 2014].

Spatial validity is closely related to the choice of the cadastral database (see [Section 4.2.4](#)).

4.2.9 Underground View

Modern cities are growing and becoming more complex, both above and below the earth surface. Many underground developments such as pipelines, shops, parking spaces, tunnels, etc. are now present in urban areas. Lacking underground visualization will lead to a non-reality-like view of the world. The possibility of displaying the physical and legal underground objects in 3D cadastre applications is, therefore, of primary importance.

In order to have an underground view, the camera should be able to move under the terrain. Although this requirement can seem quite straightforward, many 3D web viewers and virtual globes lack this functionality, an example is Google Earth [Shojaei, 2014]. [Figure 4.14](#) shows an example of underground view of a 3D model.

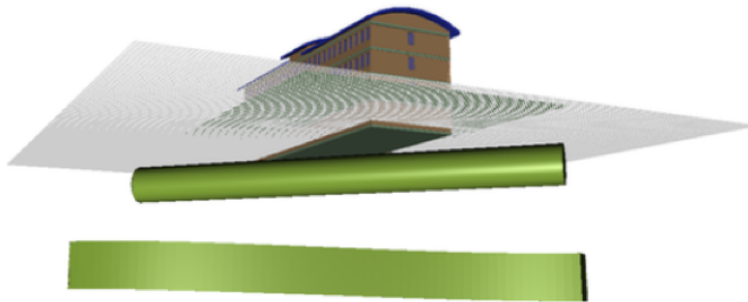


Figure 4.14: Example of underground view [Shojaei, 2014]

4.2.10 Open source platform

An open source platform is a platform that provides the source code of the application to the users. The advantage of an open source solution is that the community is encouraged to participate and possibly contribute, if the platform can be extended (see [Section 4.2.11](#)), to the development of the applications [Shojaei, 2014].

4.2.11 Possibility for the platform to be extended

Since cadastre visualization has specific needs, which are rather different from the ones of traditional 3D visualization, using an extensible platform will give the possibility for the software developer to build a custom-made application that perfectly

fits the purpose of 3D cadastre [Shojaei, 2014]. In other words, the user should have the possibility to write his own code and modify, add or remove the functionalities included in the platform.

4.2.12 2D overview map (orientation)

Including a 2D overview map in the 3D cadastre web viewer would give a vertical view over the scene. This will help the user to understand the location of the camera with respect to the 3D world.

The 2D overview map is very much used in games and it is generally a small rectangle on the corner of the canvas to provide orientation during navigation [Shojaei, 2014].

4.3 IMPLEMENTATION PRIORITIES

Developing all the requirements is rather ambitious within the time frame set for this MSc research, therefore the definition of the key requirements and of the priorities among them is crucial for the successful implementation of the 3D cadastre prototype. Table 4.1 and Table 4.2 distinguish three categories of requirements according to their feasibility in the given time frame and to the available technology:

YES

This category includes the key requirements, or in other words, the requirements that will be for sure implemented during the development of the thesis research. The technology is in place and their implementation is possible in the given time frame. Also, to this group of requirements is given a higher priority because it will constitute the base of the 3D cadastre prototype.

MAYBE

This group includes all the requirements that are technologically feasible, but probably not achievable in the given time span. These requirements has been given a lower priority, therefore they will probably be implemented in the near future, since there is evidence of their feasibility and they would complement the ones implemented in the first group.

NO

This category incorporates all the requirements that, at the moment, are not technologically feasible and cannot be implemented in the near future. Therefore, this group of requirements has the lowest priority and they will surely not be implemented, but they need to be taken into account for the future development of the prototype.

<i>Requirements for 3D visualization</i>	
YES	Navigation tools and view controls Tooltip Integrating topography and reference objects Transparency Wireframe display Object selection Object search
MAYBE	Cross-section view 3D measurement tools Display partly unbounded objects and 'complex' geometries
NO	Explode view Visualization cues Sliding 3D buffer

Table 4.1: Requirements priorities for 3D visualization

<i>Requirements for the web-viewer</i>	
YES	Platform and browser independence Layers control Support of 3D topographic visualization Support for geo-referencing Underground view Open source platform Possibility for the platform to be extended
MAYBE	Database support Support different models (vector/polyhedral, raster/voxel, point clouds) 2D overview map (orientation)
NO	Handling massive data and caching/tiling between server and client Ensure spatial validity (3D vector topology)

Table 4.2: Requirements priorities for the web-viewer

5

THE WEB BASED VIEWER

The previous chapter addressed the requirements description, now the focus is on the contemporary web-based viewers for visualizing 3D cadastral data. [Section 5.1](#) analyses and describes in detail the characteristic of six different web platforms. [Section 5.2](#) makes a comparison of the platforms previously described to highlight their pros and cons. Finally, [Section 5.3](#) motivates the choice of the most suitable platform to develop the prototype.

5.1 ANALYSIS OF THE EXISTING WEB VIEWERS

Choosing the appropriate visualization application for 3D cadastre is a big challenge because of the wide variety available, due to the ever-growing new technologies. For this reason, a good knowledge of existing platforms and their capabilities can help building a successful cadastral prototype [[Shojaei, 2014](#)].

The web viewers analyzed are: iTowns, Cesium JS, OSM Buildings, WebGL Earth, GeoBrowser 3D and ESRI CityEngine Web Viewer. They are all based on the cutting-edge WebGL technology. Another viewer which has been considered is Google Earth, although it has not been analyzed in detail because only partially based on WebGL, closed source and not browser independent (for more information about Google Earth, see [Section 6.6.1](#))

As described by the [Khronos Group \[2017\]](#) who developed it, WebGL is a cross-platform open source web standard for a 3D graphics API based on OpenGL ES. The API context is obtained from the HTML5 <canvas> element, that means that no plug-in is needed in the web browser to use the application [[Pereira, 2013](#)]. Since WebGL is a low-level JavaScript API, designing an elementary 3D geometry needs a lot of work. Consequently, various open source JavaScript libraries were developed to make the programming process easier. One of the most popular ones is *three.js*, which provides higher level access to the API to simplify programming [[Shojaei, 2014](#)].

A big advantage of WebGL is that it brings 3D into the web without the installation of a plug-in and it is implemented directly into the web browser. Moreover, most browser vendors like Apple, Google and Mozilla are members of the WebGL Working Group [[Khronos Group, 2017](#)]. Hence, WebGL is supported by all major browsers and it works on many mobile platforms [[Mackey, 2017](#)]. The payoff in using WebGL is that it is considerably faster than the 2D canvas context, so it performs rather well in case of complex visualizations. It ensures a good performance and a wide variety of visualization options/functions. Also, it can generate a degree of realism and configurability that is hardly possible with other solutions [[Danchilla, 2012](#)].

Although WebGL represents the state of the art in 3D graphics for the web, some limitations must be pointed out:

- Visualizing massive datasets in WebGL can be an issue. Often browsers and mobile devices have a limited amount of cache memory that cannot be exceeded. Hence a smooth visualization is not always possible [[Shojaei, 2014](#)].
- WebGL has been designed for today's graphic cards, therefore old generation computers may not support it [[Shojaei, 2014](#)].

- Setting up WebGL without a framework is not convenient due to the amount of work needed to build up the application from scratch. For this reason, a good option is use an existing WebGL based viewer in which many of the functionalities needed are already implemented and some others can be added [Danchilla, 2012].

5.1.1 iTowns

iTowns is an open source JavaScript/WebGL framework for 3D geospatial data visualization based on *three.js* [iTowns, 2017]. It was initially developed by the French Institut National de l'Information Géographique et Forestière (IGN) to integrate street images with terrestrial Lidar point cloud data acquired by the *Stereopolis* mapping vehicle [IGN, 2015].

iTowns is a powerful framework for fast and easy prototyping via a modern web interface. It was primarily designed to handle various types of geo-data and to make precise 3D measurements [iTowns, 2017]. Recently the platform developed from version 1 to version 2. The new version contains a full package of 3D visualization tools making it a good candidate for the visualization of 3D cadastre, the only concern is about speed and user experience. Figure 5.1 shows an example of an application based on iTowns.



Figure 5.1: WFS extruded on globe: an application developed with iTowns [iTowns, 2017].

5.1.2 Cesium JS

Cesium JS is an open-source JavaScript library which uses WebGL to create 3D geospatial applications. It is led by the Cesium Consortium and thanks to the *Apache 2.0 license* it is free for commercial and non-commercial uses [Cesium, 2018].

Cesium JS supports open formats wherever available, and develops new open formats when they are not; in this way, it sets new standards for open 3D geo-spatial formats.

The big advantage of Cesium JS is the existence of a big community supporting developers in their work; developers can post questions on the Cesium forum about specific problems and get answers from other members of the community. By doing so, the expertise can be easily shared [Cesium, 2018].

Furthermore, Cesium JS allows to switch among 3D virtual globe, 2D map and 2.5D Columbus view within a single Application Programming Interface (API). Time dynamic simulations are also supported, together with realistic environment features such as sunlight, atmosphere, fog, water and moon [Inc., 2017]. Cesium also includes an interesting live code editor and example gallery called *Sandcas-*

tile, which allows developers to edit and run the code directly from the web browser [Chaturvedi, 2014]. Figure 5.2 shows a demo of Cesium JS.



Figure 5.2: An application built on top of Cesium JS showing the buildings in New York city coloured according to their height value [Cesium, 2018].

5.1.3 OSM Buildings

OSM Buildings is a JavaScript library for visualizing the geometry of OpenStreetMap buildings in 2D and 3D. It is free and open source, so the source code can be downloaded and modified. Two versions are available: the classic 2.5D and the modern 3D version [OSM buildings, 2017]. The classic version is great for devices compatibility, good performance on old hardware, and shadow simulation. The new version is based on WebGL and it is now available on the Github repository [Github, 2017]. This last version is performing well on modern graphic hardware, it can handle a big amount of objects and combine different data sources. An example of OSM Buildings is shown in Figure 5.3.

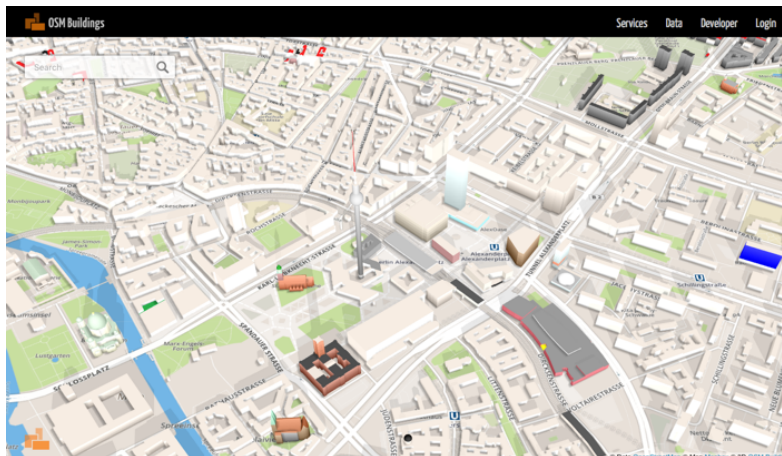


Figure 5.3: Example of OSM Buildings [OSM buildings, 2017].

5.1.4 WebGL Earth

WebGL Earth is an open source software for the visualization of maps, satellite images and aerial photographs on top of a virtual globe. It is built on top of WebGL

and therefore allows to build applications without the need of installing a plug-in in the web-browser [Chaturvedi, 2014]. WebGL Earth uses “under the hood” the Cesium JS project for the rendering of the data, as can be noticed from the similar interface [WebGL Earth, 2017]. Figure 5.4 shows a WebGL Earth application.



Figure 5.4: An example of WebGL Earth application [WebGL Earth, 2017].

5.1.5 GeoBrowser 3D

GeoBrowser 3D is developed by GraphiTech. It is powered by WebGL graphics and offers a 3D Open Geospatial Consortium, Inc. (OGC) compliant solution. The application supports maps, vectors graphics surface models and three-dimensional buildings that the user can directly add into the application. It is also possible to perform precise measurements of distances and areas directly on the viewer [Geobrowser 3D, 2017]. Similarly to WebGL Earth, also GeoBrowser is powered by Cesium JS. An example of the GeoBrowser 3D is shown in Figure 5.5.

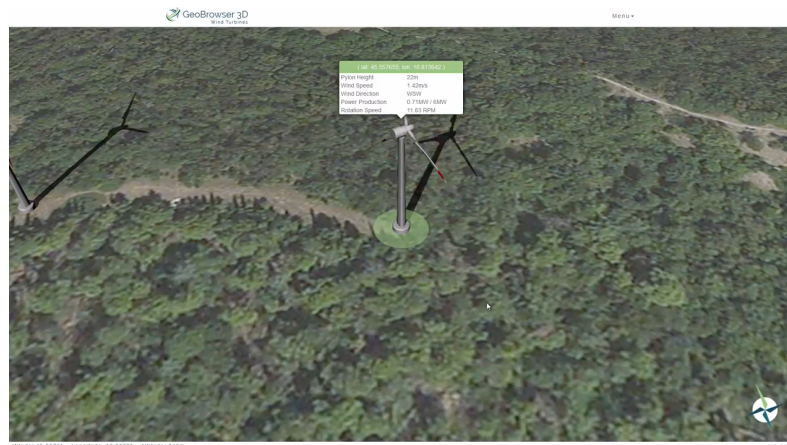


Figure 5.5: An example application built on top of GeoBrowser 3D showing a wind turbine and the information related to it [Geobrowser 3D, 2017].

5.1.6 ESRI CityEngine Web Viewer

CityEngine Web Viewer was released by ESRI and it is based on HTML5 WebGL technology. Users can create their own 3D scenes using the software package *ESRI CityEngine* or the *Export To 3D Web Scene* geoprocessing tool [ESRI ArcGis, 2017]. The main difference between this web viewer and the others described above is that

CityEngine is commercial and non-open-source, hence it is not possible to modify the source code, but it integrates a Python scripting interface which enables the user to carry out queries and other tasks [Ribeiro et al., 2014].

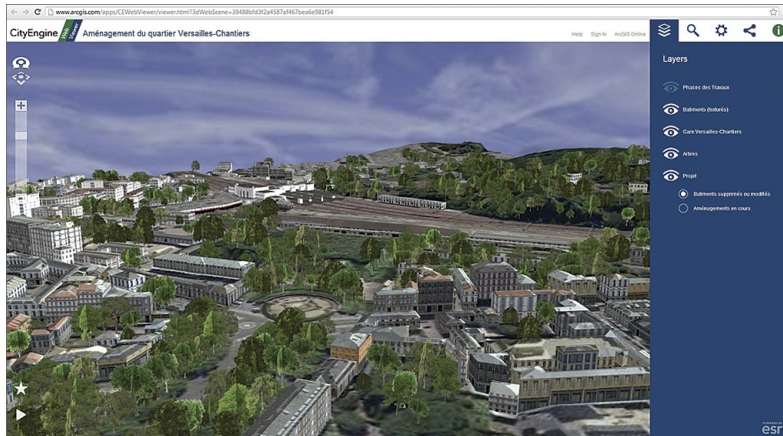


Figure 5.6: An example application using ESRI CityEngine Web Viewer [ESRI ArcGis, 2017].

5.2 COMPARISON OF THE 3D WEB VIEWERS

Six different WebGL based viewers have been identified; mature and in early development stage, open source and commercial. Now, they will be analyzed and compared against the requirements defined in the first part of the paper. The goal is to look for platforms which already have a considerable number of requirements implemented to facilitate and fasten the development of the thesis project. A demo with a high number of functionalities has more chances of being chosen to implement the final prototype, since there is the concrete possibility to develop them again and find related support to do so. It follows that, the extensibility of the platform is an essential requirement to develop the remaining functionalities.

The analysis phase consists of checking the presence of the requisites in demos or prototypes built on top of that specific application. Nevertheless, some requirements are hard to check only based on evidence, in this case the related literature has also been considered.

Table 5.1 and 5.2 summarize the requirements check according to the different platforms. The green checkmark (✓) means that the requirement has been verified to exist, the orange question mark (?) means that no evidence has been found or no one still implemented it, and the red cross mark (✗) means that it is certainly not possible to implement that requirement.

Most of the demos have different purposes than 3D cadastre and many of the functionalities are just not needed for those use cases. In Table 5.1, all the basic functionalities for 3D visualization, such as navigation tools, transparency, topography support, and so on, are present in all web viewers. On the other hand, there is no evidence of the range of capabilities requested by 3D cadastre, like explode view and cross section view. A possible explanation could be the different purpose of the demos analyzed. Most of them are in the geospatial domain, and specifically meteorology, history, defense and smart cities seemed to be quite popular fields. Since none of these domains is directly related to 3D cadastre, there is no need for these applications to develop such functionalities. Nevertheless, thanks to the extensibility of the platforms (see Table 5.2), most of these requirements can probably be implemented.

Requirements related to 3D visualization in the available WebGL platforms						
Requirements	iTowns	Cesium JS	OSM Buildings	WebGL Earth	GeoBrowser 3D	ESRI CityEngine
Navigation tools and view controls	✓	✓	✓	✓	✓	✓
Tooltip	✓	✓	✓	✓	✓	✗
Integrating topography and reference objects	✓	✓	✓	✓	✓	✓
Transparency	✓	✓	✓	✓	✓	✓
Object selection	?	✓	✓	✓	✓	✓
Object search	?	✓	✓	✓	✓	✓
Wireframe display	✓	✓	?	?	?	✓
Explode view	?	?	?	?	?	?
Sliding	?	?	?	?	?	?
Cross-section view	?	?	?	?	?	✓
Visualization cues	?	?	?	?	?	?
3D measurement tools	✓	✓	?	?	✓	✓ only through programming
3D buffer	?	✓	?	?	?	?
Display partly unbounded objects and 'complex' geometries	?	?	?	?	?	?

Table 5.1: Requirements check in the different WebGL platforms (3D visualization)

Requirements related to the web viewer in the available WebGL platforms						
Requirements	iTowns	Cesium JS	OSM Buildings	WebGL Earth	GeoBrowser 3D	ESRI CityEngine
Platform and browser independence	✓	✓	✓	✓	✓	✓
Handling massive data and caching/tiling between server and client	✓	✓	✓	✓	✓	? but foresees the possibility of handling massive cadastral data
Layers control	✓	✓	✓	✓	✓	✓
Database support	?	?	?	?	?	✓
Support different models (vector, raster, point clouds)	✓	✓	✗	✗	✓	✓
Support of 3D topographic visualization	✓	✓	✗	✓	✓	✓
Support for georeferencing	✓	✓	✓	✓	✓	✓
Ensure spatial validity (3D vector topology)	?	?	?	?	?	?
Underground View	?	?	?	?	?	✓
Open source platform	✓	✓	✓	✓	✓	✗
Possibility for the platform to be extended	✓	✓	✓	✓	✓	✓ Python scripting interface
2D overview map (orientation)	✓	✓	?	?	?	?

Table 5.2: Requirements check in the different WebGL platforms (Web viewer)

5.3 SELECTION OF THE PLATFORM TO DEVELOP THE PROTOTYPE

Considering all the tested web viewers, along with the specified requirements, the most likely applicable and capable of supporting 3D cadastre visualization seem to be iTowns, Cesium JS and ESRI CityEngine.

The first two have many of the requirements already implemented and a vast community of users, and on top of that, dedicated forums and blogs are actively helping developers to implement their own applications. Cesium JS has a detailed [API](#) documentation, code examples and tutorials to practice and get to know the library [[Cesium, 2018](#)]. iTowns provides great support for users and a wide variety of examples [[iTown, 2017](#)].

ESRI CityEngine is the only commercial and non-open-source solution analyzed. The number of requirements met is considerably high and could be sufficient to implement some basic functionalities. In addition, the integrated Python scripting interface enables the user to carry out attribute queries, control repetitive tasks and automate specific actions [[ESRI ArcGis, 2017](#)].

On the other hand, the three remaining options - namely OSM Buildings, WebGL Earth and GeoBrowser 3D - seem less promising.

As Cesium JS and iTowns, also OSM Buildings provides a good [API](#) documentation [[OSM buildings, 2017](#)] but it owns less functionalities, therefore this candidate is less likely to be used in the actual implementation of the 3D cadastre viewer.

WebGL Earth and GeoBrowser 3D are still in their early development stages, their community is relatively small and therefore they have limited support compared to more mature web viewers [[WebGL Earth, 2017](#); [Geobrowser 3D, 2017](#)]. As already mentioned, they are powered by Cesium JS which has a wide community of users and good support for developers. Although, compared to Cesium JS, their users support and documentation are still rather limited.

On the basis of this analysis, Cesium JS has been chosen to develop the 3D cadastre prototype. The research will focus on the capabilities of Cesium JS in order to properly visualize and disseminate 3D cadastral parcels. The existence of a live code editor which allows the developers to edit and run the code directly in the web browser is good to facilitate the development phase. Moreover, Cesium JS has the largest community of users among the platforms analyzed, making it the best candidate for the development of the 3D cadastre prototype.

Nevertheless, given the uncertainty of developing most of the requirements, there is still the possibility to switch to another platform later on in the research in case Cesium JS will result not suitable for the purpose.

6

DEVELOPMENT OF THE 3D CADASTRE PROTOTYPE

This chapter describes the implementation details related to the 3D cadastre prototype, following the theoretical guidelines introduced in [Section 2.4](#). The main decisions and consequent results are explained, focusing on the issues and shortcomings that arose during the process.

The chapter is structured as follows. First of all, [Section 6.1](#) describes the overall server-client system architecture. After that, [Section 6.2](#) gives an overview of the datasets used in the prototype, [Section 6.3](#) summarizes the storage schema developed for the Queensland cadastre, [Section 6.4](#) explains the process of data encoding, [Section 6.5](#) illustrates the process of selection of a suitable DTM. Then, [Section 6.6](#) to [6.12](#) describe the implementation of different requirements together with related issues. Finally, [Section 6.13](#) provides a summary of the requirements included in the 3D cadastre prototype.

6.1 SERVER-CLIENT SYSTEM ARCHITECTURE

Adopting a web-based solution is optimal for dissemination of the data as web browsers offer a relatively hardware/software independent platform, reaching many possible users without great efforts at the user side.

This section contains an overall description of how all the parts of the system work together in synergy.

A custom-made Java encoding software extracts the [KML](#) files from the PostGIS [DB](#) and stores them as files on the local machine. At present, the geometric 2D and 3D information is loaded on the client as [KML](#) files to expose the data to the outside world. In the future, the creation of the [KMLs](#) could also be carried out on the fly (i.e. in real time) by accessing the PostGIS database from the Cesium JS client.

At the server side, a Web Feature Service ([WFS](#)) is established on top of the Land Administration Domain Model ([LADM](#)) views in the database in order to retrieve the persons/parties and the [RRRs](#) related to the 2D and 3D parcels.

Further at the client side, Apache Tomcat is used as a web server, while Geoserver is used as map/data server. Then, a 3D geo-information aware [WebGL](#) based solution is applied to visualize the 3D parcels (and reference objects) and interact with the information.

Since the amount of data loaded on the client is not huge, there was no need to put in place tiling/caching techniques, although in the future development the amount of data could increase considerably, to host the whole dataset of the Queensland [DCDB](#). In that case, the client should load only a portion of the information, as all data in the [DCDB](#) would be far too much. Server-client caching techniques could be used to manage the communication between server and client when panning over the area.

One of the uncertainties of this project is about the definition of the server-client interaction, and in particular whether it is better to use a [DB](#) or encodings, such as [KML](#), JavaScript Object Notation ([JSON](#)), GL Transmission Format ([gITF](#)), etc. For the back-end storage, the database option seems appealing for two reasons. The advantage of having the data in one place and always up to date, and the fact that the 2D parcels of the Brisbane [DCDB](#) are already stored in a [DBMS](#). On the other hand, for the data transfer to the client the use of encodings such as [KML](#), [JSON](#), [gITF](#), etc.,

can be convenient, since the 3D parcels encoded until now are in KML format. An overall diagram showing the server-client system architecture is shown in Figure 6.1

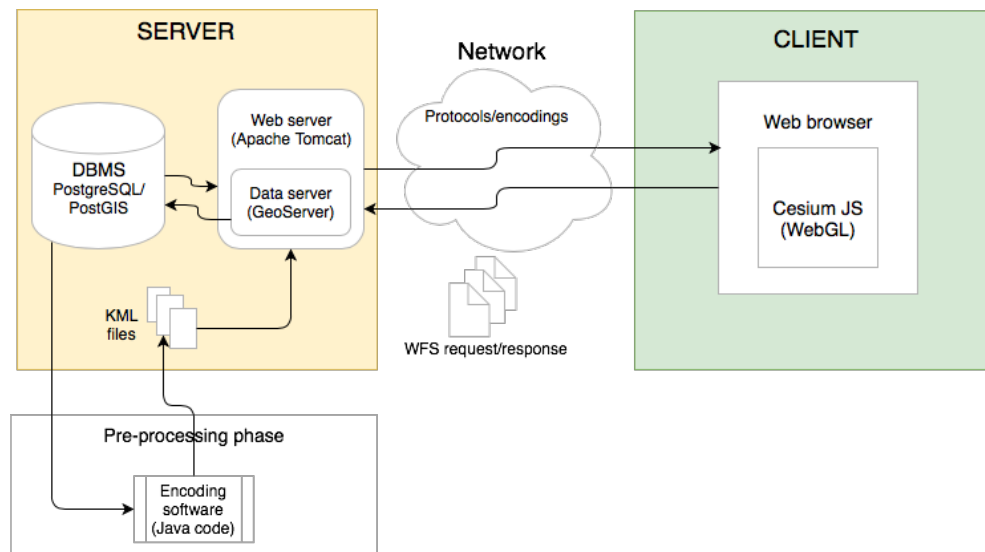


Figure 6.1: Server-client system architecture diagram of the 3D cadastre prototype.s

6.2 THE DATASETS

Setting up a prototype requires the availability of sample data to make it realistic. Although the scope is to build a 3D cadastre prototype, in order to better investigate the visualization and dissemination aspects of 3D cadastre, it was decided that the combination of 2D and 3D cadastral data was the most realistic scenario. In this way, both the 2D and 3D DCDB datasets can be maintained in the same place.

The cadastral data used for the project was taken from the Queensland DCDB, a collection of all the ownership boundaries and related property description of land parcels in Queensland (Australia). The choice stems from its long tradition, dating back to 1997, and a wide amount of data available. In fact, about 200.000 3D building format parcels and about 3.000 3D volumetric parcels have been submitted to the Queensland DCDB up to now [Thompson et al., 2018].

A significant subset of the current Queensland DCDB has been used for the prototype and transferred from an Ingres database to a PostgreSQL/PostGIS database, while the logical structure of the original database has been maintained. Only a table of faces, representing the boundaries of the 3D parcels, has been added to the original structure [Thompson et al., 2018].

In general, data from different origins will be used in the prototype in order to make it more realistic and user friendly:

- **2D cadastral parcels** from Queensland DCDB, recorded as polygons with line styles. The so-called "base parcels" are a subset of these 2D parcels forming a complete and non-overlapping coverage. Moreover, 2D secondary interests or easements are also recorded as polygons and overlap with the base parcels [Thompson et al., 2018].
The 2D DCDB has also a time component recorded since about 1990, therefore it can be considered a 2D+time dataset [Thompson et al., 2018].
- **3D survey plans** from Queensland Cadastre, either in *building format* and in *volumetric format*, described in Section 6.4.1 and Section 6.4.2 respectively. More precisely, the volumes will represent three main categories in the real world: buildings, tunnels and air space.

- **Rights** or registration of Rights, Restrictions and Responsibilities (**RRRs**). The availability of this kind of dataset could be related to privacy issues, in that case simulated data will be used.
- **Persons/parties** dataset if available. In case privacy issues are involved, the creation of a dummy dataset could be useful to perform queries and make the prototype more realistic.
- **2.5D terrain surface**, i.e. elevation data in the form of **DTM**, grid, Triangulated Irregular Network (**TIN**), point cloud or contour lines. The use of elevation data makes the visualization more complete and meaningful by adding ground data.
- **Reference data**, such as topographic objects in 2D or 3D (i.e. buildings, roads, etc), orthophotos and satellite images. The addition of reference objects and terrain elevation data contributes to create a reality-like environment, helping the user to recognize and orientate through the legal objects that would otherwise float on the canvas.

The area of interest is situated in the centre of Brisbane, approximately in the area of the Story Bridge and the Kangaroo Point neighborhood. This part of the city is not the most dense, but it has been chosen because it combines a variety of 2D and 3D objects: 3D building format units (described in [Section 6.4.1](#)), an underground tunnel and other parcels represented as 3D volumetric format units (described in [Section 6.4.2](#)), land and watercourse 2D spatial units and 3D parcels which changed over time. The bounding box of the area is shown in [Figure 6.2](#).

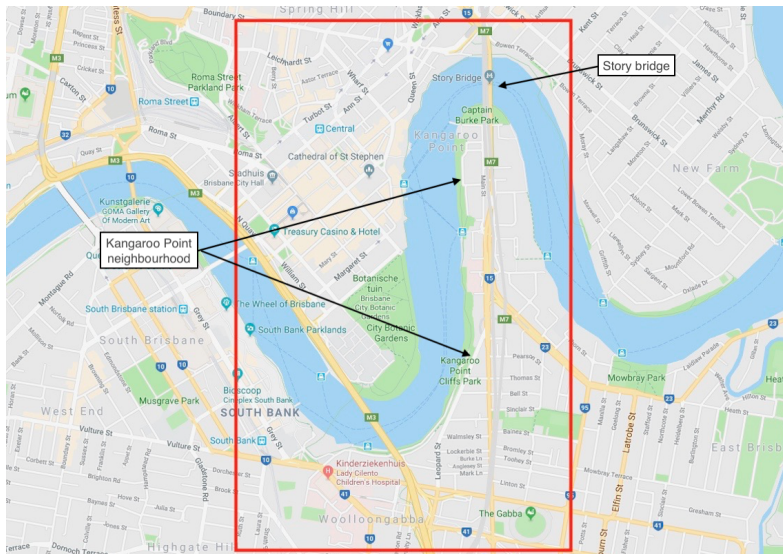


Figure 6.2: Bounding box of the Brisbane City centre including the Story Bridge and the Kangaroo Point areas.

6.3 THE STORAGE SCHEMA

All the data described in [Section 6.2](#) needs to be accommodated in a **DB**, therefore a suitable storage schema is needed.

Since this is not one of the primary goals of this research, only a short introduction about the key concepts related to the storage schema will be given.

The current data model of the Queensland cadastre does not support 3-dimensional parcels and reference objects, thus the model necessitates an update to be able to integrate those objects. Additionally, the current Queensland data model is not

compliant with the ISO's LADM standard [ISO-TC211, 2012].

In order to solve these issues, Thompson et al. [2018] developed a LADM compliant extension of the Queensland DCDB.

6.3.1 The LADM compliant storage schema

The storage schema has been designed by Thompson et al. [2018] with the purpose of being LADM compliant and accommodate a variety of data sources (see Section 6.2).

The Land Administration Domain Model provides a formal framework for describing land administration data, such as cadastral and deeds data. It is implemented by developing an application schema of the LADM model. Also, it enables involved parties, both within one country and in different countries, to communicate based on a shared vocabulary, suggested by the model.

The storage schema of the 3D cadastre prototype shows the structure of the DB. The actual schema is based on the Queensland DCDB schema with some additions (Figure 6.3). The orange tables were already present in the Queensland DCDB. Four new tables (coloured in red) have been added: *Plan*, *Polyhedron_face*, *QC_Right* and *QC_Party*. The *Unit_nr* attribute has been added in the *Plan* table, while the *ObjectID*, *Unit_nr* and *Unit_part* attributes have been added in the *Parcel* table. Finally, the "views" that have been added are indicated with a blue box and a QC prefix, for Queensland Cadastre. Details about the technical schema can be found in Appendix A.

The form of the database used for this research is rather similar to the current Queensland DCDB but some "views" have been exposed in order to be compatible with the LADM format. This brings several advantages:

- It allows loading of data from the Queensland DCDB.
- It allows the modification of the prototype database structure without interfering with the visualization.
- It provides a LADM derived protocol for the delivery of mixed 2D+t and 3D+t cadastral data.
- It indicates that a database that has not been created with LADM in mind can still support that protocol [Thompson et al., 2018].
- Outsiders (i.e. non Queensland DCDB experts) can still understand the content in the viewer, as standard terms (ISO 19152) are used.

6.4 THE DATA ENCODING PROCESS

The data encoding process is not one of the main goals of this thesis research however, without data there is nothing to visualize, so this an essential passage to prepare the data and making it usable. The process will be explained step by step to give to the reader the feeling of the complexity of the operation.

As previously mentioned, the 3D survey plans of the Brisbane DCDB are stored in plain PDF files according to the Queensland regulations [Department of Natural Resources and Mines (Queensland Government), 2013, 2016]. For this reason, an encoding process is needed to convert the information from paper to digital format.

6.4.1 Encoding *building format* spatial units

Building format spatial units are defined by the walls of the building itself, making a Building Information Model (BIM) or a building maintenance plan a good source

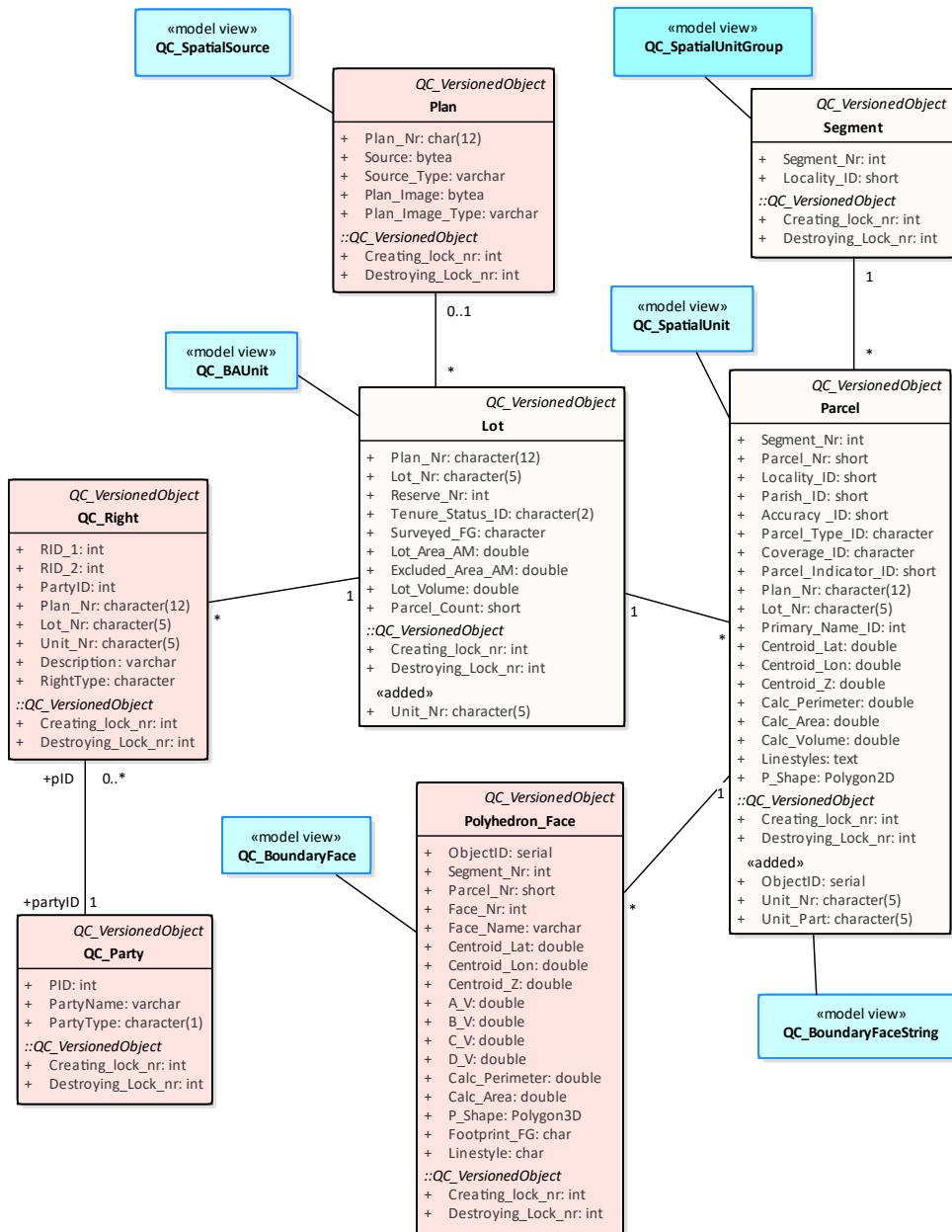


Figure 6.3: Database schema of the 3D cadastre prototype. Original tables from the Queensland DCDB are shown in light orange. New tables are shown in red, while the “views” are in blue.

of data. In this way, the data would be submitted in a digital format as a component of the survey plan. However, 3D building format plans were registered at different points in time and the only ways to submit the survey plans were paper format or, later on, PDF or TIFF formats [Thompson et al., 2018].

Figure 6.4 displays an example of building format plan showing a floor plan of a building, but no dimensions or connections to reference points are displayed because the legal definition of property is given by the physical walls of the buildings. The unique shapes of the units would allow them to be matched to similar shaped units above or below them.

In general, floor plan dimensions can be estimated in different ways. First, the sketches of the 3D spatial units can have, recognizable cadastral corners on the same sheet. Second, some plans have a scale bar present, which allows to rather accurately scale the dimensions from the PDF while encoding them. Finally, in some

circumstances detective work is needed to match the positioning of the boundaries. Height values are represented in several ways: reduced levels of the floors, heights above or below local ground level (sometimes not giving local ground level) or no Z values at all, just floor identifiers (i.e. A, B, etc.).

A custom-made software has been used for the data capture, i.e. digitization from images, due to the rather complicated process. Only the approximate position of the walls is recorded, but for this purpose can be considered acceptable since it is not the legal boundary being captured but the walls that define the legal boundary. It is important to keep in mind that the process of encoding is a time consuming operation, approximately it takes between 30 minutes to an hour per floor plan to encode an high rise building [Thompson et al., 2018].

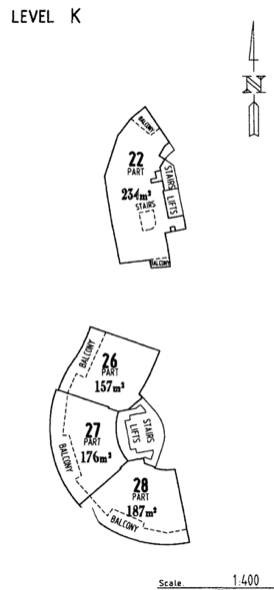


Figure 6.4: Excerpt from a building format spatial unit [Thompson et al., 2018]

6.4.2 Encoding volumetric format spatial units

Differently from the building format plan, the volumetric format spatial units record the metes and bounds of each corner. The volumes are defined thanks to the bearing and distance measurements which enable to determine the X and Y location of all the corners in the plan. The absolute Z coordinate is supplied as well.

An excerpt of the paper format of the 3D volumetric plan is shown in Figure 6.5. In particular, in Figure 6.5a the location of the corners (i.e. 91, 9, 8, 7, 93, 92) is defined relative to one another in 2-dimensions. In the isometric view in Figure 6.5b the Z values are assigned (i.e. 93a has a Z value of -48.264 and 93b has a Z value of -31.264) and the boundary faces are defined.

The values are input manually in a Microsoft Excel spreadsheet file (XLS) extracted from the 2D representation of the parcel in the DCDB, as shown in Figure 6.6.

After that, the Excel file is read by a Java code courtesy of Dr. R. J. Thompson and the 3D parcels are then loaded as 3D multi-polygons into a PostgreSQL/PostGIS DB. Note that the corner numbers (Cnr Nr) in the Excel sheet correspond to the corners of the volumetric parcel. The letter appended to the corner number indicates whether the corner is on the top (b) or bottom (a) of the parcel. The upper and lower footprints of the parcels are respectively stored counterclockwise and clockwise. The process of encoding can be summarized with the diagram in Figure 6.7.

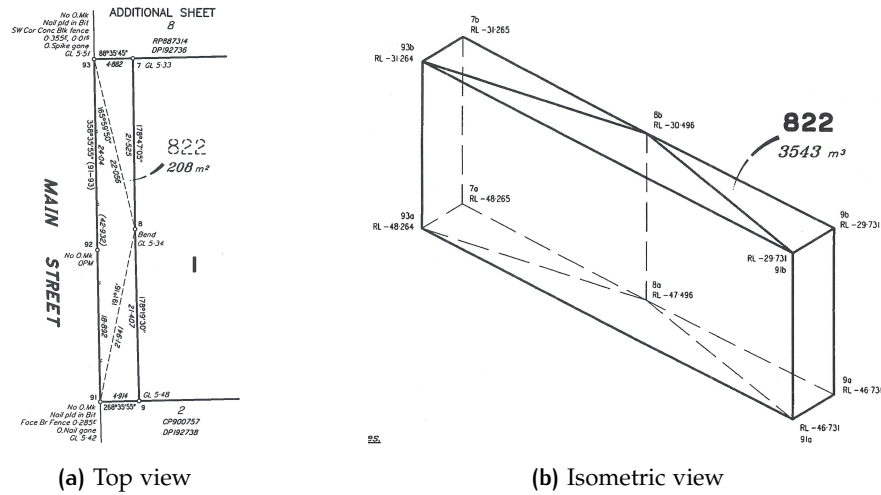


Figure 6.5: Excerpt from a 3D survey plan (SP192737)- Lot 822 [Thompson et al., 2018].

DCDB Parcel 822/SP192737 25159/233	Lon	Lat	Pid	Cnr Nr	X	Y	B and D	Elevations
S SW	153.0354519	-27.466975	-1	91	103	0		
SE E	153.0355025	-27.466974	-1	9	601	12		
	153.0354952	-27.466781	-1	8	530	2156		
N NE	153.0354911	-27.466598	-1	7	489	4181		
NW W	153.0354415	-27.466599	-1	93	0	4170		
	153.0354445	-27.466708	-1	92	30	2961		
****	additional edges if needed							
****	additional elevations if needed							
93 a -48.264 b -31.264								
91 a -46.731 b -29.731								
9 a -46.731 b -29.731								
8 a -47.496 b -30.496								
7 a -48.265 b -31.265								
****	parcels							
Parcels Lot 822 25159/233								
Footprint								
A 93a 7a 8a 9a 91a								
B 93b 91b 9b 8b 7b								
****	Textural data							
Date	06/12/2007							

Figure 6.6: Excel sheet used for the encoding of Lot 822 (SP192737). The manual data entry is highlighted in yellow [Thompson et al., 2018].

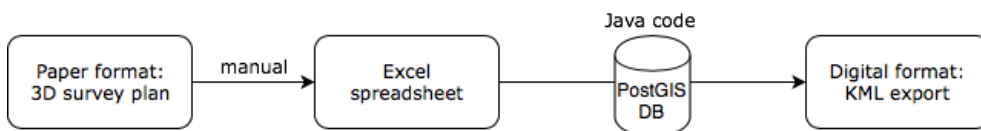


Figure 6.7: Diagram showing the main phases carried out for the encoding process.

The process of encoding is simplified due to the fact that in many cases the volumetric parcels of the Queensland DCDB have planar faces in both vertical and horizontal directions [Thompson et al., 2015]. Nevertheless, it must be underlined that, even though the encoding of the parcels is performed automatically by a computer program, the transition from the PDF/TIFF survey plan to the XLS format has to be done manually. Therefore, it is a non-trivial and time-consuming operation. The overall workload to encode a volumetric unit can vary significantly depending on the complexity of the parcel. Appendix B shows the entire survey plan document of a volumetric parcel of the Brisbane DCDB.

6.4.3 Outcome of the encoding process

As already mentioned earlier, capturing the 3D parcels is a difficult task, but some progress has been made for both types of spatial units, as shown in Figure 6.8. Figure 6.8a shows an example of building format units in KML format extracted

from the PostgreSQL/PostGIS database after the encoding process. The three units consist of five buildings (two of the plans are composed by two buildings each), made of 349 individual units which also include car parks and balconies; therefore the real number of units amounts to about 200 [Thompson et al., 2018].

In Figure 6.8b the volumetric format plans are encoded and imported in the Google Earth web viewer. The plans include the “Clam 7” tunnel, which is not represented by one unique parcel but from many individual parcels. The reason of this splitting comes from the fragmentation of the the volume of the tunnel at the boundaries of the 2D surface parcels. The “tall” parcel which runs from the tunnel level up to a height of 200 meters above ground does not relate to a construction, but it has probably been given that shape to prevent building above the tunnel access point. The long and narrow parcel on the right side of the image represents air space under a segment of the Story Bridge ramp.

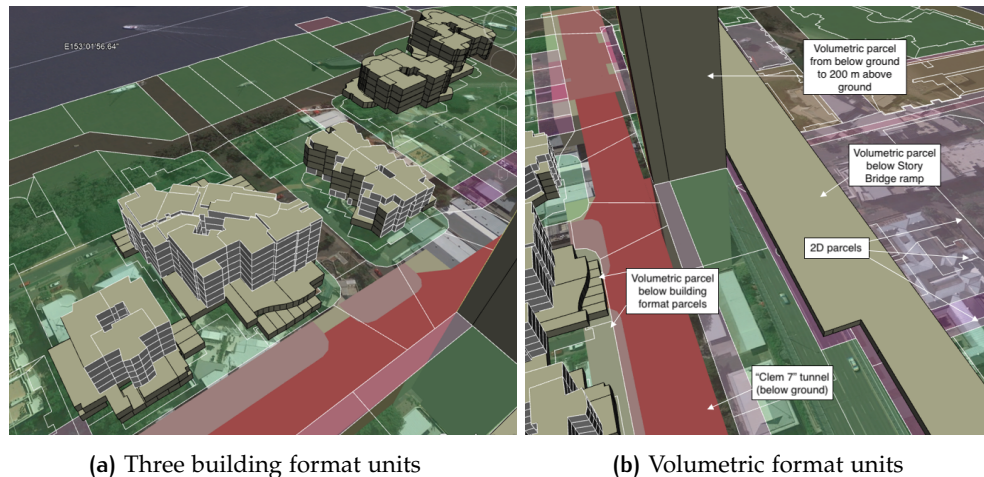


Figure 6.8: Building and volumetric format units imported in Google Maps [Thompson et al., 2018]. Note that, due to the impossibility of visualizing underground objects in Google Earth, all parcels have been raised 50 meters above ground.

In addition, to make the process easier, the bespoke software created by Dr. R.J. Thompson provides a pop-up window which allows the user to extract the desired KML files from the database (see Figure 6.9). Three different choices are possible to select the desired legal information: inserting *Unit/lot/plan* number, *Segment/parcel* number or *Window* (which includes all the encoded parcels). It is possible to choose among 2D, 3D or mixed 2D and 3D data and to insert an offset in order to shift vertically the parcels for visualization purposes. Moreover, it is also possible to choose a *DTM* to use as a reference surface for the parcels, this will drape the 2D parcels on the *DTM* surface to create 2.5D parcels.

Once the options are set, the user clicks the *Output KML* button which saves the KML files in the local machine.

6.5 SELECTION OF THE DTM

The selection of the *DTM* is an important step to build a successful prototype, the terrain surface contributes creating context and orientation for the user. As already mentioned in Section 5.1.2, Cesium JS allows to easily switch between a 3-dimensional globe, a 2-dimensional map and a 2.5 dimensions Columbus view, but these reference surfaces do not provide any terrain elevation data, so it is necessary to import it from external sources.

The *DTM* should fulfill some prerequisites in order to be suitable for 3D cadastre vi-

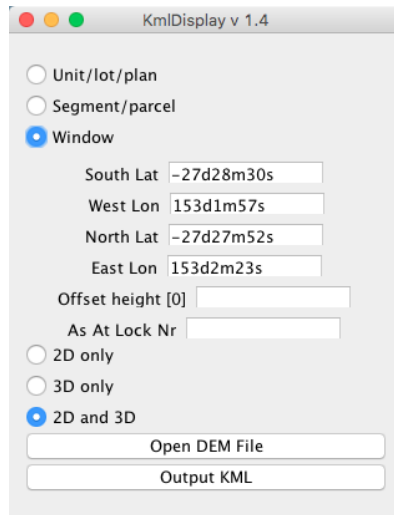


Figure 6.9: Pop-up window containing the options that can be chosen to extract the KML files from the database.

sualization. First of all, the resolution should be of at least 0.5/1 meter, and secondly the data format should be supported by Cesium JS. Finally, it should be possible to easily combine it with the [KML](#) parcels, therefore a [DTM](#) in [KML](#) format would be the ideal solution.

6.5.1 The STK World Terrain

STK World Terrain is a worldwide terrain elevation model in high resolution provided by Analytical Graphics, Inc. (AGI), which keeps growing in resolution when new sources of data are available. The approximate maximum spacing between height samples in the area of Brisbane is 10 to 30 meters [Cesium, 2018]. Although this resolution could be suitable for various applications, the visualization of 3D cadastral parcels requires more accurate matching between the surface and the parcels. As can be seen in [Figure 6.10](#), the STK World Terrain represents the city of Brisbane as a mountainous area, probably because it takes into account the height values of the high rise buildings of the Kangaroo Point neighborhood as part of the terrain surface. Consequently, this elevation data was not considered appropriate to develop the cadastre prototype, and other options need to be explored.

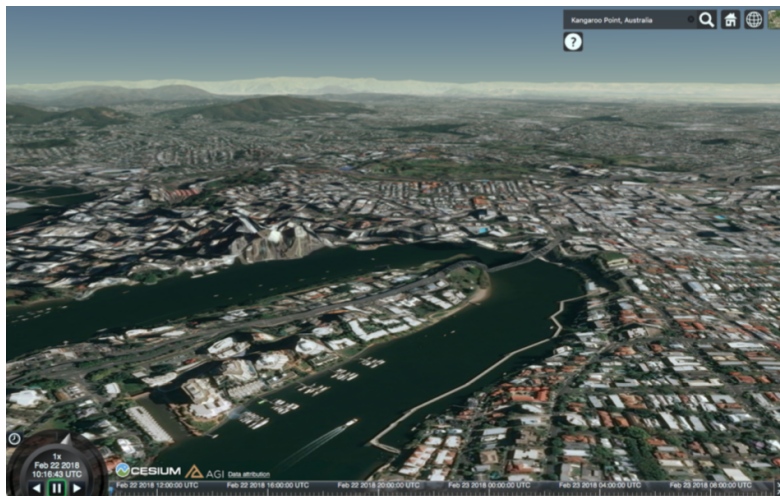


Figure 6.10: STK World Terrain loaded in Cesium JS.

In addition, the STK World Terrain was deprecated since March 1st 2018. Its substitute, Cesium World Terrain, hosted by the commercial Cesium Ion platform, will replace it starting from September 1st 2018. Although the resolution of this terrain model is significantly higher in many areas of the world, the populated areas and coasts of Australia have a resolution of about 5 meters, which is still not accurate enough for the purpose of this research.

Nevertheless, Cesium JS promised to make available in the coming months many terrain data updates to improve the resolution in different areas of the world in order to achieve an upgraded terrain experience, which will culminate with the possibility of having the terrain being enabled by default [Cesium, 2018].

6.5.2 The DTM provided by Fugro

The search for suitable terrain elevation data continues. The possible sources are point clouds in LASer File Format (LAS), TIN and grid/raster (TIFF) format files.

Fugro, an international survey company based in the Netherlands, which is active in the oil and gas industry, in the telecommunications and infrastructure fields, has been asked to be the data provider, due to numerous collaborations with Delft University of Technology.

The elevation data supplied by Fugro is in grid format, in particular a TIFF file with a resolution of 50 cm, hence accurate enough for the purpose of 3D cadastre. As can be seen in Figure 6.11 the DTM is divided into square tiles of equal dimension, their borders are clearly detectable due to the different intervals set in the grey scales of the neighboring tiles. In other words, in the proximity of the tile borders, similar height values are assigned to different grey values, so this does not compromise the accuracy of the DTM, as it might be concluded from a first look.

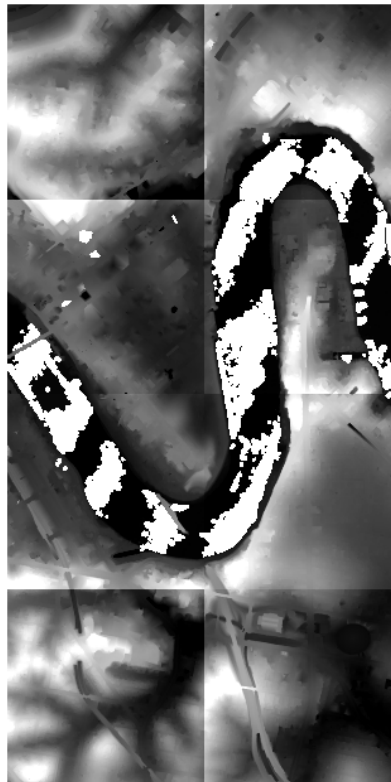


Figure 6.11: Elevation data of the Brisbane City centre in raster format (courtesy of Fugro).

6.5.3 Conversion of the DTM

The elevation data needs to be loaded into Cesium JS with the purpose of providing context to the legal data. In order to be loaded, the [DTM](#) needs to be converted into a suitable format. Three steps were necessary to convert the data from Geospatial Tagged Image File Format ([GeoTIFF](#)) to [KML](#):

1. *From GeoTIFF to ESRI ASCII grid*

The [GeoTIFF](#) format is similar to [TIFF](#) with the addition of geo-referenced information, for instance map projections, coordinates systems, datum, ellipsoids and all the information useful to geo-referencing the file [[OSGeo, 2018](#)]. On the other hand, the American Standard Code for Information Interchange ([ASCII](#)) grid format can be used to transfer information to or from other raster files. The ESRI [ASCII](#) file will then record the information about the input file in an header followed by cell value information, as shown in [Figure 6.12](#). Note that the X and Y grid values are implicit, while the height values are explicit. The conversion has been carried out with the software package FME by Safe Software as a courtesy of Drs. Marianne de Vries.

```
ncols 2000
nrows 2000
xllcorner 503000
yllcorner 6961000
cellsize 0.5
nodata_value -10000
10.187972068786621 10.162620544433594 10.139581680297852
10.118855476379395 10.098129272460938 10.077402114868164
10.049394607543945 10.014105796813965 9.978816032409668
9.9435272216796875 9.9193716049194336 9.9063491821289063
...
...
1.3176814317703247 1.3226256370544434 1.3277195692062378
```

Figure 6.12: Structure of the ESRI ASCII grid containing the DTM information.

2. *From *.asc to comma delimited ASCII *.xyz*

The second step is the conversion into a space or comma delimited [ASCII *.xyz](#) file format. The [ASCII XYZ](#) format is a data format for raster data. In [Figure 6.13](#) we can see the structure of the [DTM](#) file used in the project. This part of the transformation is courtesy of Dr. Rodney J. Thompson by means of a specific software package.

3. *From comma delimited ASCII *.xyz to KML*

The last phase is the conversion into a [KML](#) file, which can then be input in the Cesium JS web viewer. This process takes place in the custom made software created by Dr. Rodney J. Thompson to encode the parcels, described in [Section 6.4.3](#).

6.6 DISPLAY OF 3D OBJECTS

Once the parcels have been correctly encoded, they are ready to be imported into the web viewer. Given that the output of the encoding process produces [KML](#) files (from 3D polygons stored in the PostgreSQL/PostGIS [DB](#)), it has been decided to start experimenting with that data format.

In the course of the description we will examine the main drawbacks related to this data format and we will consider other solutions supported by Cesium JS, such as [JSON](#) and [gITF](#).


```

503000.25,6961999.75,10.1879720687866
503000.75,6961999.75,10.1626205444336
503001.25,6961999.75,10.1395816802979
503001.75,6961999.75,10.1188554763794
503002.25,6961999.75,10.0981292724609
503002.75,6961999.75,10.0774021148682
503003.25,6961999.75,10.0493946075439
503003.75,6961999.75,10.014105796814
503004.25,6961999.75,9.97881603240967
503004.75,6961999.75,9.94352722167969
503005.25,6961999.75,9.91937160491943
503005.75,6961999.75,9.90634918212891
503006.25,6961999.75,9.89332580566406
503006.75,6961999.75,9.88030433654785
503007.25,6961999.75,9.87665367126465
503007.75,6961999.75,9.88237285614014
503008.25,6961999.75,9.88809394836426
503008.75,6961999.75,9.89381313323975
503009.25,6961999.75,9.900071144104
...
...
503999.75,6961000.25,1.32771956920624

```

Figure 6.13: Structure of the ASCII XYZ file containing the DTM information.

6.6.1 About KML

The [KML](#) file format became an international standard in 2008 when Google submitted it to the [OGC](#) that, since then, maintains and evolves the standard. The [KML](#) presents a tag-based structure built on top of [XML](#) grammar. The tags are case sensitive and consist of nested elements and attributes. An example of the structure of the [KML](#) file is shown in [Figure 6.14](#)

This kind of format is generally used to encode geographic data to be visualized on a virtual globe, such as Google Earth, Cesium JS and many others. Moreover, the [KML](#) standard is open and non-proprietary, which makes it more suitable for dissemination purposes [[OGC, 2015](#)]. Being [KML](#) one of the most popular data formats in the 3D industry, it is well supported by Google Earth, which has also been used to check the appearance of the [KML](#) files generated by the encoding process ([Figure 6.15](#)). [Figure 6.15a](#) shows the 2D parcels, while [Figure 6.15b](#) shows the 3D parcels imported in Google Earth.

Now, the attentive reader would think: why not using Google Earth for the development of the 3D cadastre prototype, since [KML](#) is its native format? Before answering this question we need to keep in mind the reasons why [WebGL](#) has been selected as underlying technology for the web-viewer: (1) good performance and speed, (2) platform and browser independence, and (3) open source/extensibility. Google Earth performs very well in case of complex visualizations, but it does not fulfill completely the other requirements. Therefore, the answer can be summarized as follows:

- One of the big advantages of [WebGL](#) is that it is supported by all major browsers without the installation of a plug-in. On the other hand, the current version of Google Earth is based on both [WebGL](#) and the Native Client technology, the latter is (for now) supported only by Google Chrome, which needs to be installed in the local machine if the user has another browser. Nevertheless, in the FAQs related to the latest Google Earth implementation one can read that, in the future, Google Earth will be available for most of the major browser vendors [[Google Inc., 2018](#)]. Therefore, this makes Google Earth plat-

```

<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.0">
  <Document>
    <Style id="pclL">
      <LineStyle>
        <color>ff64ff64</color>
        <outline>1</outline>
        <colorMode>normal</colorMode>
        <width>1</width>
      </LineStyle>
      <PolyStyle>
        <fill>1</fill>
        <color>6464ff64</color>
      </PolyStyle>
    </Style>
    <Style id="pclR">
      <LineStyle>
        <color>ff70808c</color>
        <outline>1</outline>
        <colorMode>normal</colorMode>
        <width>1</width>
      </LineStyle>
      <PolyStyle>
        <fill>1</fill>
        <color>c870808c</color>
      </PolyStyle>
    </Style>
    ...
    <Folder id="ParcelType-E">
      <name>ParcelType-E</name>
      <Folder id="25200.64">
        <styleUrl>#pclE</styleUrl>
        <name>D/RP859295</name>
        <description>
          <![CDATA[
<p>seg/par = 25200/64<br>
Parcel type: EE<br>
Create = 2495105<br>
Destroy = 2147483647</p>
]]>
          </description>
          <Placemark>
            <styleUrl>#pclE</styleUrl>
            <name>D/RP859295</name>
            <description>
              <![CDATA[
<p>seg/par = 25200/64<br>
Parcel type: EE<br>
Create = 2495105<br>
Destroy = 2147483647</p>
]]>
              </description>
              <Polygon>
                <altitudeMode>absolute</altitudeMode>
                <outerBoundaryIs>
                  <extrude>1</extrude>
                  <LinearRing>
                    <coordinates>
                      153.03704026341438,-27.474857971072197,24.1
                      153.03704879432917,-27.47484987974167,24.1
                      153.0370507761836,-27.474866650998592,24.1
                      153.03704026341438,-27.474857971072197,24.1
                    </coordinates>
                  </LinearRing>
                </outerBoundaryIs>
              </Polygon>
            </Placemark>
          </Folder>
        </Folder>
      </Folder>
    </Document>
  </kml>

```

Figure 6.14: Example structure of KML encoding.



(a) 2D land and water parcels are displayed in different colours (b) 3D parcels (shifted above ground to be visible in Google Earth)

Figure 6.15: KML files imported in Google Earth.

form independent (only when using Google Chrome as web browser), but not browser independent.

- One of the most important requirements related to the web viewer is its extensibility in order to allow the developers to add the desired functionalities. Google Earth is a closed source platform and therefore it does not allow to view and/or modify the source code. On the other hand, thanks to the Earth Engine JavaScript and Python *APIs* the user can implement new functionalities, perform queries and run algorithms [Google Inc., 2018].
- Lastly, Google Earth does not allow to look under the ground surface (except in some particular cases in which the camera goes under the surface, but it is not free to navigate around), neither making the globe surface transparent, two huge shortcomings when dealing with 3D cadastre visualization.

6.6.2 Exploring the possibilities of KML (and other formats) in Cesium JS

After importing the cadastral parcels in Google Earth, the same thing has been done in Cesium JS. The process went smoothly and the parcels were automatically placed in the correct location, without any issues related to the Coordinate Reference System (CRS). Moreover, the styling of the volumes, i.e. filling and outline colour of the parcels, was consistent with the one declared in the *KML* tags.

The first impression was therefore highly positive, but the issues will start later in the implementation process, while dealing with more complicated issues that go beyond the simple loading of the files, for example the feature selection and highlight described in Section 6.10.

The first *KML* implementation in Cesium JS dates back to 2013 in occasion of the Google Summer of Code, when a considerable amount of core features were implemented. Nowadays, a large amount of the *KML* standard is supported in Cesium JS, although the support is still not complete. Moreover, being *KML* a large specification, the support is still under development and new features will be presumably available in the following Cesium JS releases [Cesium, 2018].

For more detailed information about what are the supported and not supported features, see the official Cesium JS repository by Analytical Graphics Inc. [2017].

With respect to the issues relative to the partial support of *KML* in Cesium JS, other data encodings have been considered for the display of the 2D and 3D parcels:

JSON

The *JSON* format is well supported in Cesium JS and it is particularly suited for 2D data and for 3D data generated from extrusion of the 2D footprints. In short,

this means that [JSON](#) does not allow to create “real” 3D data, but only extruded 2D volumes. Specifically, Cesium JS provides a function called “extrudedHeight” which sets the value for the polygon extrusion to create the 3D volume [[Cesium, 2018](#)].

The disadvantage of using this format in Cesium JS, in the field of 3D cadastre, is that the shape of legal spaces is, in many cases, more complex and extruded volumes are not able to represent the administrative parcels accurately enough.

glTF (and B3DM)

The [glTF](#) format is known as *the* 3D model format for Cesium JS and an essential part of 3D tiles [[Cesium, 2018](#)]. Numerous examples are available in the Sandcastle live code editor provided by Cesium JS.

Differently from the [JSON](#) format, [glTF](#) supports “real” 3D data making it a reliable candidate for the display of 3D cadastral data.

On the other hand, the creation of this data format is not so straightforward. First, the custom made software for the encoding of the parcels should be modified in order to output [glTF](#) directly or an intermediate (and easier) format, the Relocatable Object Module Format ([OBJ](#)). Then, the [OBJ](#) files must be converted into [glTF](#) by means of a special converter provided by Cesium JS [[Cesium, 2018](#)].

In general, this is a technologically feasible operation, but once again the time restrictions do not allow to investigate further on this aspect.

Additionally, the 3D tiles streaming format defines the Batched 3D Model ([B3DM](#)) format used in Cesium JS. This data format merges visual elements and additional attributes related to the objects, and it defines the concept of batches, which allows to identify objects and assign properties such as feature type, unique ID and custom attributes. These properties linked to the objects can be used for highlighting functionalities, for customizing the features and for querying the web services and retrieving additional information (based on the unique ID). As the name suggests, [B3DM](#) is also used to “batch” data, i.e. group/create blocks of data, which can be useful for server-client caching techniques. [Figure 6.16](#) shows the structure of a [B3DM](#) file made of a batch table (containing all the object properties), a binary [glTF](#) block and a header. The batches provide an extremely efficient way of identifying objects; they define each feature by means of its vertices, improving significantly the performance [[Schilling et al., 2016](#)].

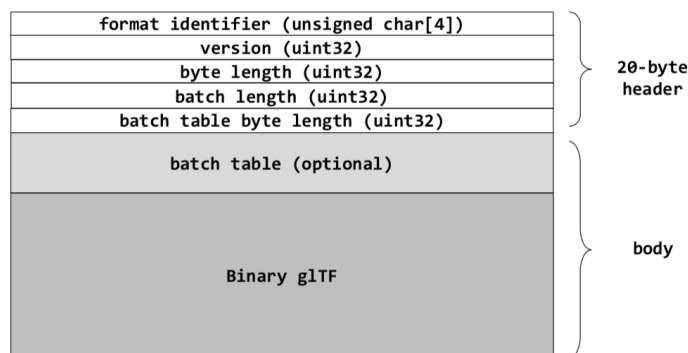


Figure 6.16: Structure of a [B3DM](#) file [[Schilling et al., 2016](#)]

To sum up, it has been decided not to use [JSON](#) and [glTF](#) to encode the 3D geometries because of suitability and technology issues respectively. [JSON](#) did not result suitable for 3D cadastre visualization purposes due to the extruded nature of the 3D volumes, while [glTF](#) (or [B3DM](#)) would be appropriate, but changing the bespoke Java program to output [glTF](#) instead of [KML](#) would not be feasible in the short term.

6.7 LAYERS CONTROL

Layers control has been described in [Section 4.2.3](#) as the possibility for the user to add and remove layers, toggle on/off their visibility and modify their transparency. In the 3D cadastre prototype, the layers control functionality has been implemented to switch off the 2D cadastral parcels and better visualize the 3D parcels, both above and below ground.

The tool has been implemented with a checkbox. At start up, the checkbox is enabled, which means that the 2D parcels are visible, therefore the user can switch off the layer according to the task that he wants to perform.

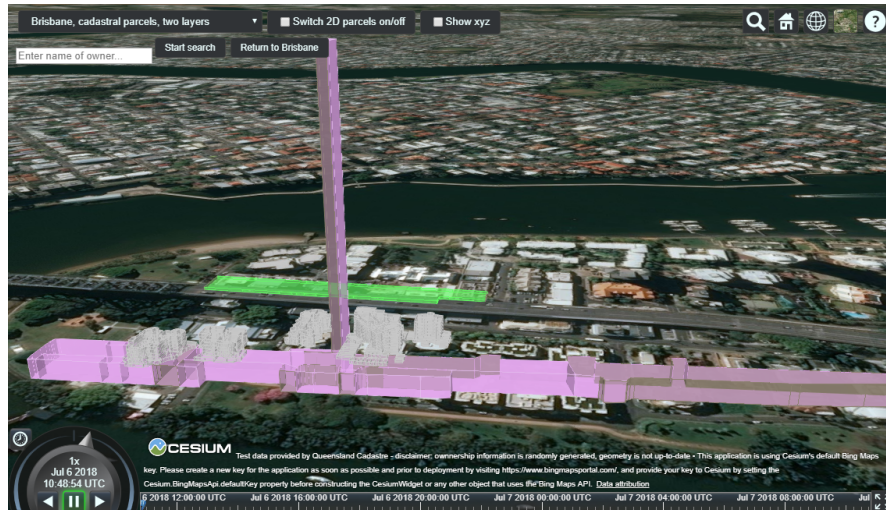


Figure 6.17: The 2D parcels are switched off, so that the 3D ones can be better visualized. Note: the “Switch 2D parcels on/off” checkbox is unchecked.

6.8 CAMERA START-UP POSITION

The positioning of the camera at start-up was not included in the initial list of requirements, although it has been added to improve the user-friendliness of the prototype. This functionality consists of placing the camera in the right position (i.e. pointing at the Brisbane city centre) at the start up of the viewer. In this way, the user does not have to manually navigate to the location. An excerpt of the JavaScript code used to set the initial position of the camera on Brisbane is shown in [Figure 6.18](#).

The `Cesium.Cartesian3.fromDegrees` method sets the initial position over the city centre of Brisbane by setting the latitude, longitude and height of the camera initial position. The `Cesium.HeadingPitchRoll.fromDegrees` method defines the orientation of the camera by means of heading, pitch and roll, with the angles of the rotation expressed in degrees [[Cesium, 2018](#)].

In addition, if the user navigates to another location, the “return to Brisbane” button resets the initial location.

6.9 SUBSURFACE VISUALIZATION

Subsurface visualization, as described in [Section 4.2.9](#), concerns the visualization of the 3D parcels with respect to the earth surface; namely, whether it is possible to properly visualize parcels that are located below the ground surface.

The main issue is that the topic of “underground geometries” is not included in the

```

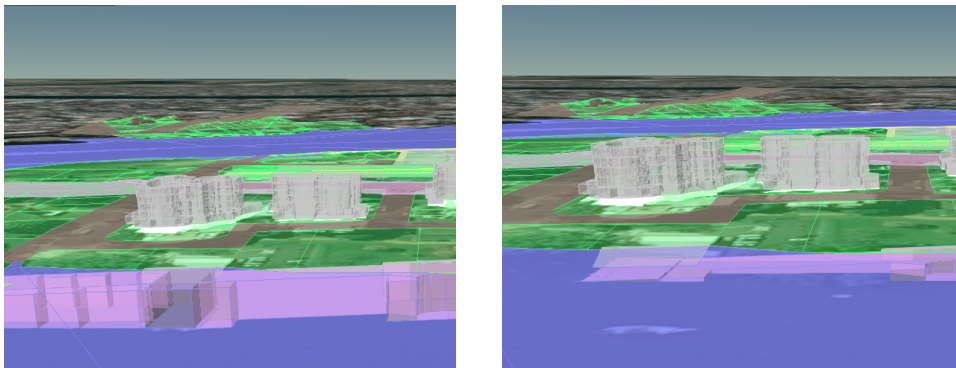
1 // Set the initial camera view to look at Brisbane
2 var initialPosition = Cesium.Cartesian3.fromDegrees(153.0331294,
3   -27.4721613, 553);
4 var initialOrientation = new Cesium.HeadingPitchRoll.fromDegrees
5   (21.27879878293835, -21.34390550872461, 0.0716951918898415);
6
7 viewer.scene.camera.setView({
8   destination: initialPosition,
9   orientation: initialOrientation,
10  endTransform: Cesium.Matrix4.IDENTITY
11 });
12
13 var options = {
14   camera : viewer.scene.camera,
15   canvas : viewer.scene.canvas
16 };

```

Figure 6.18: Excerpt from the JavaScript code used to set the camera initial position on Brisbane, Australia.

official documentation of Cesium JS and, therefore, not (yet) supported. Furthermore, Cesium JS is a virtual globe, these kind of techniques are relatively recent and they were not developed with the objective of subsurface visualization, which remains one of the main technological challenges of virtual globes [Ziolkowska and Reyes, 2016].

Curiously, the underground parcels can be seen “from above” when loaded in Cesium JS, as when the surface is in transparent mode. Figure 6.19 shows that zooming too close to the parcels makes them disappear and, in any case, the issue of underground visualization still needs to be solved, since it is not possible to view them from below.



(a) The underground parcels can be seen “from above”. (b) Zooming too close makes the underground parcels disappear.

Figure 6.19: The underground parcels loaded in Cesium JS can be seen “from above” as if the ground surface was in transparent mode, but there are some limitations while zooming too close to the objects.

6.9.1 Three alternatives to overcome the problem of subsurface visualization

Since subsurface visualization is a very popular discussion topic in the Cesium JS community, several developers found and implemented different options and workarounds to overcome this problem. Three possible alternatives are proposed:

- The *Ground push plug-in*, which allows the user to define a rectangle and push the surface of that rectangle up or down of a desired amount (Figure 6.20). The shape of the push rectangle cannot be changed or adapted to a specific parcel once initialized, which means that the rectangle should be big enough to fit also the bigger parcels. In addition, zooming too close to the ground surface can make the rectangle disappear due to the impossibility for the Cesium camera to go below the earth surface [Cesium, 2018].

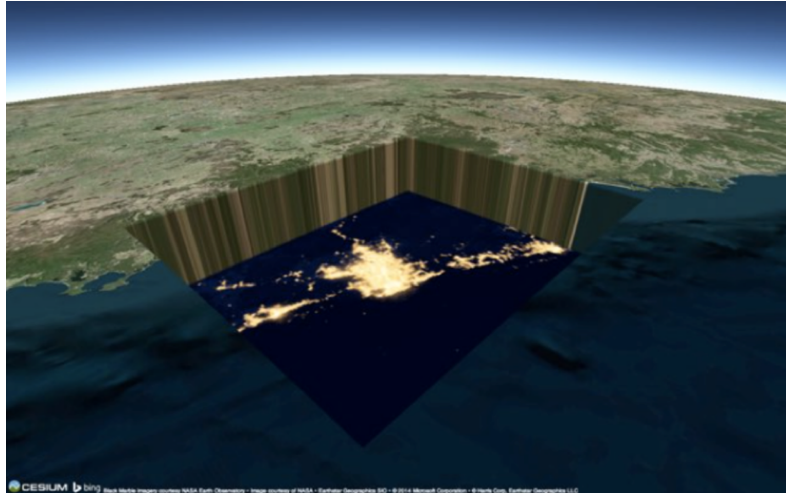


Figure 6.20: Cesium JS Ground Push Plug-in [Cesium, 2018]

- *Making the terrain translucent* is another option to obtain the underground effect. In this case, as in the Ground Push Plug in, the camera does not have the freedom to move in every direction, and presumably it is only possible to visualize the underground parcel "from above". Therefore, zooming close to a parcel to visualize its boundaries would not be possible. Figure 6.21 shows a 3D volume partly above and partly below the earth surface, the effect of translucency can be clearly seen on the lower part of the volume.

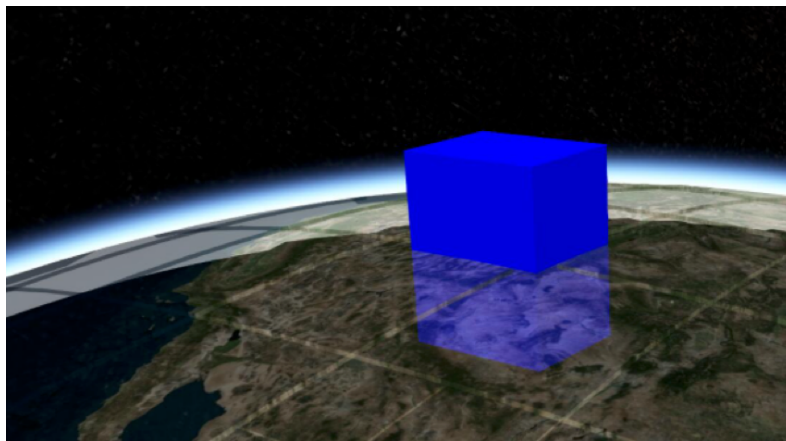


Figure 6.21: Translucent terrain in Cesium JS [Cesium, 2018]

- Using a *fake surface* on top of the actual terrain surface. In this way, there will be actually two surfaces, and the original one will be ignored. This will allow to overcome the problem of the camera, as this is blocked only by the original surface, but it will generate some issues regarding the height values which will not be reliable. As a consequence, also parcels (and reference objects)

need to be shifted of a predefined amount to match the height of the fake surface.

This method has been widely used (in combination with transparency) for the visualization of subsurface earthquakes (Figure 6.22).

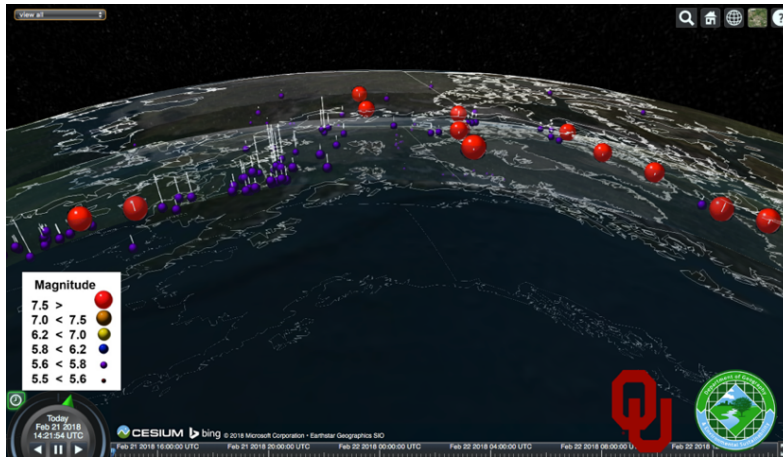


Figure 6.22: Using a fake surface for the visualization of subsurface earthquakes [Cesium, 2018]

In an early stage of the prototype development, a combination of fake surface and translucent terrain has been used with the purpose of visualizing the underground tunnel in the city centre of Brisbane, which was otherwise hidden by the earth surface.

Figure 6.23 shows the DTM and the 2D parcels draped on it styled with a certain degree of transparency, together with the 3D parcels, everything is raised of about 100 meters above ground to simulate a fake surface. This solution allows the user to visualize underground objects, but creates confusion in the perception of position, size and shape of the parcels. We are now looking at the parcels from an oblique perspective, so that the underneath globe results shifted of a certain amount and it does not match with the parcels. A new solution needs to be implemented in order to overcome this problem.

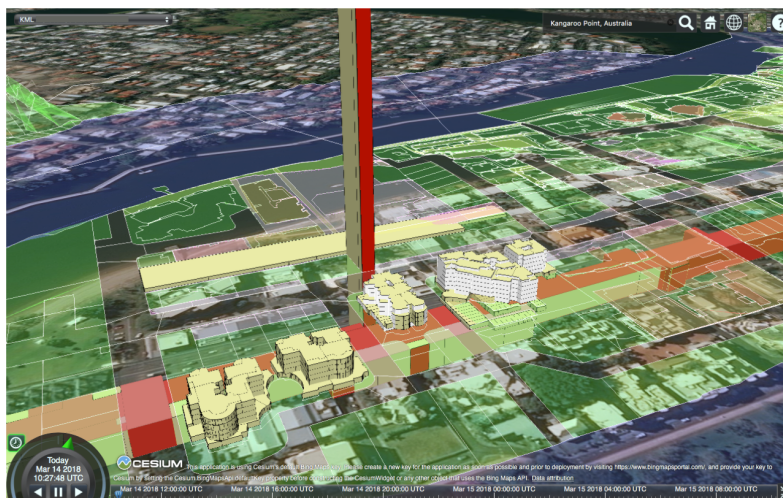


Figure 6.23: Combination of fake surface and transparent terrain in the early stage of the development of the prototype.

6.9.2 Development of a new requirement: the dynamic elevation tool

In [Section 4.1](#) a list of requirements related to 3D visualization challenges has been developed. However, a new requirement is needed to overcome the problem of underground visualization. The *dynamic elevation tool* would provide the possibility to move the 3D objects up or down of a user defined amount to be able to visualize in detail the underground parcels that are hidden by the earth surface.

The advantage of using this tool is that the user can have a reality-like visualization at the viewer start up (i.e. with the parcels in the correct location and height) and, if needed, alter the visualization in order to better visualize the hidden parcels. This results in a better understanding of the situation, the user still has a feeling of the actual context of the parcels before the visualization gets distorted to visualize the details.

Different implementations of the dynamic elevation tool are possible according to the set of objects to be shifted, the direction of the shift and the graphical control element used to power the tool:

- *Set of objects*

The set of objects to be considered is central in the implementation of the dynamic elevation tool since it determines which and how many objects are altered from their initial position.

A single parcel could be shifted in case the user needs to analyze in detail the boundaries of that specific parcel, without the context provided by the surrounding objects.

A (sub-)set of parcels could be used when the user needs to inspect an underground object composed of multiple parcels (i.e. the "Clam 7" tunnel in the Brisbane city centre)

All the parcels could be shifted if there are no occlusion issues involved in the visualization and the user wants to have a comprehensive view of both above ground and underground parcels.

In addition, the set of object could grow (i.e. adding new objects to the set) or shrink (i.e. removing objects from the set) according to the administrative/s-patial queries that the users wishes to perform.

- *Direction*

The term refers to the orientation of the shifting of objects with respect to their initial position.

Vertical. Shifting the parcels in vertical direction, from the bottom towards the top, is probably the most straightforward way to implement this tool.

Horizontal. After the parcel has been raised above ground, a horizontal shift could be applied in case the view of the parcel is occluded by other parcels. This functionality is similar to the sliding tool described in [Section 4.1.9](#).

In every direction. Nevertheless, in case the user requests a higher degree of freedom, the tool could be implemented by allowing the shift in every direction, maybe also supporting rotations of objects.

- *Graphical control element*

The graphical control element in a Graphical User Interface (GUI) refers to an element of interaction.

Drop-down menu. The use of a combo box/drop-down menu allows the user to choose between a limited set of predefined Z values to shift the (selected) parcels.

Text entry box. Using a text entry box allows the user to type any Z value without minimum and maximum value restrictions. Although, the risk is that the users type an invalid or non meaningful value.

Interactive slider. The implementation of a slider/track bar is, likely, one of the most advanced options for the implementation of the dynamic elevation tool, since it allows to perform an interactive elevation on-the-fly and it does not have the problem of inserting invalid values.

For the purpose of this research, the dynamic elevation tool has been implemented by means of a drop-down menu on the whole set of parcels (included the DTM) supporting a shift in the vertical direction only. In Figure 6.24 an example of the dynamic elevation tool is displayed, in particular Figure 6.24a represents a view of the parcels before the shift, while in Figure 6.24b the parcels are shifted of a defined amount to allow the user to visualize the underground objects. Eventually, Figure 6.25 shows the details of the “Clam 7” underground tunnel.

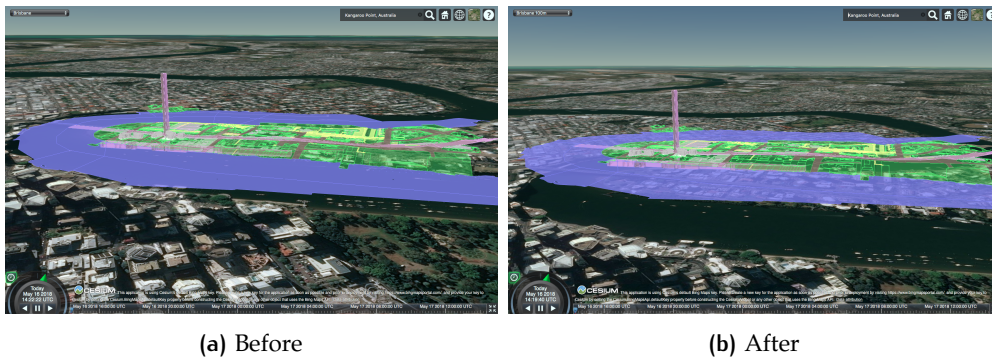


Figure 6.24: Dynamic elevation tool: initial position of the parcels (before using the tool) and parcels shifted to 100 meters above the ground (after using the tool).

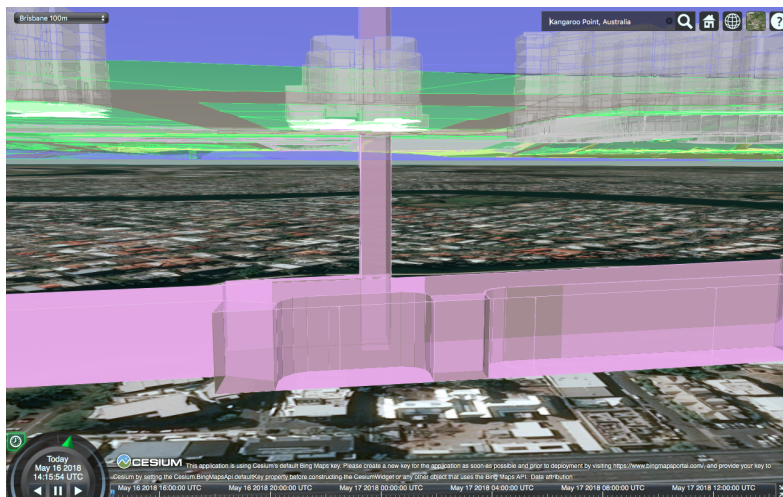


Figure 6.25: Zoom to underground parcels is possible thanks to the dynamic elevation tool.

6.10 FEATURE SELECTION AND HIGHLIGHT

The feature selection functionality, and consequent highlight, has been described in the list of requirements related to 3D visualization in Section 4.1.5 as a crucial tool to support interaction in any 3D visualization system. For the purpose of this prototype, this functionality would allow the user to select one or more parcels and

highlight them in a different colour, with the possibility of further processing the selected subset.

Since the experimentation started with [KML](#), it has been decided to continue implementing the 3D cadastre prototype using that format, even though no examples about highlighting functionalities using [KML](#) data are present in the Sandcastle showcase library.

6.10.1 Limitations of KML in Cesium JS

So far, no issues were encountered regarding the implementation of the requirements using the parcels encoded in [KML](#) format. However, now complications are arising due to the limitations of the [KML](#) implementation made by Cesium JS.

A first attempt to implement the feature selection tool has been made, but with limited results. As a rule, the `scene.pick` method in Cesium JS returns the first (top) primitive in the scene and it can be enabled by a (left or right) mouse click to make it interactive. In addition, the styling and colour of the feature can be changed according to the user needs. Although this might seem rather straightforward, its implementation combined with a [KML](#) data format does not lead to the expected result.

The outcome obtained is a selection of all the parcels which originally had the same colour, while attempting to highlight only a single parcel. Examples of this are shown in [Figure 6.26](#) for 2D parcels and in [Figure 6.27](#) for 3D parcels.

The rationale behind this phenomena can be found in the way in which the [KML](#) is structured. The features in the file are organized into folders (i.e. the `<folder>` tags in the [KML](#) file), therefore the `scene.pick` method selects the whole folder and all the objects in that folder will be consequently highlighted.

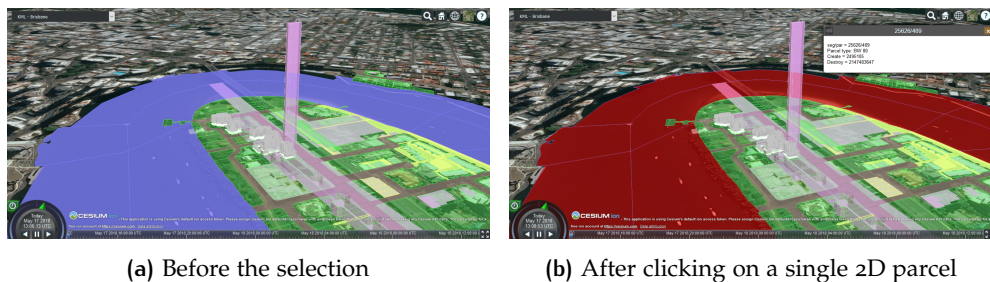


Figure 6.26: Outcome of the highlight functionality on 2D parcels using the `scene.Pick` method.

A second attempt has been made, this time using another method provided by Cesium JS, the `scene.drillPick` method, which returns a list of objects, each one containing a primitive property. Since the primitives in the list are ordered from front to back as they are in the scene it is necessary to loop through the features to pick the last one in the list, which will be the desired object. [Figure 6.28](#) shows the feature selection and highlighting tool as implemented in the prototype. It must be noted that, the functionality is still not working perfectly so, it is suggested to switch off the 2D parcels before selecting the 3D ones, otherwise they might get wrongly highlighted. This probably occurs because [KML](#) selection is based on ray casting and the ray is likely to pass through the parcels and highlight objects that are behind them.

In order to solve this issue two options could be adopted:

- *Use of another data format, such as JSON, glTF (or B3DM), etc.*
Choosing this option would solve the problem of the imprecise selection of the objects.

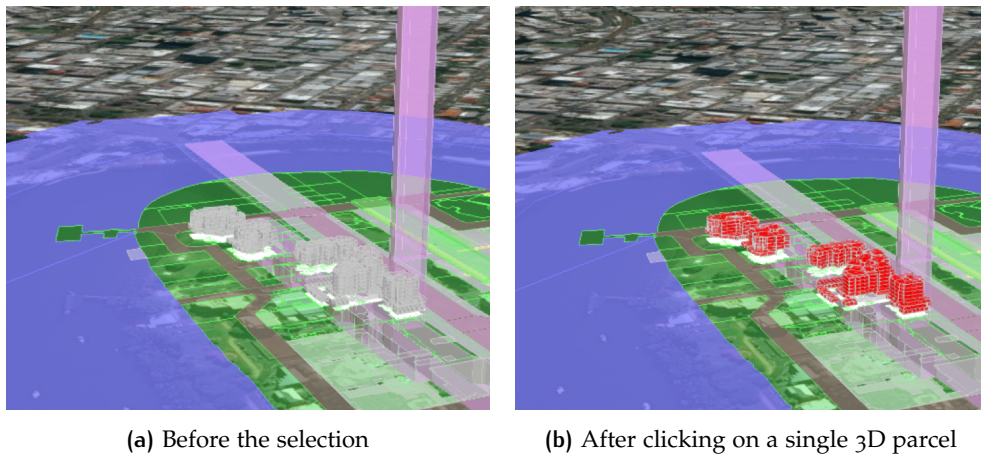


Figure 6.27: Outcome of the highlight functionality on 3D parcels using the `scene.Pick` method.

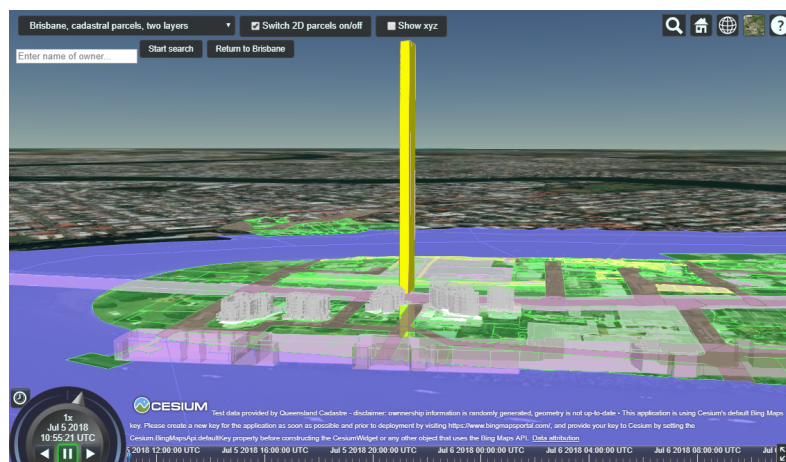


Figure 6.28: Second attempt to implement the highlight functionality using the `scene.drillPick` method.

Plenty of examples of 3D tiles are contained in the showcase library of Cesium JS. In particular, the highlight functionality is developed using the `B3DM` format (introduced in [Section 6.6.2](#)), which in turn embeds a binary `glTF`. Nevertheless, the problem is in the generation of the `glTF` itself, since no converter is available at the moment and even with the FME¹ software the output is invalid [Schilling et al., 2016].

- *Assign a slightly different colour to each parcel*
This solution would just be an “easy fix” to make the highlight work. On the other hand, the large variety of colours could cause one to lose overview of the different kinds of parcels, compromising readability of the viewer. For this reason, this has not been considered as a valid solution.

6.11 TOOLTIP FUNCTIONALITY

The tooltip functionality has already been introduced in [Section 4.1.2](#) as part of the list of requirements related to 3D cadastre visualization. The principle behind the tooltip functionality is similar to the one used for highlighting a feature, i.e. the feature needs to be “picked”.

¹ FME is an ETL (Extract, Transform, Load) software for conversion of data formats.

Cesium JS allows to pick one single KML feature showing the information related to that parcel on the viewer. The default tooltip functionality provided by Cesium JS allows the user to pick a feature, by showing a bright green highlight box indicating the selected parcel as shown in Figure 6.29.

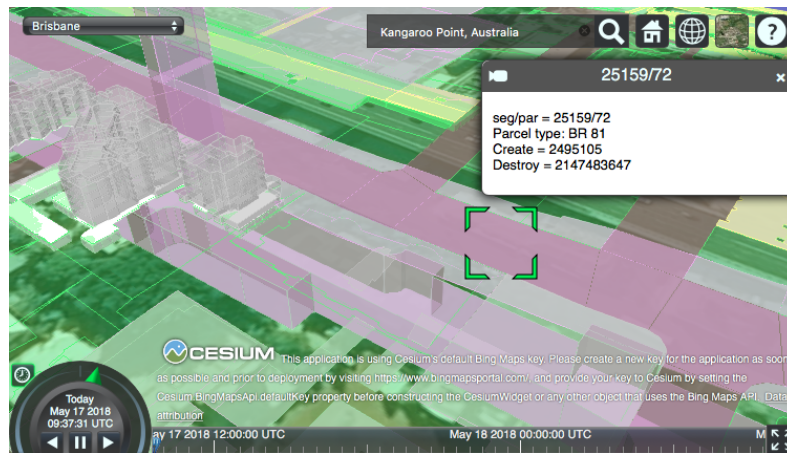
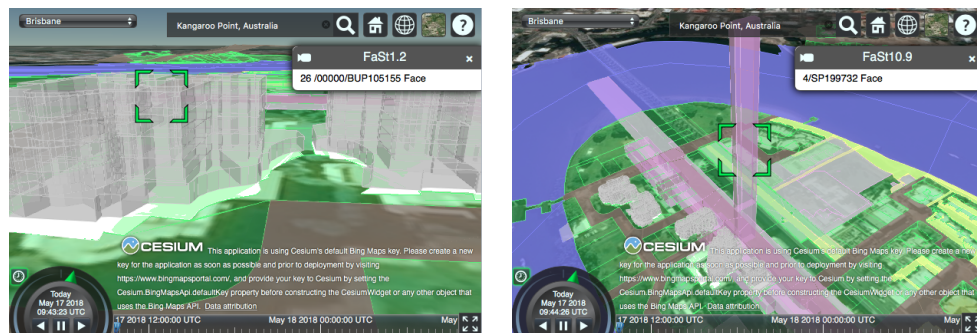


Figure 6.29: First implementation of the tooltip functionality using the default Cesium JS style.

At a first look, this might seem ideal, for the purpose of this research, except that, the shape and size of the highlight box are fixed and while clicking on a random point of a face the highlight box always aims at the center of the face. Figure 6.30 shows two examples in which this tool can be inconvenient in the context of 3D cadastre: the first situation shows an apartment building made of many units condensed together, where it is impossible to tell which is the selected parcel (see Figure 6.30a), the second situation shows a long and narrow parcel where the highlight box is aiming only at its center without enclosing the whole parcel, in this case it is still possible to understand which is the selected parcel but it is not ideal (see Figure 6.30b).



(a) Difficult to tell to which apartment unit the information refers to. (b) The highlight box only aims at the center of the parcel.

Figure 6.30: This implementation of the tooltip functionality can be inconvenient in some cases.

The possibility to dynamically modify the shape and size of the highlight box according to the shape and size of the parcels selected would help to identify the selected parcel. Since this is not possible, it has been decided to use the highlight functionality previously implemented. The result is shown in Figure 6.31. The selected parcel is highlighted in yellow and a pop-up window shows the information related to that parcel.



Figure 6.31: Second implementation of the tooltip functionality using the highlight feature functionality.

6.1.1.1 Integration of fake RRRs and persons/parties in the database

Due to privacy concerns, RRRs and persons/parties linked to the parcels were not available, therefore a dummy dataset has been created by Dr. Rodney J. Thompson to simulate a reality-like situation.

Two “views” have been included into the PostgreSQL/PostGIS database to assign owners and ownership rights to the 2D and 3D cadastral parcels (the definition of the views is included in Appendix A).

It must be noted that the names of the owners do not correspond to physical persons, but they have been invented for the purpose of the application. In addition, the relationship between the owners and the parcels is generated randomly. For example, if in reality a person is the owner of three units in an apartment building, it is not guaranteed that those three buildings will also have one unique owner in the fake dataset.

6.1.1.2 Retrieval of administrative information from the database

In order to retrieve the administrative information from the DB, a data server (in this case, GeoServer) has been put in place by Drs. Marian de Vries. A WFS request is sent to GeoServer when the user presses the “Get administrative info” button. While the response is converted into a table, as shown in Figure 6.32. In particular, the “partyname” field contains the name of the owner of the parcel selected.

objectid	segment_nr	parcel_nr	partyname	creating_lock_nr	destroying_lock
4679900	25159	262	Clem 7	2495105	2147483646

Figure 6.32: Example of WFS response converted into a table, as implemented in the prototype.

The raw [WFS](#) request to get the owner (and some additional information) when the user clicks on a parcel looks like the following:

```
http://pakhuis.tudelft.nl:8088/geoserver/geoweb/ows?service=WFS&version=1.1.0&request=GetFeature&outputFormat=application/json&typeName=geoweb:v_owner_of_parcel&cql_filter=segment_nr=25159andparcel_nr=262
```

The raw response is shown in [Figure 6.33](#).

```

1  {
2    "type": "FeatureCollection",
3    "totalFeatures": 1,
4    "features": [{
5      "type": "Feature",
6      "id": "v_owner_of_parcel.fid--56af4823_16468417e2c_-1b3b",
7      "geometry": null,
8      "properties": {
9        "objectid": 4679900,
10       "segment_nr": 25159,
11       "parcel_nr": 262,
12       "partyname": "Clem 7",
13       "creating_lock_nr": 2495105,
14       "destroying_lock_nr": 2147483646
15     }
16   }],
17   "crs": null
18 }

```

Figure 6.33: WFS response to get the owner of the selected parcel (and some additional information).

6.12 OBJECT SEARCH ON ADMINISTRATIVE DATA

The object search functionality has been introduced in [Section 4.1.6](#). Within the 3D cadastre prototype, this tool has been implemented on administrative data, and in particular on ownership of persons/parties. The search can be performed by inserting an owner name in the search box at the top left corner of [Figure 6.34](#), then by clicking on the “Start search” button a [WFS](#) request is sent and all the parcels owned by “Clem 7” will be highlighted in red. The [WFS](#) request to get all parcels owned by a certain person/party (in this case the party is “Clem 7”):

```
http://pakhuis.tudelft.nl:8088/geoserver/geoweb/ows?service=WFS&version=1.1.0&request=GetFeature&outputFormat=application/json&typeName=geoweb:v_parcel_owned_by&propertyName=objectid,segment_nr,parcel_nr&cql_filter=partynamelike' %Clem7%'
```

Note that, when inserting an invalid owner, the system will not highlight any parcel, while to clear the selection, it is necessary to empty the search box and press the “Start search” button again.

The raw [WFS](#) response is encoded in [JSON](#) format, as shown in [Figure 6.35](#). Note that not all the attributes are requested, but only the ones needed to pick the right object in the scene:

- The object id
- The segment number
- The parcel number

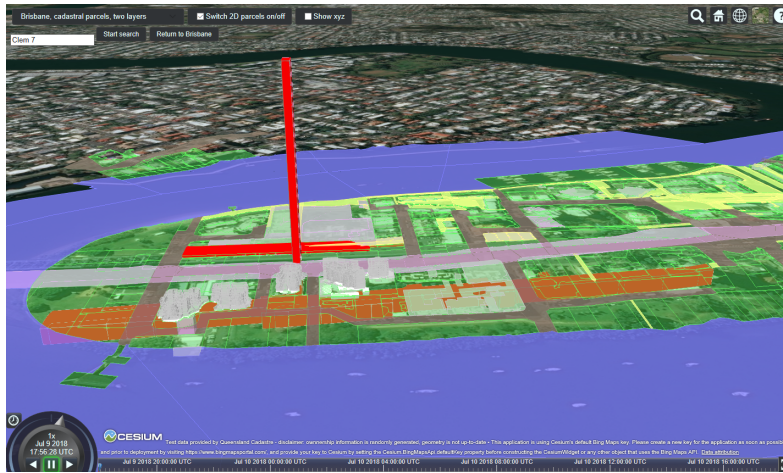


Figure 6.34: Example of object search carried out on administrative data. The owner "Clem 7" has been entered in the search box and all the parcels owned by Clem 7 are now highlighted in red. Note that also the parcels underground are highlighted.

```

1  {
2    "type": "FeatureCollection",
3    "totalFeatures": 19,
4    "features": [{
5      "type": "Feature",
6      "id": "v_parcel_owned_by.fid--56af4823_16468417e2c_-1b3a",
7      "geometry": null,
8      "properties": {
9        "objectid": 4679767,
10       "segment_nr": 25159,
11       "parcel_nr": 210
12     }
13   },
14   ...
15   ...
16   {
17     "type": "Feature",
18     "id": "v_parcel_owned_by.fid--56af4823_16468417e2c_-1b28",
19     "geometry": null,
20     "properties": {
21       "objectid": 4679900,
22       "segment_nr": 25159,
23       "parcel_nr": 262
24     }
25   },
26   "crs": null
27 }

```

Figure 6.35: WFS response to get all parcels owned by a certain person/party.

6.13 SUMMARY OF THE REQUIREMENTS INCLUDED IN THE PROTOTYPE

Within this chapter the implementation phase has been described, paying particular attention to the issues encountered, explaining how they have been solved when possible.

In general the 3D cadastre prototype includes the following requirements related to 3D visualization:

- *Navigation tools and view controls.*

- *Tooltip*, which shows information about the parcel and the administrative data related to it.
- *Integration of topography* (i.e. a [DTM](#)).
- *Transparency*.
- *Object selection and highlight*.
- *Object search*, to check which parcels are owned by a certain person.
- *Dynamic elevation tool* to solve the problem of subsurface visualization.
- *Camera start-up position* implemented to start up the viewer at the right location.

Note that the last two items in the list were not included among the initial requirements in [Section 4.1](#), but they have been added to overcome some technology issues and to increase the user-friendliness of the prototype.

Furthermore, the following requirements related to the web viewer are also included in the 3D cadastre prototype:

- *Platform and browser independence* thanks to the [WebGL](#) technology.
- *Layers control*, although it is not possible for the user to add his own layers or modify the existing ones (i.e. by changing the level of transparency). On the other hand, the users can toggle on and off the 2D parcels according to the information that they want to visualize.
- *Support for topographic visualization* from a technological point of view.
- *Support for geo-referencing* thanks to the virtual globe technology.
- *Underground visualization* is now possible as a result of the vertical shifting of the parcels.
- *Open source platform*.
- *Possibility for the platform to be extended*.

The last requirement on the list is probably the most important one, since it made possible the implementation of all the others by changing the JavaScript source code of Cesium JS and add new functionalities to the already existing ones (i.e. feature highlight and layer control among others).

7

USABILITY TEST

This chapter contains the last phase of this MSc research: the usability test that will be carried out by potential users of the 3D cadastre prototype.

The chapter is divided into two main sections. [Section 7.1](#) introduces the concepts of usability and usability testing in this particular context. [Section 7.2](#) explains the four phases to conduct a usability test: (1) recruitment of the potential users, (2) definition of the main goals and tasks to be carried out, (3) creation of a questionnaire and (4) results processing to extract general feedback.

7.1 ABOUT USABILITY AND USABILITY TESTING

In order to carry out usability tests it is important to give a definition to the term "usability" in this particular context.

According to the [ISO 9241-11](#) usability is: "*the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use*". Thus, usability is about effectiveness, in other words, whether users can complete tasks and achieve their goals. Usability is also about efficiency, which is often measured in time and effort for the user to complete his tasks. Finally, usability is connected with user's satisfaction and his opinion about the product/prototype [[ISO, 2018](#)].

The absence of frustration while using something is what makes it usable. For this reason, usability is considered an issue only when it is lacking or absent.

Usability testing is considered a research tool, a fundamental evaluation method which enables to assess a product or, in this case, a prototype. This process uses people as testing participants who are sampling the user groups to which the prototype is addressed [[Rubin and Chisnell, 2008](#)]. Although usability testing is a crucial phase for the assessment and further development of the 3D cadastre prototype, some limitations of this approach must be pointed out:

- *Testing is an artificial situation.* In other words, it depicts an actual situation of usage, but it is not the situation itself. Therefore, in some cases this can affect the results.
- *The results of the testing do not prove that a product works.* The results are dependent on the way in which the test is conducted and it is not guaranteed that they are reliable.
- *The participants are not always fully representative of the actual user groups.* The actual end users of the prototype can be hard to fully identify, thus the completeness of the representatives depends on your ability to identify the target audience.
- *Testing through a questionnaire with sample users is not always the best method to assess usability.* In some situations, other techniques are preferred in terms of costs, time and accuracy. An example is a method called "expert review", an heuristic evaluation made by an expert in the field, called to assess the prototype.

In spite of these limitations, usability testing is still considered a valid method to evaluate the usability of a product and to spot its weaknesses [Rubin and Chisnell, 2008]. Through usability testing, developers can have direct information about the usability and user friendliness of their applications and an insight into the possible issues encountered by the testers [Ivory, 2003].

7.2 FOUR STEPS TO CONDUCT USABILITY TESTING

The testing process can be subdivided into four main phases:

1. Recruiting users
2. Define goals and tasks to be performed
3. Create a questionnaire
4. Process results and obtain feedback

The phases need to be executed in the given order, so only when the previous phase is completed we can proceed to the next one, and so on.

7.2.1 Step 1: Recruiting users

The first phase of usability testing involves the definition of the potential user groups of the 3D cadastre prototype. Finding real users can be difficult and time consuming, so it is better to start this phase as soon as possible in the development process. It might seem logic that the bigger is the number of users willing to participate in the testing, the more accurate the results will be. However, Nielsen [2000] demonstrated that five users are enough to spot about 80% of the potential issues. He calculated the number of problems found in prototypes according to the equation:

$$N(1 - (1 - L)^n)$$

where n is the number of users, N is the number of usability problems and L is the proportion of usability problems determined from the testing of a single user. The typical value of L is given and it is about 31%, averaged from a substantial number of projects studied. Figure 7.1 shows the plot of the curve for $L = 31\%$.

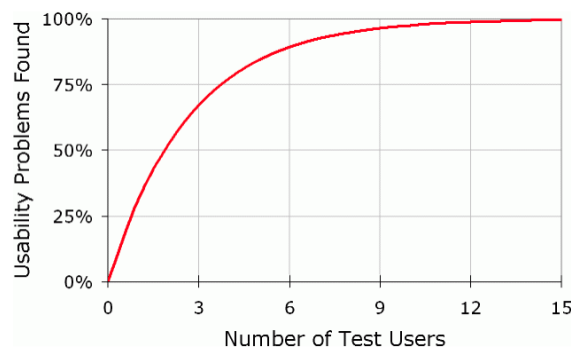


Figure 7.1: Percentage of usability problems found according to the number of test users. Plot of the curve for $L = 31\%$, where L is the proportion of usability problems determined from the testing of a single user [Nielsen, 2000].

In short, each additional test person provides a diminishing added value. This is due to the fact that different users will observe the same thing multiple times generating a bunch of redundant information and very little new insights. No more than 15 test users are needed to discover 100% of the issues. Only in case the prototype

is addressed to highly different groups of users, then it is recommended to take a bigger sample of test users [Nielsen, 2000].

In the context of the 3D cadastre prototype, the potential users can be condensed into five different groups, in particular:

- Researchers in the field of 2D and 3D cadastre (including staff of the MSc Geomatics for the Built Environment of TU Delft);
- Professionals using cadastral applications (lawyers, notaries, engineers, architects, land surveyors, building managers, etc.);
- Managers in the government and municipal authorities in charge of the maintenance of the cadastral system;
- Public and private entities, companies;
- Students of Geomatics for the Built Environment at TU Delft;
- General public, citizens.

Note that, this is an non-exhaustive list and new end users can pop up at any point in time, being this a rather important innovation in the field of cadastre.

Additionally, some studies recommend to test the prototype on “extreme users”, which means on both regular and non-regular users. Regular users are professionals and people involved in the field of cadastre, while non-regular users are people that are not involved in the field and do not have any (or only little) knowledge of cadastral applications [Dam and Siang, 2018]. In the list above non-regular users are represented by the general public/citizens.

7.2.2 Step 2: Define goals and tasks to be performed

Once the users have been identified and selected, the next step is to define a clear goal that the users have to keep in mind while performing their tests.

The overall goal is to ensure the usability of the 3D cadastre prototype. More in detail, this means that the prototype *must*:

- be useful to the target users;
- help users to be effective and efficient in performing their tasks;
- be easy (or even satisfying) to learn/use;
- eliminate issues and frustration for the users [Rubin and Chisnell, 2008].

Said that, in order to obtain feedback, there is the need to specify clearly-defined tasks to be carried out by the users (on the basis of the main goal). Here is a list of tasks defined for the 3D cadastre prototype:

- Navigate through the viewer, pan, zoom and rotate view to get familiar with the controls;
- Toggle on and off the visibility of a layer;
- Visualize the underground parcels, i.e. zoom close enough to see the details and navigate around it to see the boundaries from every angle;
- Visualize information about a single parcel, i.e. ownership information, and unit/lot/plan number, etc.;
- Search for a single owner and visualize all the parcels owned by that person.

These tasks must be formulated in a clear and not misleading way, possibly accompanied by a short explanation, i.e. an example scenario, to give the user some context about the action that he/she is asked to perform [Nielsen, 2014].

7.2.3 Step 3: Create a questionnaire

The next phase is the formulation of a questionnaire to be handed over to the test users. Before starting, the users need to fill in their personal details and answer some multiple choice questions about the user group they belong to and their knowledge of Cesium JS.

The core of the questionnaire is composed of five main sections. Each section includes a description, a task, an opinion and a grade.

The description of the task to be performed gives context to the test user and explains why that functionality is crucial in a 3D cadastre application. After that, the user has to perform a specific task in order to answer a multiple choice question (in which four options are possible). Once the task is completed, the user can give an opinion which will contribute to the future improvement of the prototype. Finally, since extracting feedback in an objective way can be a rather difficult task, each section has to be graded by the user according to the usability of the functionality it relates to. The questionnaire has been created with Google Forms, an online tool to create surveys, questionnaires, etc.

An overview of the sections included in the questionnaire is given below, for a complete version, see [Appendix D](#):

Section #1

Description: Please make sure you have a working Internet connection. Open the following link on your web browser to start up the prototype: [3D Cadastre Prototype](#). Before testing more advanced functionalities, it is crucial to get familiar with the basic navigation tools and view controls.

Note: it is suggested to use a mouse.

Task: Navigate to the Brisbane Airport and check where it is located with respect to the river. You can do this in two ways, if you know where the airport is approximately located just pan and zoom to the location, otherwise click on the magnifier icon and type "Brisbane Airport, Australia".

Opinion: Can you easily navigate through the viewer? Are the controls intuitive?

Section #2

Description: Once you are familiar with the web viewer you can start playing with the data available by changing the visibility of the layers. The prototype contains layers with cadastral parcels both in 2D and 3D. Depending on the application, you may want to visualize certain layers rather than others.

Task: Which layers can be toggled on/off?

Opinion: Was it easy to understand which layers could be toggled on/off? If applicable, please explain the difficulties you encountered.

Section #3

Description: One of the advantages of a 3D cadastral system is the possibility to store and visualize 3D underground parcels. A limitation that many globe based web-viewers have is that the camera cannot navigate under the ground surface, making impossible the visualization of subsurface parcels. For this reason, the "dynamic elevation tool" has been implemented, so that the parcels can be shifted up of a defined amount to be able to see the ones under the earth surface.

Task: Suppose you want to take a look at the boundaries of the underground tunnel in our 3D Cadastre test area near the Brisbane city centre. In order to do that, you have to shift the ground surface and navigate around the parcels to visualise them in detail. What is the lowest z-value? (i.e. deepest point of the lowest 3D parcel below the surface)? Note: if you shifted the parcels you have to take in to account the amount of the shift.

Opinion: Were you able to navigate and see the details of the parcels from every angle? Could you see the coordinates on the screen?

Section #4

Description: Cadastre is not only about volumes and parcel boundaries, but it is also about legal information. Every parcel has attached legal information about owner(s), ownership rights, and identification number.

Task: Who is/are the owner(s) of the underground 3D tunnel parcels?

Opinion: Could you visualize the information easily? Was it easy to understand what to look for?

Section #5

Description: Cadastre is also about people and, in particular, owners. Even though the primary focus of the prototype is on visualization, it is important for the user to have access to the 2D and 3D parcels based on ownership of persons.

Task: Search for owner "Annie" (note fake names in database for privacy). Which is the highlight colour used to show the parcels owned by Annie?

Opinion: Was the search functionality intuitive?

Finally, an additional section allows the users to suggest ways of improving the design and point out what they did like/dislike about the existing functionalities of the prototype.

Additional remarks

What is the issue that mostly limits the user-friendliness and usability of the 3D cadastre prototype?

What functionality did you like best? What do you think is the most useful functionality?

Please choose 3 functionalities that, according to you, are crucial for the improvement of the 3D cadastre prototype.

Here the user can choose among the items of the two lists of requirements or add his own suggestions.

7.2.4 Step 4: Process results and obtain feedback

The final phase of the usability test consists of processing the results obtained and extracting general feedback for the future improvement of the 3D cadastre prototype.

A considerable amount of test users, 20 in total, replied to the questionnaire. As shown in the pie chart in [Figure 7.2](#), the most numerous user group is composed of the students of the MSc Geomatics for the Built Environment at TU Delft (40%), followed by the researchers in the fields of 2D or 3D cadastre, including the staff of the MSc Geomatics (30%) and the general public (10%). No professionals and managers in the government replied to the questionnaire.

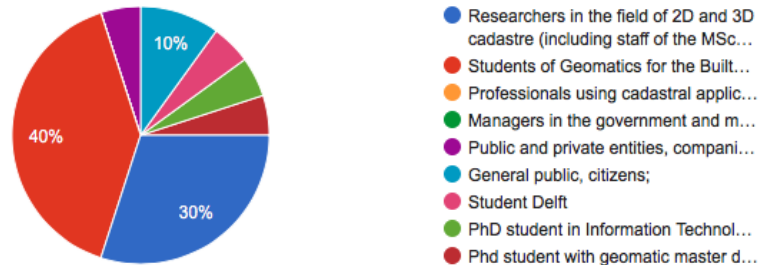


Figure 7.2: Pie chart showing the percentage of representatives of the users groups who carried out the test.

In addition, only about 10% of the users were familiar with Cesium JS, while the rest was not very familiar with the application or never used it before, as shown in [Figure 7.3](#)

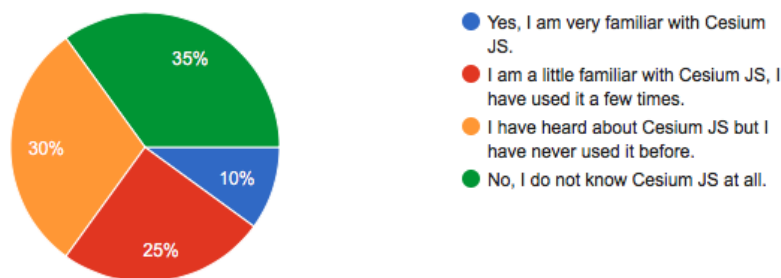


Figure 7.3: Pie chart showing the knowledge of Cesium JS by the different users.

An overview of the tasks is now given, highlighting the main feedback and the average score for each functionality.

Section #1:

Opinion: *Can you easily navigate through the viewer? Are the controls intuitive?*

70% of the test users stated that the controls are intuitive, which means that they found it easy to navigate through the viewer and perform the requested task. A 10% suggested to insert a North arrow in order to keep orientation after rotating the view. On the other hand, a 20% replied that the controls are not straightforward and it takes some time to get familiar with them.

The weighted average score for this functionality is: 7.7.

Section #2:

Was it easy to understand which layers could be toggled on/off? If applicable, please explain the difficulties you encountered.

It was intuitive to understand which layers could be toggled on/off for about half of the users, while a 10% stated that it was not clear at all. Including the possibility to toggle on/off the 3D parcels and make the label of the button more descriptive were the main suggestions for improvement. In addition, some users confused the drop-down menu used for the dynamic elevation tool with some kind of layers control functionality.

The weighted average score for this functionality is: 7.15.

Section #3:

Were you able to navigate and see the details of the parcels from every angle? Could you see the coordinates on the screen?

A considerable amount of users (65%) declared that they were able to visualize the underground parcels and the coordinates of the lowest parcel in the viewer. On the other hand, a quarter of the users could not find the way to properly visualize underground parcels or did not find it intuitive. The main criticisms were about the difficulty of visualizing the coordinates because they were hidden behind the volumes or the mouse hovering was not showing the values continuously.

The weighted average score for this functionality is: 6.2.

Section #4:

Could you visualize the information easily? Was it easy to understand what to look for?

This question refers to the retrieval of administrative information (RRRs and persons/parties - owners) of a selected parcel.

Three quarters of the total users responded that they found the information easily, while the rest could not achieve the task either because the WFS request gave an error or because they did not find it intuitive (i.e. the table showing administrative information was unclear).

The weighted average score for this functionality is: 7.5.

Section #5:

Was the search functionality intuitive?

Here the question refers to the object search carried out on administrative data.

The opinions of the users about this functionality are divided. The first group (55% of the total) carried out the task easily and gave positive feedback, while the second group (45% of the total) could not carry out the task, either due to a WFS request error, or because they did not realize that the parcels were highlighted. Accordingly, the main suggestions were about the implementation of a bounding box around the selected parcels to give a hint about their location, otherwise while looking from a distance, the user might not notice that the parcels are highlighted. In addition, some users could not reset the search or got confused by the two search boxes in the viewer.

The weighted average score for this functionality is: 6.2.

Additional remarks section:

What is the issue that mostly limits the user-friendliness and usability of the 3D cadastre prototype?

The main limitation for the user-friendliness of the 3D cadastre prototype was, according to the users, the dynamic elevation tool. Visualizing underground parcels was not so straightforward and some bugs were found in the "Show XYZ" functionality. Additionally, the search functionality was not so intuitive, because the viewer does not zoom to the selected parcels and the users are getting confused because of the presence two search boxes. Also, some remarks about speed were made, since the navigation/interaction was sometimes an obstacle due to the slow performance (i.e. during the load of the application and while clicking/highlighting some features).

What functionality did you like best? What do you think is the most useful functionality?

The most appreciated functionality was the possibility to get administrative information from a selected parcel, a very crucial functionality in cadastre applications. This was followed by the possibility to visualize the underground parcels (although some users did not like the actual implementation by means of a drop-down menu). Also, the search functionality was among the most useful ones according to a con-

siderable amount of users.

Please choose 3 functionalities that, according to you, are crucial for the improvement of the 3D cadastre prototype.

The top three functionalities that are crucial for the improvement of the prototype are:

- *Object search* which, in the description of the requirements, is intended as a search tool on either spatial and non spatial data and it can be based on address, geocode, owner's name (as the one implemented in the prototype), coordinates, etc.
- *3D measurement tools*, which allow to estimate the dimensions of the parcels by performing different measurements, such as, area, volume, distance between points, and so on.
- *2D overview map*, to provide a vertical view over the scene in order to help the user orientate through the viewer.

To sum up, the usability test has been carried out by both specialized and non specialized users, although two of the user groups mentioned in the list in [Section 7.2.1](#) are missing: professionals using cadastral applications, such as lawyers, notaries, engineers, architects, land surveyors, building managers, etc., and managers in the government and municipal authorities in charge of the maintenance of the cadastral system. For this reason, a second usability test, including some of the requested improvements, is scheduled in August 2018 and members of the International Federation of Surveyors (FIG) Working Group on 3D Cadastre will be invited to participate.

The less intuitive functionality (and most difficult to achieve) according to a large part of the test users was the object search functionality, which in most cases did not lead to any result. Therefore, for the future development of the prototype, this tool should be improved since it has also been indicated as the most crucial in the context of cadastre. The most appreciated feature was, instead, the possibility to retrieve administrative data from the DB by means of a WFS request. Accordingly, the vast majority of the users managed to perform this task correctly, giving positive feedback about the functionality. For every task the majority of the users answered the question correctly. In general, the users achieved correctly 4 out of 5 tasks with an average of 3.7 points, as shown in [Figure 7.4](#). The detailed results relative to each questions are included in [Appendix D](#).

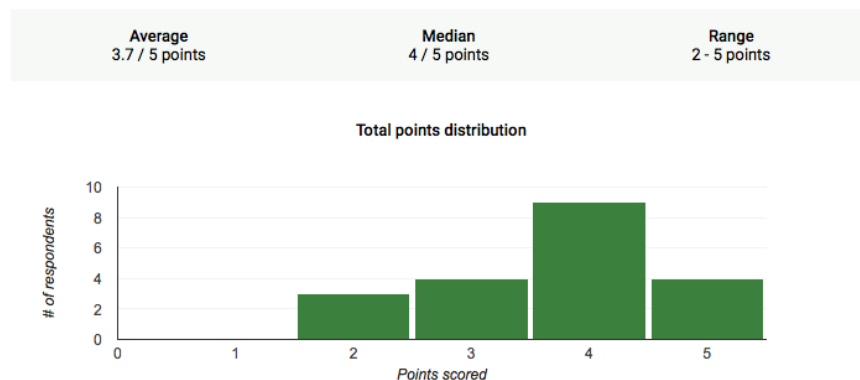


Figure 7.4: Bar chart showing the total points distribution.

This chapter concludes the research described in this document and summarizes the results achieved.

The purpose of this research is to overcome the issues of 3D cadastre visualization, such as occlusion, distortion, unbounded volumes and difficult perception of objects, through the implementation of a web-based prototype. The initial focus is on the requirements related to 3D visualization and on the ones related to the web-viewer. The choice of the most suitable WebGL based platform is a crucial phase to continue the research for this reason, the well-known (in the field of online visualization), Cesium JS web viewer, has been chosen. After that, the implementation of the requirements takes place with a number of shortcomings. Among them, we can mention the highlighting functionality, which is crucial for the user interaction, and the visualization of subsurface objects, a central issue in cadastral applications since today's cities are expanding in all directions, and in particular under the ground surface. Finally, in order to assess the effectiveness and reliability of the prototype a usability test has been carried out on potential users.

The chapter is subdivided in two main parts: [Section 8.1](#) which aims to give an answer to the research question and [Section 8.2](#) introducing ideas for future work.

8.1 CONCLUSIONS

The conclusion section is composed by the following subsections: [Section 8.1.1](#) answering the main research question and relative subquestions, [Section 8.1.2](#) explaining the contribution of this research in the field of Geomatics and [Section 8.1.3](#) including some reflections about the used methodology.

8.1.1 Research questions

In order to draw the conclusions of this research, the research question stated in [Section 1.3](#) must be answered. In doing so, the sub-questions will be answered first, followed by a comprehensive answer for the research question.

Note that, in this section, only a summary of the answer is given, for more detailed insights, the previous chapters should be used as a reference.

1. *How to tackle issues such as occlusion and ambiguous perception (in terms of position, size and shape) of objects while interacting with 3D Cadastre?*

One of the main issues in 3D cadastre visualization systems is occlusion management. This technique helps increasing spatial awareness and facilitating ordinary tasks, such as searching, navigating and understanding the 3-dimensional world.

Another issue is the position, size or shape ambiguity of an object as perceived by the user; namely, an object can be perceived differently according to the point of view from which we look at it. This research fixes these problems by using the following interaction techniques:

Navigation tools and view controls

The range of functionalities explained in [Section 4.1.1](#) such as zoom, pan, rotate and tilt view are crucial to provide basic interaction, allowing the

user to better understand the 3D world by looking at it from different angles. A static 3D image cannot be investigated in detail, while navigating into the 3D viewer can reveal relevant details about the position, size and shape of the 3D objects and (partially) eliminate the problem of occlusion.

Transparency

The different degree of transparency or opacity, as described in [Section 4.1.4](#), has been used to overcome occlusion issues since it allows to see through the boundaries of the 3D objects and investigate their interiors. Moreover, transparency has also been used for displaying the *DTM* so that it is possible to see the parcels underneath it.

Layers control

The possibility to toggle on/off the visibility of layers (as described in [Section 4.2.3](#)) allows the user to select the information needed, without having unnecessary objects on the screen. This helps diminishing the occlusion issues given by the high number of objects in the scene.

Underground view

Another source of occlusion is given by the earth surface, i.e. the virtual globe or the *DTM*, hiding the 3D parcels located underground. The ideal solution would be a 3D web visualization platform supporting subsurface visualization (see [Section 4.2.9](#)), however the selected technology (Cesium JS) still does not support this kind of functionality.

The dynamic elevation tool, described in [Section 6.9.2](#), is a workaround to fix the problem of subsurface visualization. The tool allows the user to vertically shift the parcels of a desired amount to be able to visualize the underground volumes. At the moment, the tool is implemented by means of a drop-down menu which allows the user to choose between ground level or 200 meters above ground. The future implementation foresees a slider thanks to which the user can drag the cursor (in the defined range) and change the height accordingly or a the possibility to interactively drag/elevate a set of parcels.

2. *What is the best way to combine the visualization of legal boundaries (cadastral parcels) and reference objects (topography)?*

The combination of administrative information and topographic information is a double-edged sword. On one hand, it is useful for orientation purposes, but on the other hand, it makes the visualization more confusing.

Moreover, when we observe the real world, only the physical objects are visible, but in the 3D cadastre web viewer we need to make a step forward and visualize, in a meaningful way, also what is invisible to our eyes. Different techniques have been adopted to achieve this goal:

Use of a virtual globe

The use of a virtual globe, such as Cesium JS (described in [Section 5.1.2](#)), contributes to give reference to the legal objects and to locate them in the correct location on the earth surface. This also allowed the implementation of a geocoder functionality, which points the camera to the city of Brisbane at the viewer's start up, so that the user does not need to do it manually.

Draping the parcels on the *DTM*

Another issue in the combination between physical and legal objects was the selection of a detailed *DTM* able to match the detail of the 2D and 3D parcels. Thus, it was decided to take the *DTM* as reference surface and drape the 2D parcels and the footprints of the 3D parcels on it. By doing this, the problem of the matching is solved, avoiding incompatibility between the parcels and the topographic surface.

Transparency

Once again, the use of transparency, as described in [Section 4.1.4](#), is of primary importance. In order to allow the user to see the orthoimage on the virtual globe and see through the volumes to analyze the boundaries, it has been decided to use transparency to represent both the [DTM](#) and the parcels, so that it is possible to see through them.

Interaction techniques

Finally, different kinds of interaction techniques have been adopted to make the combination of legal and physical objects more straightforward for the user. Layer control and navigation tools can help the user to distinguish between the two different types of objects through interaction.

3. *Which interaction techniques should be used in order to enhance user experience in 3D cadastre applications?*

The 3D cadastre prototype has been designed with the intent of being user friendly, functional and intuitive for the end users. The aim was to avoid frustration for the user in searching for the right tools by making the interface as intuitive as possible. In order to estimate the usability of the prototype, it was decided to carry out a usability test on sample users. The result of the testing showed that the interaction techniques approved/criticized by the test users are the following:

Functionalities thought to improve user-friendliness

The most appreciated functionalities were the following. First of all, the possibility to retrieve administrative information from the PostgreSQL/-PostGIS database, indeed the linking between physical and legal boundaries is of primary importance in a cadastral application. Second, the possibility to visualize underground parcels was very appreciated, probably also due to the fact that many 3D viewers do not allow to move the camera below the earth surface, making impossible the visualization of certain parcels. Finally, also the search functionality on administrative data has been considered useful to visualize the parcels owned by specific persons/parties.

Functionalities thought not to improve user-friendliness

Among the most criticized functionalities we can find the dynamic elevation tool, meant to visualize underground parcels. Although the subsurface visualization is one of the most appreciated features, its implementation by means of a drop-down menu has been considered an obstacle for the user-friendliness of the cadastre application. In the same way, the search functionality has been both appreciated and criticized by different users, due to the presence of some bugs which did not allow to visualize the selected parcels. Finally, an additional remark has been made about speed and performance of the application, although this was not in the main scope of the research.

4. *What are the options to store and transfer the data? (DBMS or encodings such as XML (KML), JSON, glTF, etc.)*

Many cadastral prototypes struggle with the issue of evaluating whether it is best to store the data in a [DB](#) or using different encodings, such as [KML](#), [JSON](#), [glTF](#) (and [B₃DM](#)), etc. In this research the possibilities are evaluated, either with the support of the related literature or (when possible) with experimentation during the development phase of the prototype.

Since the bespoke encoding software is creating [KML](#) output files from data stored in the PostgreSQL/PostGIS [DB](#), it was decided to experiment with that format first. This turned out to be a rather good option because it supports

3D data and it was possible to implement a considerable amount of requirements. The downside is that the [KML](#) support in Cesium JS is available only from version 1.7, therefore it is still rather limited. However, Cesium JS seems to be reliable in the long term due to its wide community of users and a very active forum, so certainly the support will increase with the release of the following versions. Despite that, the limited [KML](#) support was an obstacle for the implementation of some requirements, such as the feature selection/highlight functionality.

The [KML](#) issue pushed the research towards the exploration of other data formats, namely [JSON](#) and [gITF](#) (and [B3DM](#) which is based on [gITF](#)) (see [Section 6.6.2](#)). Both formats have advantages and disadvantages with regards to our purpose.

The [JSON](#) format is well supported in Cesium JS and examples of feature highlighting exist with 2D data. On the other hand, [JSON](#) does not support real 3D data but can only render extruded volumes, which are not suitable for 3D cadastre purposes.

The [gITF](#) (and [B3DM](#)) format is completely supported in Cesium JS, but there are no tools able to directly convert data into that format (within the set time frame). One may argue that the software FME has a function to do that, but the process drops the IDs of the objects, which are crucial in 3D cadastre applications.

Finally, the [DB](#) option have not been used in the context of this research, mainly because of the time consuming experimentation with [KML](#) and also due to the nature of the test area chosen.

The focus was more on the variety of data types (building and volumetric parcels, above ground and underground parcels, 2D and 3D volumes) rather than on scalability and storage of huge amounts of data. For this reason, it was decided to experiment primarily with data encodings and leave for future work the on-the-fly data retrieval from the database. The only experimentation with the [DB](#) was the retrieval of the administrative data ([RRRs](#) and persons/parties) from the PostgreSQL/PostGIS tables via a [WFS](#) request, which has been chosen for speed reasons.

To sum up, different options are available to store and transfer the data; [KML](#) is a rather good option but since the limited support can obstacle the implementation of additional requirements, it might be wise to switch to [gITF](#), and in particular [B3DM](#) which is very well supported in Cesium JS. On the other hand, this depends on the availability of data in this data format, or on the possibility to convert other formats into [gITF](#). However, in a real life situation a connection to the [DB](#) would be ideal to generate the volumes on the fly (i.e. in real time) through a client request.

5. *What is a good system architecture for 3D cadastre visualization?*

The fifth subquestion deals with the system architecture characteristics, and in particular communication protocols, encodings, shipping, caching and data storage. See [Figure 6.1](#) for an overview of the system architecture of the 3D cadastre prototype.

In the beginning, this has been formulated as a question to be investigated practically, nevertheless the time constraints allowed its investigation only from a theoretical point of view, as described in [Section 6.1](#). The following list investigates and compares the options available to determine the best system architecture components:

Shipping vs. online connection

Two options are possible when disseminating cadastral data: shipping of data through files or making the data available online (i.e. using a web

service).

In the past and still nowadays, many cadastral systems allow the users to obtain requested data only through shipping of files, mainly via e-mail. In this research, there was the assumption that, cadastral data was made available through an online connection, in order to facilitate the dissemination through the web and to have the data always up to date.

Data generation on the fly vs. storage in files

A further design option generates from the online connection to the database: whether is it better to transform the data on the fly (i.e. in real time) from the **DB** to the 3D model or to prepare it beforehand and store it in files on the server.

Unfortunately, this aspect could not be completely compared, but it is important to keep in mind that while working with files, a mechanism for the file update has to be put in place in order to have always reliable data. On the other hand, generating on the fly data could create performance related issues while dealing with large amounts of data.

Caching

What follows from the answer to subquestion n.4 is that, in the context of this research, the caching and tiling aspects are not needed, due to the nature of the dataset: the amount of data is not big enough to require tiling and caching, and all the parcels can be visualized at once without any problems of performance.

Nevertheless, looking towards the future of the prototype, the need of integrating a larger amount of data will make tiling and caching techniques essential to maintain a high level performance and a large amount of users.

Communication protocols

This aspect is again relating to the vision for the 3D cadastre prototype in the future, assuming the implementation of an on-demand selection of the data. At the actual state, the **KML** files are loaded in the viewer at the server side and the user cannot request customized data, thus a possible improvement for the future could be the support of a **WFS** to request data on demand. At the moment a **WFS** is in place to retrieve administrative information from the **DB**.

However, in case the prototype would be implemented using **gITF** data, the **WFS** protocol would not be suitable.

In short, an optimal system architecture should be capable of disseminating data through the web (i.e. through an online connection), making the information readily available for everyone without a formal request by the user. Then, it should allow the generation of data in real time according to the requests of the client, making sure that the performance is not compromised by the creation of the data on the fly. In addition, caching and tiling mechanisms should be put in place in case the dataset is considerably large and cannot be loaded all at once. Finally, the possibility to request data via a **WFS** would allow the user to obtain customized data.

Having answered the subquestions, now the main research question can be answered too.

What is a good system design to obtain a proper, clear and not misleading visualization of 3D cadastral parcels through a geo-web viewer?

To answer this question, in order to build a 3D cadastre visualization system, the advice deriving from this research is the following.

A good system design to obtain a proper, clear and not misleading visualization of 3D cadastral parcels should consider the implementation of most (or even all) of

the requirements listed in [Chapter 4](#), should be based on a good storage schema developed following the standards as the one described in [Section 6.3](#) and should be considered user-friendly by the majority of its potential users. In addition, it should be able to solve the most common visualization issues but, at the same time, the importance of the administrative data should not be put in second place. Moreover, it should be able to combine legal and physical objects in a meaningful way, also by combining the parcels with a *DTM*, following the requirements set in [Section 6.5](#). Finally, it should promote the efficient storage of the information, by means of a *DB*, depending on the needs and the size of the dataset.

Hopefully, this research contains at least part of these characteristics and can represent a starting point for further improvement of 3D cadastre visualization and dissemination.

8.1.2 Contribution in the field of Geomatics

This MSc research investigates the current bottlenecks of 3D cadastre visualization by developing a web-based prototype for the visualization of mixed 2D and 3D parcels, which solves (some of) the issues of previous applications.

The prototype provides a plug-in free interface, it allows the visualization of mixed 2D and 3D parcels in the same scene and it combines together legal and physical objects; these three functionalities were not available in most of the existing prototypes, an example is the Prototype for the Russian Federation (described in [Section 3.1](#)).

Additionally, the prototype is addressed to the Queensland cadastre, which contributed to provide the data of the Brisbane city centre. This makes the application a concrete solution that can be used in the real world, or even as a starting point for the development of the 3D cadastre web-viewer of the State of Queensland, Australia.

8.1.3 Reflection and discussion

Within this section, an analysis of the “big picture” related to the 3D cadastre visualization research is carried out. The goal is to present an in-depth reflection on what has been achieved successfully and on what are the limitations of the current status of the prototype and of the related MSc research.

Achievements

This research achieved several goals. First of all, a big achievement is the use of a plug-in free interface, due to the [WebGL](#) technology on which Cesium JS is based upon. Furthermore, the reduction of the occlusion issues by means of transparency and interaction techniques, which allows the user to navigate around the parcels in order to achieve a more familiar view of the world. Finally, the difficult perception of position, size and shape of objects is partially solved thanks to interaction techniques, an example is the implementation of the dynamic elevation tool (as described in [Section 6.9.2](#)) for a proper visualization of underground parcels.

In addition, the creation of a working prototype supported the entire theoretical research, from the definition of the requirements to the usability testing.

The prototype is primarily suitable for the Queensland cadastre, but it could be used by other cadastral systems after some modifications. This is possible mainly thanks to the [LADM](#) compliant data model.

As a result, this research marks a starting point for further development in the field of 3D cadastre visualization. A research subject for the future could be the further refinement of the prototype, starting from the implementation of the suggestions listed in [Future work](#).

Limitations

On the other hand, some criticism about the research needs to be pointed out. The initial ambitions had to be redefined according to the shortcomings that popped out during the implementation of the prototype and the time schedule set for the project.

The first criticism regards the *status of the 3D cadastre prototype*, which, at the moment, is still incomplete, both functionality wise and data wise (because it only contains a subset of the actual 3D parcels). Indeed, since the beginning it has been stated that the prototype was not going to be a finished product, ready to be used by the Queensland cadastre as operational system, but rather, a starting point to overcome the challenges of visualization in the field of 3D cadastre. Accordingly, it was not possible to support the whole list of requirements contained in [Chapter 4](#) but only a portion of them, the remaining ones are material for future work (see [Section 8.2](#)).

The second criticism concerns the *programming language and code quality*. The JavaScript programming language has been used in the wider [WebGL](#) framework embedded in Cesium JS. It must be pointed out that, the quality of the code is not guaranteed to be very efficient in its actual state, since it has been implemented starting from existing code examples in the Cesium JS library and adapted to the current needs. Moreover, most of the functions, together with their relative members and methods, are part of the Cesium JS library and they are explained in the official documentation.

The third criticism refers to the *platform used for the implementation of the prototype*. As already mentioned, the partial support for [KML](#) in Cesium JS was one of the main shortcomings in the implementation of the prototype, and knowing that beforehand could have oriented the research in different directions, such as:

- *Selecting another platform* for the development of the prototype, although the preliminary overview and testing of the platforms drove the choice on Cesium JS for different reasons described in [Section 5.3](#).
- *Using a different data encoding* as it has been discussed in [Section 6.6](#). Even though right now [KML](#) seems the best (temporary) solution, a potential candidate is [B3DM](#) which is very well supported in Cesium JS.
- *Implementation of a custom made software*, albeit not in the scope of the research. The prototype has been built on a generic platform; the term “generic” refers to a platform that can be flexibly used in different domains without a lot of modifications. Cesium JS is focusing primarily on simple visualization, but it requires a substantial effort for implementing advanced interaction tools and for linking the geometry with the semantic, i.e. administrative information. For this reason, it is difficult to achieve good results with such a platform. Being 3D cadastre an extremely specific domain, it would be worth considering, in a future research, the implementation of a custom made software for its proper visualization, interaction and dissemination.

8.2 FUTURE WORK

This section introduces some suggestions for future work which were not implemented, due to time or technology limitations. The aim of the future work is to arise some lines of research that could be pursued as an extension of this study. The order of the items suggests their importance; in other words, the first items are the ones with the highest priority.

8.2.1 Refinement of the dynamic elevation tool

The first improvement that can be done on the prototype is the implementation of the dynamic elevation tool, either by means of a slider, or by dragging/shifting a set of parcels. This provides the possibility to vertically slide the geometries in order to solve the underground parcels visualization problem.

As already mentioned, at the time of writing, the dynamic elevation tool has been implemented by means of a drop down menu containing two different layers: one at ground level and the other one at a height of 200 meters.

The use of a slider would make the tool more interactive and would help to maintain the orientation of the user in the 3D world. The maximum height value should be chosen in order to allow navigating around and below the parcels and visualizing the details from every angle.

The implementation of a slider/dragging functionality is surely possible in Cesium JS, the only concern is about the support for [KML](#) data. The slider should be able to modify the Z value of the parcels in order to slide them in a vertical way, even though at the moment, no evidence of this possibility has been found in the existing examples.

8.2.2 Refinement of the highlight functionality

At the moment, the highlight functionality is not working perfectly, which means that when selecting a 3D parcel, in some cases a 2D parcel placed behind it gets highlighted unintentionally.

The highlight functionality is helpful to distinguish the selected parcel from the surrounding ones, hence eliminating confusion. The main shortcoming here is the structure of the [KML](#) file which casts a ray to the location of the mouse click. For this reason, some parcels get highlighted accidentally by the user.

In order to fix this problem, either the Cesium JS support for [KML](#) should improve or another data format should be used. A good candidate for this purpose would be [B3DM](#), which is not based on ray casting but on vertices coordinates.

In addition, the support for the selection of multiple parcels at the same time, would allow the creation of an active set of features, which can then be further processed.

8.2.3 Refinement of the object selection concept: the active set

The object selection tool is directly connected to the above mentioned highlighting tool. At the moment, it is only possible to select an object and display the information about a single parcel on the screen, i.e. a sort of tooltip functionality.

The next step would be the dynamic selection of objects, for example the possibility to select (and deselect) one or more objects at a time, and the creation of an *active set* of objects that can be used for further analysis.

The selection of an active set is a rather useful/intuitive concept in cadastral applications and it can be implemented in different ways. For instance, the selection could be based on spatial and/or administrative data, and it could be carried out manually (by mouse click) or on an ownership basis, which means that by searching for a person's name, all the parcels owned by that person get selected. Other criteria could be also used to perform selection on administrative data, such as year of acquisition, total volume, private/public property, type of 3D parcel (volumetric or building format) and so on. Moreover, the active set could be used to store the selection for further processing. Different options are possible such as, selecting the neighbors of the active set (either in 2D or 3D), specify a 2D or 3D buffer, query the rights of the active set, combine the active set with the sliding tool (described in [Section 4.1.9](#)) in order to move the parcels horizontally or vertically, etc [[van Oosterom et al., 2000](#)].

8.2.4 Integration of administrative data (RRRs and persons/parties)

Although, in this research the focus is on visualization, administrative data, such as [RRRs](#) and persons/parties involved in ownership of parcels, is a crucial component of a reliable cadastral application. Therefore, the combination of administrative and spatial data, together with their topological relationships, would make the research more complete.

For the purpose of this thesis, it was not possible to obtain and make available real administrative data from the Queensland [DCDB](#) due to privacy concerns. Instead, a dummy dataset has been created in order to carry out some queries and make the prototype more reality-like. Ideally, the integration of administrative data would enable the users to carry out specific queries about ownership and legal rights. In addition, [LADM](#) data stored in PostgreSQL/PostGIS could be tested and then implemented in the prototype.

8.2.5 Towards XML structured 3D survey plans

The 3D survey plans of the Queensland [DCDB](#) are, at the present time, still on paper/[PDF](#)/[TIFF](#) format. This implies a non-trivial and time consuming operation to encode them into [XML\(KML\)](#) format, as explained in [Section 6.4](#).

Collecting and directly creating digital data formats (such as [KML](#) or other suitable formats) would fasten and automatize the encoding process of both building format and volumetric format survey plans, that in this research required a significant amount of work.

8.2.6 Formal usability test

The usability test carried out after the completion of the prototype is rather informal, mainly because of time constraints.

The limitations of the informal usability test are primarily connected with the selection of the user groups which, do not cover all the groups listed in [Section 7.2.1](#) and with the limited amount of time left to carry out the test.

A formal usability test foresees the recruitment of "extreme users" (described in [Section 7.2.1](#)) in order to have diverse test users and get insights from different perspectives. As an example, professionals in the field of 3D cadastre would be able to give more detailed feedback about the user friendliness of the prototype, on the other hand, citizens would spot more straightforward issues. The combination of expert (i.e. members of the [FIG Working Group on 3D cadastre](#)) and non-expert users would contribute to have a more comprehensive evaluation of the 3D cadastre system.

8.2.7 Virtual Reality and Augmented Reality

Augmented Reality ([AR](#)) and Virtual Reality ([VR](#)) are two techniques aimed at supporting and improving visualization and interaction with 3D geodata. [AR](#) allows the user to see the actual scene, together with 3D objects superimposed upon or combined with the real world. While, [VR](#) is a realistic representation of the world, including highly realistic details, sounds and movements.

3D cadastre applications could benefit from these relatively new technologies, in particular the 3D cadastre prototype could be modified to support [AR](#) or [VR](#) in order to improve the user experience and enhance the visualization techniques for representing physical and legal boundaries (including [RRRs](#)).

BIBLIOGRAPHY

- T. Aditya, F. Iswanto, A. Wirawan, and D. P. Laksono. 3D Cadastre Web Map: Prospects and Developments. In *2nd International Workshop on 3D Cadastres - Delft, the Netherlands*, pages 189–208, 2011. URL http://www.gdmc.nl/3dcadastre/literature/3Dcad_2011_22.pdf.
- Adobe Acrobat. Acrobat User Guide - Displaying 3D models in PDFs, 2018. URL <https://helpx.adobe.com/acrobat/using/displaying-3d-models-pdfs.html>.
- Analytical Graphics Inc. KML Support - issue#873, 2017. URL <https://github.com/AnalyticalGraphicsInc/cesium/issues/873>.
- B. Cemellini, R. Thompson, M. de Vries, and P. van Oosterom. Visualization/dissemination of 3D Cadastre. In *FIG Congress 2018 - Istanbul, Turkey*, 2018. URL http://fig.net/resources/proceedings/fig_proceedings/fig2018/papers/ts05c/TS05C_cemellini_rod_et_al_9591.pdf.
- Cesium. Cesium JS, 2018. URL <https://cesiumjs.org>.
- K. Chaturvedi. Web based 3D analysis and visualization using HTML5 and WebGL. Master's thesis, University of Twente - Faculty of Geo-information Science and Earth Observation - Department of Geoinformatics, Enschede, The Netherlands, 2014. URL https://www.iirs.gov.in/iirs/sites/default/files/StudentThesis/MSc_Thesis_KanishkChaturvedi.pdf.
- R. Dam and T. Siang. Test Your Prototypes: How to Gather Feedback and Maximise Learning, 2018. URL <https://www.interaction-design.org/literature/article/test-your-prototypes-how-to-gather-feedback-and-maximise-learning>.
- B. Danchilla. *Beginning WebGL for HTML5*. Apress, 2012. URL https://books.google.nl/books?hl=en&lr=&id=nFKRgF4swBMC&oi=fnd&pg=PR17&dq=Beginning+WebGL+for+HTML5&ots=JzZWGlms-G&sig=fWv-PiySel08Ey_DqmYGwr-jVrI#v=onepage&q=Beginning%20WebGL%20for%20HTML5&f=false.
- Department of Natural Resources and Mines (Queensland Government). Registrar of titles directions for the preparation of plans, 2013. URL https://www.dnrm.qld.gov.au/_data/assets/pdf_file/0011/97319/directions-for-the-preparation-of-plans.pdf.
- Department of Natural Resources and Mines (Queensland Government). Cadastral Survey Requirements, 2016. URL <https://www.dnrm.qld.gov.au/?a=105601>.
- G. Elizarova, S. Sapelnikov, N. Vandysheva, S. Pakhomov, P. van Oosterom, M. de Vries, J. Stoter, H. Ploeger, B. Spiering, R. Wouters, A. Hoogeveen, and V. Penkov. Russian-Dutch Project “3D Cadastre Modelling in Russia”. In *3rd International Workshop on 3D Cadastres: Developments and Practices - Shenzhen, China*, pages 87–102, 2012. URL <http://dare.ubvu.vu.nl/bitstream/handle/1871/39276/Russian-Dutch?sequence=1>.
- ESRI ArcGis. About CityEngine Web Viewer, 2017. URL <http://doc.arcgis.com/en/arcgis-online/reference/about-cityengine-web-viewer.htm>.
- Geobrowser 3D. Geobrowser 3D, 2017. URL <http://geobrowser3d.com>.

- Github. Github OSM buildings, 2017. URL <https://github.com/OSMBuildings/OSMBuildings>.
- Google Inc. Google Earth, 2018. URL <https://www.google.com/earth/resources/>.
- R. Guo, L. Li, S. Ying, P. Luo, B. He, and R. Jiang. Developing a 3D Cadastre for the Administration of Urban Land Use: A Case Study of Shenzhen, China. *Computers, Environment and Urban Systems*, pages 46–55, 2013. URL <https://pdfs.semanticscholar.org/fa27/0c45b4b39003913265e5e8ff7a1df713d416.pdf>.
- J. Henssen. Basic principles of the main cadastral systems in the world. In *FIG - Proceedings of the One Day Seminar held during the Annual Meeting of Commission 7 - Delft, The Netherlands*, 1995. URL https://www.fig.net/organisation/comm/7/activities/reports/events/Delft_seminar_95/paper2.html.
- A. Hevner and C. Samir. *Design Science Research in Information Systems*. Integrated Series in Information Systems. Springer, 2010. URL https://link.springer.com/chapter/10.1007%2F978-1-4419-5653-8_2.
- IGN. iTowns, 2015. URL <http://www.ign.fr/institut/innovation/itowns>.
- A. G. Inc. AGI, 2017. URL <https://www.agi.com/products/open-source>.
- ISO. ISO 9241-11 - Ergonomics of human-system interaction - Part 11: Usability: Definitions and concepts, 2018. URL <https://www.iso.org/obp/ui/#iso:std:iso:9241:-11:ed-2:v1:en>.
- ISO-TC211. ISO 19152, Geographic information — Land Administration Domain Model (LADM), 2012. URL <https://www.iso.org/standard/51206.html>.
- iTowns. iTowns, 2017. URL <http://www.itowns-project.org>.
- M. J. Ivory. Usability Testing Methods - Researchers' and Practitioners' Perspectives. In *Automated Web Site Evaluation*, chapter 3, pages 23–37. Springer, Dordrecht, 2003.
- E. Kalogianni. Linking the legal with the physical reality of 3D objects in the context of land administration domain model. Master's thesis, Delft University of Technology - Department of OTB, Delft, the Netherlands, 2016. URL <https://repository.tudelft.nl/islandora/object/uuid:2e3d7f53-9ca3-4003-b09b-77b82f312873?collection=education>.
- Khronos Group. WebGL overview, 2017. URL <https://www.khronos.org/webgl/>.
- A. Mackey. Pluralsight - Why you should learn WebGL, 2017. URL <https://www.pluralsight.com/blog/software-development/webgl-basics>.
- J. Nielsen. Why You Only Need to Test with 5 Users, 2000. URL <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>.
- J. Nielsen. Turn User Goals into Task Scenarios for Usability Testing, 2014. URL <https://www.nngroup.com/articles/task-scenarios-usability-testing/>.
- OGC. OGC KML 2.3, 2015. URL <http://docs.opengeospatial.org/is/12-007r2/12-007r2.html>.
- H. Olfat, D. Shojaei, and M. Briffa. The Victorian Digital Cadastre: Challenges and Investigations. In *Proc. of the 3rd Annual Conference of Research@Locate - Melbourne, Australia*, pages 47–52, 2016. URL <http://ceur-ws.org/Vol-1570/paper05.pdf>.

- OSGeo. GeoTIFF, 2018. URL <http://trac.osgeo.org/geotiff/>.
- OSM buildings. OSM buildings, 2017. URL <https://osmbuildings.org>.
- K. J. Pereira. Water Simulation on WebGL and Three.js. Master's thesis, The University of Southern Mississippi - Department of Computer Science, 2013. URL https://aquila.usm.edu/cgi/viewcontent.cgi?article=1151&context=honors_theses&sei-redir=1&referer=https%253A%252F%252Fscholar.google.nl%252Fscholar%253Fhl%253Den%2526as_sdt%253D0%25252C5%2526as_vis%253D1%2526q%253DWater%252BSimulation%252B%252BWebGL%252B%252BThree.js%2526btnG%253D#search=%22Water%20Simulation%20WebGL%20Three.js%22.
- J. Pouliot, F. Hubert, C. Wang, C. Ellul, and A. Rajabifard. 3D Cadastre Visualization: Recent Progress and Future Directions. In *5th International FIG 3D Cadastre Workshop - Athens, Greece*, pages 337–360, 2016. URL http://www.gdmc.nl/3DCadastres/literature/3Dcad_2016_28.pdf.
- J. Pouliot, C. Ellul, F. Hubert, C. Wang, A. Rajabifard, M. Kalantari, D. Shojaei, B. Atazadeh, P. van Oosterom, M. de Vries, and S. Ying. *Best Practices 3D Cadastres - Extended version*. International Federation of Surveyors (FIG), 2018. ISBN 978-87-92853-64-6. URL http://www.gdmc.nl/oosterom/FIG_3DCad.pdf.
- Queensland Government. Introduction to the Land and Water Allocation Registries D. o. N. R. a. Mines. 2009. URL https://www.dnrm.qld.gov.au/__data/assets/pdf_file/0006/97134/land-title-practice-manual.pdf.
- A. Ribeiro, J.-P. Duarte de Almeida, and C. Ellul. Exploring CityEngine as a Visualisation Tool for 3D Cadastre. In *4th International Workshop on FIG 3D Cadastres - Dubai, United Arab Emirates*, pages 197–218, 2014. URL https://estudogeral.sib.uc.pt/bitstream/10316/45895/1/Workshop2014_15.pdf.
- J. Rubin and D. Chisnell. *Handbook of usability testing - How to Plan, Design, and Conduct Effective Tests*. Wiley Publishing, Inc., 2008. URL <http://ccftp.scu.edu.cn:8090/Download/efa2417b-08ba-438a-b814-92db3dde0eb6.pdf>.
- A. Schilling, J. Bolling, and C. Nagel. Using glTF for streaming CityGML 3D City Models. In *Proceedings of the 21st International Conference on Web3D Technology - Anaheim, California*, pages 109–116, 2016. URL http://delivery.acm.org/10.1145/2950000/2945312/p109-schilling.pdf?ip=145.94.191.79&id=2945312&acc=ACTIVE%20SERVICE&key=0C390721DC3021FF%2E512956D6C5F075DE%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35&__acm__=1528827817_9d715f5e6866d824abe73867a03fbf28.
- D. Shojaei. *3D Cadastral Visualisation: Understanding Users' Requirements*. PhD thesis, University of Melbourne, Melbourne, Australia, 2014. URL <https://minerva-access.unimelb.edu.au/handle/11343/45129>.
- D. Shojaei, M. Kalantari, I. D. Bishop, A. Rajabifard, and A. Aien. Visualization requirements for 3D cadastral systems. *Computers, Environment and Urban Systems*, 41:39–54, 2013. URL https://ac.els-cdn.com/S0198971513000422/1-s2.0-S0198971513000422-main.pdf?_tid=21ad6972-2696-4b6d-b7aa-ca0166a49555&acdnat=1524667917_58e8a9ff87491fcb231cb4be0580169c.
- J. Stoter. *3D Cadastre*. PhD thesis, 2004. URL https://www.ncgeo.nl/index.php?option=com_k2&view=item&id=2563:pog-57-j-stoter-3d-cadastre&Itemid=350&lang=en.

- J. Stoter, H. Ploeger, and P. van Oosterom. 3D cadastre in the Netherlands: Developments and international applicability. *Computers, Environment and Urban Systems*, 40:56–67, 2012a. URL https://ac.els-cdn.com/S0198971512000841/1-s2.0-S0198971512000841-main.pdf?_tid=f3c573c6-681c-4aab-a4b9-7daaf198b7c9&acdnat=1524669442_0ef0f0a965d73b990166c928cc37061b.
- J. Stoter, P. Van Oosterom, and H. Ploeger. The Phased 3D Cadastre Implementation in the Netherlands. 2012b. URL <https://repository.tudelft.nl/islandora/object/uuid%3A27a647d2-bf62-4a39-bc4b-06dd3a3b3e07>.
- J. Stoter, H. Ploeger, R. Roes, E. van der Riet, F. Biljecki, H. Ledoux, D. Kok, and S. Kim. Registration of Multi-Level Property Rights in 3D in The Netherlands: Two Cases and Next Steps in Further Implementation. *ISPRS International Journal of Geo-Information*, 6, 2017. URL <http://www.mdpi.com/2220-9964/6/6/158/htm>.
- J. E. Stoter and P. van Oosterom. *3D Cadastre in an International Context: legal, organizational, and technological aspects*. CRC Press, 2006. URL <https://books.google.nl/books?hl=en&lr=&id=NNPLBQAAQBAJ&oi=fnd&pg=PP1&dq=3D+Cadastre+in+an+International+Context&ots=EiJLYhIK1t&sig=c6ntGKZR2HPQDC4FLV0w3eS0vcA#v=onepage&q=3D%20Cadastre%20in%20an%20International%20Context&f=false>.
- R. Thompson, P. van Oosterom, B. Cemellini, and M. de Vries. Developing an LADM compliant dissemination and visualization system for 3D Spatial Units. In *7th International FIG workshop on the Land Administration Domain Model - Zagreb, Croatia*, 2018.
- R. J. Thompson, P. J. M. Van Oosterom, S. Karki, and B. Cowie. A Taxonomy of Spatial Units in a Mixed 2D and 3D Cadastral Database. In *From the Wisdom of the Ages to the Challenges of the Modern World*. International Federation of Surveyors (FIG), 2015. URL <https://www.narcis.nl/publication/RecordID/oai:tudelft.nl:uuid%3A102fa9e3-5db3-4e31-961f-7dd277ba84d3>.
- UN-ECE. Land Administration Guidelines - With Special Reference to Countries in Transition. 1996. URL <http://www.unece.org/fileadmin/DAM/hlm/documents/Publications/land.administration.guidelines.e.pdf>.
- P. van Oosterom, B. Maessen, and W. Quak. Spatial, Thematic and Temporal Views. In *Proceedings 9th International Symposium on Spatial Data Handling - Beijing, China*, pages 8b.37–8b.52, 2000. URL http://www.gdmc.nl/publications/2000/Spatial_thematic_temporal_views.pdf.
- P. van Oosterom, B. Maessen, and W. Quak. Generic query tool for spatio-temporal data. *International Journal of Geographical Information Science*, 16(8):713–748, 2002. URL http://www.gdmc.nl/publications/2002/Generic_query_tool.pdf.
- N. Vandysheva, S. Sapelnikov, P. van Oosterom, M. de Vries, B. Spiering, R. Wouters, A. Hoogeveen, and V. Penkov. The 3D Cadastre Prototype and Pilot in the Russian Federation. *FIG Working Week*, 2012. URL http://www.fig.net/resources/proceedings/fig_proceedings/fig2012/papers/ts08h/TS08H_vandysheva_vanoosterom_et_al_6037.pdf.
- C. Wang, J. Pouliot, and F. Hubert. How users perceive transparency in the 3D visualization of cadastre: testing its usability in an online questionnaire. *Geoinformatica*, 21:599–618, 2016. URL <https://link.springer.com/article/10.1007%2Fs10707-016-0281-y>.
- WebGL Earth. WebGL Earth, 2017. URL <https://www.webglearth.com/#11=20.66614,24.55521;alt=16340753;h=43.173;t=-0.400>.

- S. Ying, R. Guo, W. Li, J. Yang, Z. Zhao, and L. Li. Visualization for the Coherent Set of 3D Property Units. In *5th International FIG 3D Cadastre Workshop - Athens, Greece*, pages 361–372, 2016. URL http://www.gdmc.nl/3DCadastres/literature/3Dcad_2016_29.pdf.
- A. Zare Zardini and F. Hakimpour. 3D Web Services for Visualization and Data Sharing in 3D Cadastre. *International Journal of 3-D Information Modeling (IJ3DIM)*, 2015. URL <https://www.igi-global.com/article/3d-web-services-for-visualization-and-data-sharing-in-3d-cadastre/154017>.
- J. R. Ziolkowska and R. Reyes. Geological and hydrological visualization models for Digital Earth representation. *Computers and Geosciences*, 94: 31–39, 2016. URL <https://www.sciencedirect.com/science/article/pii/S0098300416301509>.

A | TECHNICAL SCHEMA

The technical schema shows the Structured Query Language (SQL) load script to create tables and views in the PostgreSQL/PostGIS database.

The *cadviz3d* schema is part of the *geodata* database hosted on the *pakhuis* server of the OTB department at TU Delft. The DB can be accessed from pgAdmin. The conceptual schema, showing the Unified Modeling Language (UML) diagram is shown in [Figure 6.3](#) of [Section 6.3](#).

- The *parcel* table contains the 2D parcels of the Queensland DCDB, as shown in [Figure A.1](#). The table definition also includes the creation of three indexes: two non-spatial indexes (of which, one is unique) and a unique spatial index using *gist*.
- The *polyhedron_face* table contains the faces that compose the 3D parcels of the Queensland DCDB, as shown in [Figure A.2](#).
- The *qc_right* table and the *qc_party* table contains respectively the rights and parties of each parcel, as shown in [Figure A.3](#) and [Figure A.3](#).
- The *v_owner_of_parcel* and *v_parcel_owned_by* views, shown in [Figure A.5](#) and [Figure A.6](#), are “virtual tables” which allow to connect and retrieve the ownership information related to the 2D and 3D cadastral parcels.

```

1  -- Table: cadviz3d.parcel
2  -- DROP TABLE cadviz3d.parcel;
3  CREATE TABLE cadviz3d.parcel
4  (
5      objectid integer NOT NULL DEFAULT
6      nextval('parcel_objectid_seq'::regclass),
7      segment_nr integer NOT NULL,
8      parcel_nr smallint NOT NULL,
9      locality_id integer NOT NULL,
10     parish_id integer,
11     accuracy_id integer,
12     parcel_type_id character(1) COLLATE pg_catalog."default" NOT NULL,
13     coverage_id character(1) COLLATE pg_catalog."default" NOT NULL,
14     parcel_indicator_id smallint NOT NULL,
15     plan_nr character(12) COLLATE pg_catalog."default" NOT NULL,
16     lot_nr character(5) COLLATE pg_catalog."default" NOT NULL,
17     unit_nr character(5) COLLATE pg_catalog."default" NOT NULL,
18     unit_part character(5) COLLATE pg_catalog."default" NOT NULL,
19     primary_name_id integer NOT NULL,
20     centroid_lat double precision NOT NULL,
21     centroid_lon double precision NOT NULL,
22     centroid_z double precision NOT NULL,
23     calc_perimeter double precision,
24     calc_area double precision,
25     calc_volume double precision,
26     number_q integer NOT NULL,
27     linestyles text COLLATE pg_catalog."default" NOT NULL,
28     p_shape geometry(1096458),
29     creating_lock_nr integer NOT NULL,
30     destroying_lock_nr integer NOT NULL,
31     CONSTRAINT parcel_pkey1 PRIMARY KEY (objectid)
32         USING INDEX TABLESPACE users
33 )
34 WITH (
35     OIDS = FALSE
36 )
37 TABLESPACE users;
38
39 ALTER TABLE cadviz3d.parcel
40     OWNER to rod;
41
42 -- Index: parcel_lp1
43 -- DROP INDEX cadviz3d.parcel_lp1;
44 CREATE INDEX parcel_lp1
45     ON cadviz3d.parcel USING btree
46     (plan_nr COLLATE pg_catalog."default",
47      lot_nr COLLATE pg_catalog."default",
48      unit_nr COLLATE pg_catalog."default",
49      unit_part COLLATE pg_catalog."default")
50     TABLESPACE users;
51
52 -- Index: parcel_sp1
53 -- DROP INDEX cadviz3d.parcel_sp1;
54 CREATE UNIQUE INDEX parcel_sp1
55     ON cadviz3d.parcel USING btree
56     (segment_nr, parcel_nr, creating_lock_nr)
57     TABLESPACE users;
58
59 -- Index: parcel_x1
60 -- DROP INDEX cadviz3d.parcel_x1;
61 CREATE INDEX parcel_x1
62     ON cadviz3d.parcel USING gist
63     (p_shape)
64     TABLESPACE users;

```

Figure A.1: "Parcel" table definition with administrative and spatial indexes.

```

1  -- Table: cadviz3d.polyhedron_face
2  -- DROP TABLE cadviz3d.polyhedron_face;
3  CREATE TABLE cadviz3d.polyhedron_face
4  (
5      objectid integer NOT NULL DEFAULT
6      nextval('polyhedron_face_objectid_seq'::regclass),
7      segment_nr integer NOT NULL,
8      parcel_nr smallint NOT NULL,
9      face_nr integer NOT NULL,
10     face_name character varying COLLATE pg_catalog."default",
11     centroid_lat double precision,
12     centroid_lon double precision,
13     centroid_z double precision,
14     a_v double precision,
15     b_v double precision,
16     c_v double precision,
17     d_v double precision,
18     calc_perimeter double precision,
19     calc_area double precision,
20     linestyle character(1) COLLATE pg_catalog."default",
21     footprint_fl character(1) COLLATE pg_catalog."default",
22     p_shape geometry(1096458),
23     creating_lock_nr integer NOT NULL,
24     destroying_lock_nr integer NOT NULL,
25     CONSTRAINT polyhedron_face_pkey1 PRIMARY KEY (objectid)
26         USING INDEX TABLESPACE users
27 )
28 WITH (
29     OIDS = FALSE
30 )
31 TABLESPACE users;
32
33 ALTER TABLE cadviz3d.polyhedron_face
34     OWNER to rod;
35
36 -- Index: polyhedron_face_sp
37 -- DROP INDEX cadviz3d.polyhedron_face_sp;
38 CREATE UNIQUE INDEX polyhedron_face_sp
39     ON cadviz3d.polyhedron_face USING btree
40     (segment_nr, parcel_nr, face_nr, creating_lock_nr)
41     TABLESPACE users;
42
43 -- Index: polyhedron_face_x
44 -- DROP INDEX cadviz3d.polyhedron_face_x;
45 CREATE INDEX polyhedron_face_x
46     ON cadviz3d.polyhedron_face USING gist
47     (p_shape)
48     TABLESPACE users;

```

Figure A.2: "Polyhedron face" table definition with administrative and spatial indexes.

```

1  -- Table: cadviz3d.qc_right
2  -- DROP TABLE cadviz3d.qc_right;
3  CREATE TABLE cadviz3d.qc_right
4  (
5      rid_1 integer NOT NULL,
6      rid_2 integer NOT NULL,
7      partyid integer,
8      plan_nr character(12) COLLATE pg_catalog."default" NOT NULL,
9      lot_nr character(5) COLLATE pg_catalog."default" NOT NULL,
10     unit_nr character(5) COLLATE pg_catalog."default" NOT NULL,
11     description character varying COLLATE pg_catalog."default",
12     righttype character(1) COLLATE pg_catalog."default" NOT NULL,
13     creating_lock_nr integer NOT NULL,
14     destroying_lock_nr integer NOT NULL,
15     CONSTRAINT qc_right_pkey
16     PRIMARY KEY (rid_1, rid_2, creating_lock_nr, destroying_lock_nr)
17     USING INDEX TABLESPACE users
18 )
19 WITH (
20     OIDS = FALSE
21 )
22 TABLESPACE users;
23
24 ALTER TABLE cadviz3d.qc_right
25     OWNER to rod;
26
27 -- Index: qc_right_pix
28 -- DROP INDEX cadviz3d.qc_right_pix;
29 CREATE INDEX qc_right_pix
30     ON cadviz3d.qc_right USING btree
31     (partyid)
32     TABLESPACE users;
33
34 -- Index: qc_right_uix
35 -- DROP INDEX cadviz3d.qc_right_uix;
36 CREATE INDEX qc_right_uix
37     ON cadviz3d.qc_right USING btree
38     (plan_nr COLLATE pg_catalog."default",
39      lot_nr COLLATE pg_catalog."default",
40      unit_nr COLLATE pg_catalog."default")
41     TABLESPACE users;

```

Figure A.3: "qc right" table definition with administrative index.


```

1  -- Table: cadviz3d.qc_party
2  -- DROP TABLE cadviz3d.qc_party;
3  CREATE TABLE cadviz3d.qc_party
4  (
5      pid integer NOT NULL,
6      partyname character varying COLLATE pg_catalog."default",
7      partytype character(1) COLLATE pg_catalog."default" NOT NULL,
8      creating_lock_nr integer NOT NULL,
9      destroying_lock_nr integer NOT NULL,
10     CONSTRAINT qc_party_pkey PRIMARY KEY (pid, creating_lock_nr)
11     USING INDEX TABLESPACE users
12 )
13 WITH (
14     OIDS = FALSE
15 )
16 TABLESPACE users;
17
18 ALTER TABLE cadviz3d.qc_party
19     OWNER to rod;
20
21 -- Index: qc_party_nix
22 -- DROP INDEX cadviz3d.qc_party_nix;
23 CREATE INDEX qc_party_nix
24     ON cadviz3d.qc_party USING btree
25     (partyname COLLATE pg_catalog."default")
26     TABLESPACE users;

```

Figure A.4: "qc party" table definition with administrative index.

```

1  -- View: cadviz3d.v_owner_of_parcel
2  -- DROP VIEW cadviz3d.v_owner_of_parcel;
3  CREATE OR REPLACE VIEW cadviz3d.v_owner_of_parcel AS
4  SELECT p.objectid,
5         p.segment_nr,
6         p.parcel_nr,
7         i.partyname,
8         i.creating_lock_nr,
9         i.destroying_lock_nr
10 FROM parcel p,
11     qc_right r,
12     qc_party i
13 WHERE p.plan_nr = r.plan_nr AND p.lot_nr = r.lot_nr AND
14     p.unit_nr = r.unit_nr AND r.partyid = i.pid AND
15     i.creating_lock_nr < r.destroying_lock_nr AND
16     i.destroying_lock_nr > r.creating_lock_nr AND
17     r.creating_lock_nr < p.destroying_lock_nr AND
18     r.destroying_lock_nr > p.creating_lock_nr;
19
20 ALTER TABLE cadviz3d.v_owner_of_parcel
21     OWNER TO rod;

```

Figure A.5: "Owner of parcel" view definition

```
1  -- View: cadviz3d.v_parcel_owned_by
2  -- DROP VIEW cadviz3d.v_parcel_owned_by;
3  CREATE OR REPLACE VIEW cadviz3d.v_parcel_owned_by AS
4  SELECT p.objectid,
5         p.segment_nr,
6         p.parcel_nr,
7         p.plan_nr,
8         p.lot_nr,
9         p.unit_nr,
10        i.partyname,
11        p.p_shape
12  FROM parcel p,
13        qc_right r,
14        qc_party i
15  WHERE p.plan_nr = r.plan_nr AND p.lot_nr = r.lot_nr AND
16        p.unit_nr = r.unit_nr AND p.creating_lock_nr <= 9999999 AND
17        p.destroying_lock_nr > 9999999 AND r.creating_lock_nr <= 9999999 AND
18        r.destroying_lock_nr > 9999999 AND r.partyid = i.pid AND
19        i.creating_lock_nr <= 9999999 AND i.destroying_lock_nr > 9999999;
20
21 ALTER TABLE cadviz3d.v_parcel_owned_by
22        OWNER TO rod;
```

Figure A.6: "Parcel owned by" view definition

B | A VOLUMETRIC FORMAT SURVEY PLAN OF THE QUEENSLAND CADASTRE

The structure of a volumetric format plan of the Queensland [DCDB](#) is as follows:

- [Figure B.1](#) shows the cover page with all the administration information about the parcel such as, map reference, locality, local government, and so on.
- The first page ([Figure B.2](#)) shows how that parcel sits in relation to other neighboring parcels, although it does not really tell where it is located precisely. Two lots are shown: **1** and **822**, indicated in bold and dashed letters respectively. Bold letters indicate a 2D lot, while dashed letters indicate that a lot is 3-dimensional.

The plan number (in this case: *SP192737*) is indicated at the bottom of the page.

Each corner is connected by a straight line, the bearing and distance of the line are indicated in the plan itself. For instance, $88^{\circ}35'45''$ (93-99) indicates that bearing passing through the corners from 93 to 95 has an angle of $88^{\circ}35'45''$, and the distance between corners 93 and 7 is 4.882 meters.

- The second page ([Figure B.3](#)) contains a top view and an isometric view of the lot.

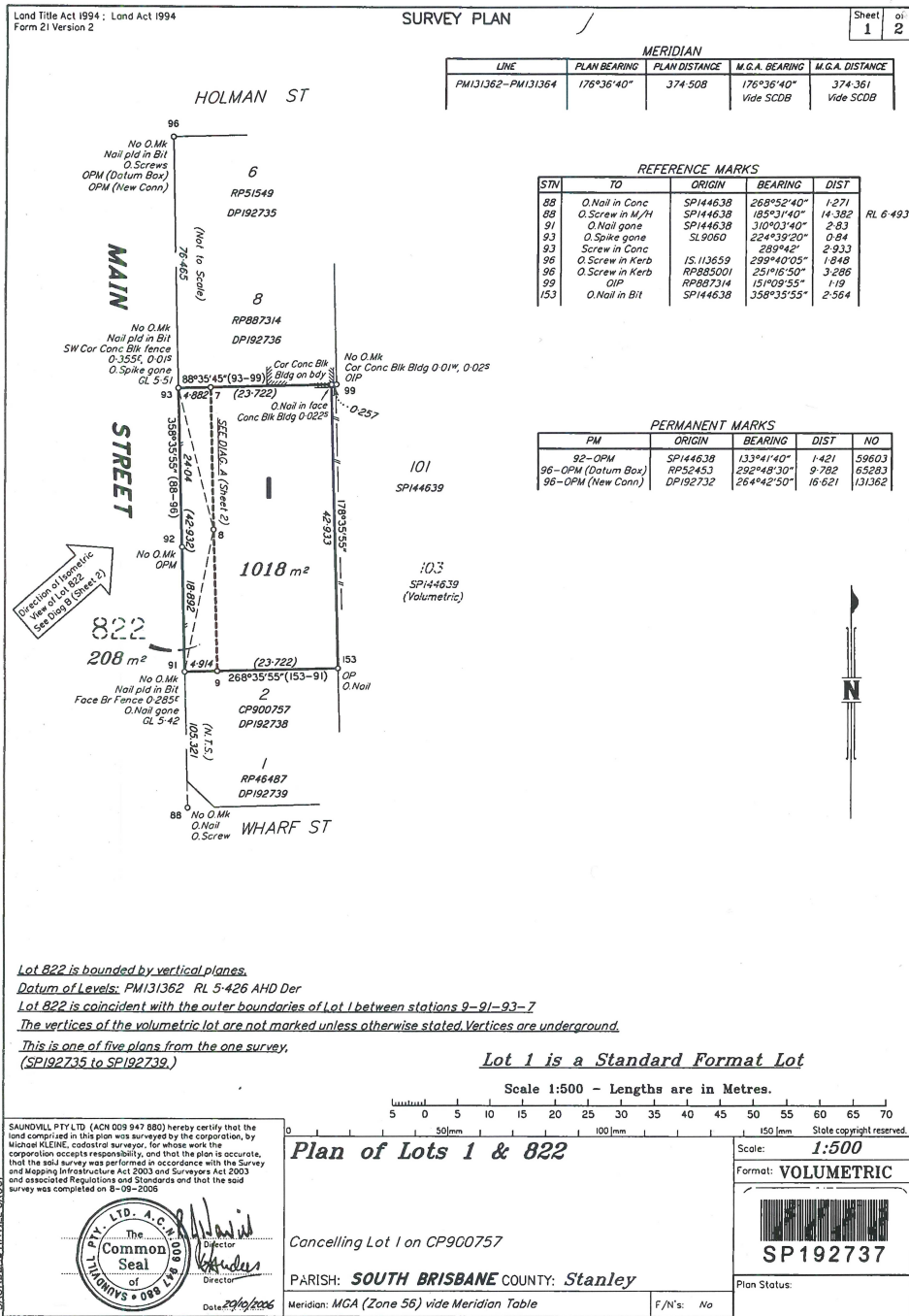
In particular, the isometric view shows the same corner numbers as the plan; for example corner n. 93 in the top view corresponds to *93a* and *93b* in the isometric view. The two corners have the same X and Y values, but the Z value is different. The height value is indicated with *RL* standing for Reduced Level¹ such as, *RL -48.264* which indicates a location below sea level (the unit of measurement is in meters).

¹ Reduced Level: in surveying refers to equating elevations of survey points with reference to a common assumed datum. It is a vertical distance between survey point and adopted datum plane.

<p style="text-align: right; font-size: 1.2em; font-weight: bold;">711248389</p> <p style="text-align: right;">\$375.30 06/12/2007 13:45</p> <p style="font-size: 1.1em; font-weight: bold;">BE 400 NT</p>	<p style="text-align: center; font-size: 0.8em; font-weight: bold;">WARNING : Folded or Mutilated Plans will not be accepted. Plans may be rolled. Information may not be placed in the outer margins.</p> <p>Registered</p> <p>5. Lodged by</p> <div style="border: 1px solid black; padding: 2px; text-align: center; font-size: 0.8em; font-weight: bold;"> 332 Brisbane City Council G.P.O. Box 1434 Brisbane Q. 4000. </div> <p style="font-size: 0.7em;">(Include address, phone number, reference, and Lodger Code)</p>																		
<p>1. Certificate of Registered Owners or Lessees.</p> <p>I/We</p> <p style="text-align: center; font-size: 0.8em; font-weight: bold;">TERENCE PATRICK FLYNN (Director Portfolio), a duly authorised delegate of the Director-General, Department of Public Works</p> <p>(Names in full)</p> <p>* as Registered Owners of this land agree to this plan and dedicate the Public Use Land as shown hereon in accordance with Section 50 of the Land Title Act 1994.</p> <p>* as Lessees of this land agree to this plan.</p> <p>Signature of *Registered Owners *Lessees</p> <p style="font-size: 0.7em; margin-top: 20px;">* Rule out whichever is inapplicable</p>	<table border="1" style="width: 100%; border-collapse: collapse; font-size: 0.8em;"> <tr> <th colspan="3">6. Existing</th> <th colspan="3">Created</th> </tr> <tr> <th>Title Reference</th> <th>Lot</th> <th>Plan</th> <th>Lots</th> <th>Emts</th> <th>Road</th> </tr> <tr> <td>50433615</td> <td>1</td> <td>CP900757</td> <td>1 & 822</td> <td></td> <td></td> </tr> </table> <p style="font-size: 0.7em; margin-top: 20px;">Encroachment notice issued to the owner of Lot 8 on RP887314 on 20/10/2006, in accordance with s19 of the Survey and Mapping Infrastructure Regulation 2004.</p>	6. Existing			Created			Title Reference	Lot	Plan	Lots	Emts	Road	50433615	1	CP900757	1 & 822		
6. Existing			Created																
Title Reference	Lot	Plan	Lots	Emts	Road														
50433615	1	CP900757	1 & 822																
<p>2. Local Government Approval.</p> <p>* hereby approves this plan in accordance with the:</p> <p>%</p> <p>Dated this ... day of</p> <p style="text-align: center;">#</p> <p style="text-align: center;">#</p> <p style="font-size: 0.7em;">* Insert the name of the Local Government. % Insert Integrated Planning Act 1997 or # Insert designation of signatory or delegation Local Government (Planning & Environment) Act 1990</p>	<table border="1" style="width: 100%; border-collapse: collapse; font-size: 0.8em;"> <tr> <td style="width: 50%;">1 & 822</td> <td style="width: 50%;">Lot 1 on CP900757</td> </tr> <tr> <td>Lots</td> <td>Orig</td> </tr> </table> <p>7. Portion Allocation :</p> <p>8. Map Reference : 9543-33313</p> <p>9. Locality : Kangaroo Point</p> <p>10. Local Government : BRISBANE CITY</p> <p>11. Passed & Endorsed :</p> <p>By: SAUNDVILL PTY LTD Date: 20/10/2006 Signed: <i>[Signature]</i> Designation: Cadastral Surveyor\Director</p> <p>12. Building Format Plans only. I certify that: * As far as it is practical to determine, no part of the building shown on this plan encroaches onto adjoining lots or roads. * Part of the building shown on this plan encroaches onto adjoining *lots and road</p> <p>13. Lodgement Fees :</p> <table style="font-size: 0.7em;"> <tr><td>Survey Deposit</td><td style="text-align: right;">\$</td></tr> <tr><td>Lodgement</td><td style="text-align: right;">\$</td></tr> <tr><td>... New Titles</td><td style="text-align: right;">\$</td></tr> <tr><td>Photocopy</td><td style="text-align: right;">\$</td></tr> <tr><td>Postage</td><td style="text-align: right;">\$</td></tr> <tr><td>TOTAL</td><td style="text-align: right;">\$</td></tr> </table> <p>14. Insert Plan Number SP192737</p>	1 & 822	Lot 1 on CP900757	Lots	Orig	Survey Deposit	\$	Lodgement	\$... New Titles	\$	Photocopy	\$	Postage	\$	TOTAL	\$		
1 & 822	Lot 1 on CP900757																		
Lots	Orig																		
Survey Deposit	\$																		
Lodgement	\$																		
... New Titles	\$																		
Photocopy	\$																		
Postage	\$																		
TOTAL	\$																		
<p>3. Plans with Community Management Statement :</p> <p>CMS Number :</p> <p>Name :</p>	<p>4. References :</p> <p>Dept File :</p> <p>Local Govt :</p> <p>Surveyor : 4087</p>																		

Figure B.1: Complete paper showing a survey plan of a volumetric format parcel - front page.

SP192737 V0 REGISTERED Recorded Date 06/12/2007 13:45 Page 1 of 3 Not To Scale



Copyright protects the plan/s being ordered by you. Unauthorised reproduction or amendments are not permitted.

Figure B.2: Complete paper showing a survey plan of a volumetric format parcel - page 1/2.

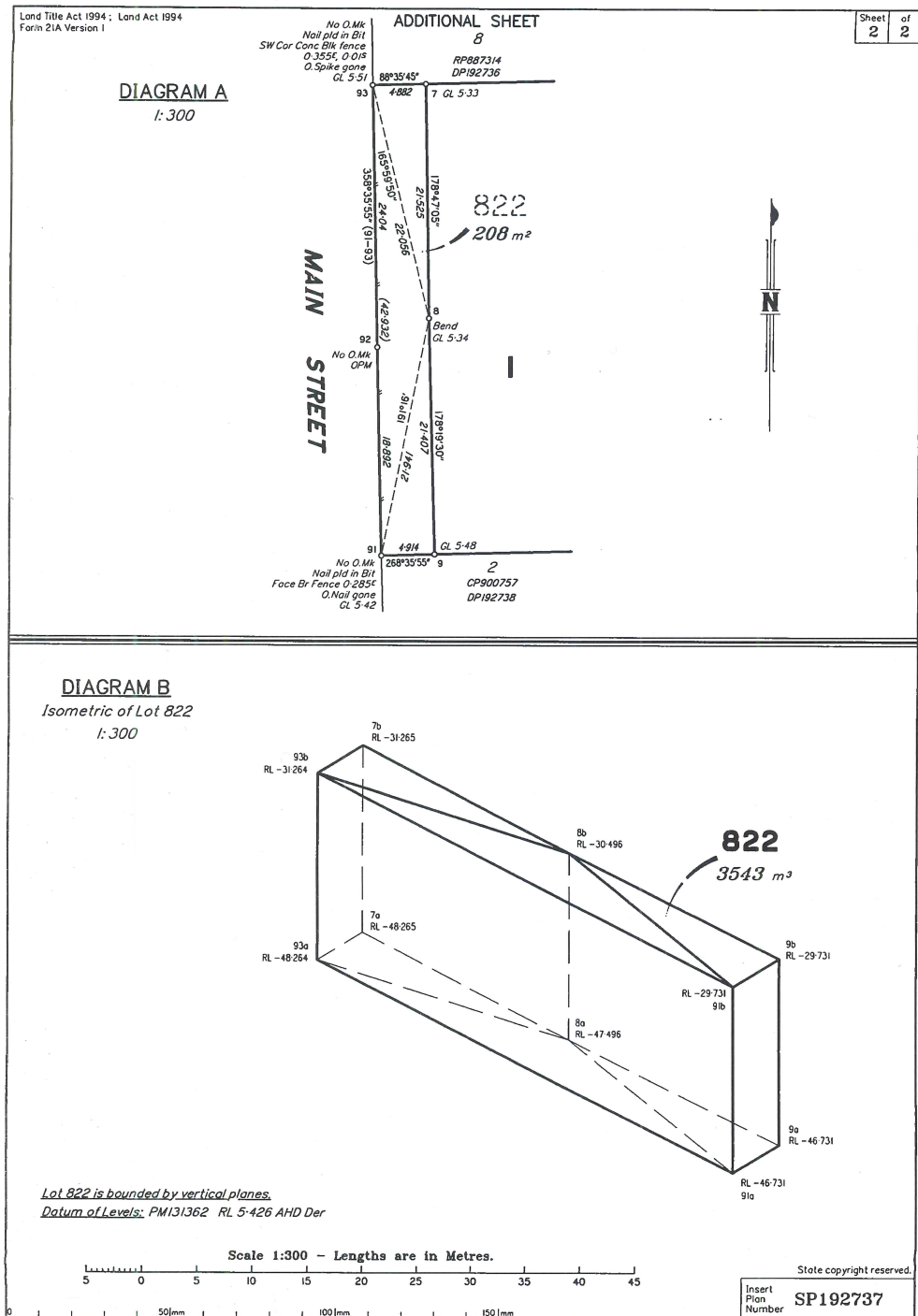


Figure B.3: Complete paper showing a survey plan of a volumetric format parcel - page 2/2.

C

JAVASCRIPT CODE

This appendix contains the whole code which has been written for the implementation of the prototype. The programming language used is JavaScript, using Cesium JS as a platform to simplify the implementation of [WebGL](#) applications. The most relevant sections of the code are listed and explained below, together with the reference to the line numbers:

- **29-37:** Create a label to show X, Y and Z values when hovering over [KML](#) objects.
- **40-57:** Add [KML](#) objects as 'entities' to the Cesium viewer.
- **62-71:** Add [KML](#) datasource to the Cesium viewer.
- **75-85:** Reset, needed when the user chooses another height level option from the selection list.
- **88-173:** Selection list control to choose one of the height levels; here is where the loading of the [KML](#) files is done (see [Section 6.9](#)).
- **178-186:** Checkbox to switch 2D parcels layer on/off ([Section 6.7](#)).
- **220-276:** Event listener for handling the "onclick" event on a [KML](#) object (i.e. selection/picking of a [KML](#) object/feature as described in [Section 6.10](#)).
- **282-300:** Function to show the descriptive information of the selected [KML](#) object (see [Section 6.11.2](#)).
- **304-352:** Function to get the owner(s) of a parcel, via a [WFS](#) request (see [Section 6.11.2](#)).
- **356-403:** Function to get the parcels belonging to a certain owner, via a [WFS](#) request (see [Section 6.12](#)).
- **405-456:** Functions for sending a [WFS](#) request and parsing the [JSON](#) response (see [Section 6.12](#)).
- **459-484:** Event listener to handle the click on 'get administrative info' button; it invokes the 'get owners' function.
- **489-538:** Checkbox to show X, Y and Z values when hovering over [KML](#) objects.
- **550-588:** Create text box and button to search for parcels belonging to a certain owner (via [WFS](#), see [Section 6.12](#)).
- **591-636:** Event listener for handling the text input and search button for finding the parcels of a certain owner (via [WFS](#), see [Section 6.12](#)).
- **639-642:** Standard reset of the Cesium Sandcastle test environment (now not used).


```

1  var viewer = new Cesium.Viewer('cesiumContainer');
2  var scene = viewer.scene;
3
4  // add credit and disclaimer
5  var credit = new Cesium.Credit( {text:'Test data provided by
   Queensland Cadastre - disclaimer: ownership information is
   randomly generated, geometry is not up-to-date'}, false);
6  viewer.scene.frameState.creditDisplay.addDefaultCredit(credit) ;
7
8  // Set the initial camera view to look at Brisbane
9  var initialPosition = Cesium.Cartesian3.fromDegrees(153.0331294,
   -27.4721613, 553);
10 var initialOrientation = new Cesium.HeadingPitchRoll.fromDegrees
   (21.27879878293835, -21.34390550872461, 0.0716951918898415);
11
12 viewer.scene.camera.setView({
13   destination: initialPosition,
14   orientation: initialOrientation,
15   endTransform: Cesium.Matrix4.IDENTITY
16 });
17
18
19 var options = {
20   camera : viewer.scene.camera,
21   canvas : viewer.scene.canvas
22 };
23
24
25
26 var checked_2D = true ;
27 var checked_xyz = false ;
28
29 var labelEntity = new Cesium.Entity() ;
30 labelEntity.label = {
31   show : false,
32   showBackground : true,
33   font : '14px monospace',
34   horizontalOrigin : Cesium.HorizontalOrigin.LEFT,
35   verticalOrigin : Cesium.VerticalOrigin.BASELINE,
36   pixelOffset : new Cesium.Cartesian2(15, 0)
37   } ;
38
39
40 function add_entities(data) {
41
42   var length = data.entities.values.length ;
43   var entity = new Cesium.Entity() ;
44
45   for (var i = 0; i < length; i++) {
46
47     entity = data.entities.values[i];
48     /*
49     if (entity._children.length > 1) {
50       console.log( 'id = '+ entity.id ) ;
51     }
52     */
53     viewer.entities.add(entity) ;
54
55   }
56   console.log('Added datasource '+data.name+' as '+length+' viewer
   entities') ;
57 }
58
59 var dataSource2D_ix = 0 ;
60 var dataSource3D_ix = 0 ;
61
62 function add_dataSource(data) {
63   //var test_folder = data.entities.getById('ParcelType-E') ;
64   //console.log(test_folder) ;
65

```

```

66     viewer.dataSources.add(data);
67     console.log('Added datasource '+data.name) ;
68
69     return viewer.dataSources.indexOf(data) ;
70
71 }
72
73
74
75 function removeAndReset() {
76
77     document.getElementById("textId").disabled = false;
78     document.getElementById("searchId").disabled = false;
79     viewer.selectedEntity = undefined ;
80     viewer.dataSources.removeAll();
81     dataSource2D_ix = 0 ;
82     dataSource3D_ix = 0 ;
83     viewer.entities.removeAll() ;
84     viewer.entities.add(labelEntity) ;
85 }
86
87
88 Sandcastle.addToolBarMenu([
89 {
90     text : 'Brisbane, cadastral parcels, two layers',
91     onselect : function() {
92
93
94         // not in reset but here, because of switching on/off layers
95         removeAndReset() ;
96
97         var promise2D = Cesium.KmlDataSource.load('../SampleData/kml/
98             window_2D.kml', options ) ;
99         promise2D.then( function(data) {
100
101             //add_entities(data) ;
102             dataSource2D_ix = add_dataSource(data) ;
103
104         }); // end then
105
106
107         var promise3D = Cesium.KmlDataSource.load('../SampleData/kml/
108             window_3D.kml', options ) ;
109         promise3D.then( function(data) {
110
111             add_entities(data) ;
112             //dataSource3D_ix = add_dataSource(data) ;
113
114             document.getElementById("textId").disabled = false;
115             document.getElementById("searchId").disabled = false;
116
117         }); // end then
118
119     }
120 },
121 {
122
123     text : 'Same parcels, height raised 200m, one layer',
124     onselect : function() {
125
126         // not in reset but here, because of switching on/off layers
127         removeAndReset() ;
128
129         var promise = Cesium.KmlDataSource.load('../SampleData/kml/
130             window200.kml', options ) ;
131         promise.then( function(data) {
132
133             add_entities(data) ;
134             //dataSource2D_ix = add_dataSource(data) ;

```

```

134
135     document.getElementById("textId").disabled = false;
136     document.getElementById("searchId").disabled = false;
137
138 }); // end then
139
140     }
141
142 },
143 {
144     text : 'New data, height raised 200m, two layers',
145     onselect : function() {
146
147         // not in reset but here, because of switching on/off layers
148         removeAndReset() ;
149
150         var promise2D = Cesium.KmlDataSource.load('.././SampleData/kml/
151             window_2D200.kml', options ) ;
152         promise2D.then( function(data) {
153
154             //add_entities(data) ;
155             dataSource2D_ix = add_dataSource(data) ;
156
157         }); // end then
158
159
160         var promise3D = Cesium.KmlDataSource.load('.././SampleData/kml/
161             window_3D200.kml', options ) ;
162         promise3D.then( function(data) {
163
164             add_entities(data) ;
165             //dataSource3D_ix = add_dataSource(data) ;
166
167             document.getElementById("textId").disabled = false;
168             document.getElementById("searchId").disabled = false;
169
170         }); // end then
171
172     }
173 }], 'toolbar');
174
175
176
177
178 Sandcastle.addToggleButton( 'Switch 2D parcels on/off', checked_2D,
179     function(checked_2D) {
180
181         viewer.dataSources.get(dataSource2D_ix).show = checked_2D ;
182
183         //viewer.entities.getById('Parcels-2D').show = checked_2D ;
184         //viewer.entities.getById('ParcelType-E').show = checked_2D ;
185     }
186 );
187
188
189
190
191 // Information about the currently selected feature
192 var selected = {
193     feature: undefined,
194     originalColor: new Cesium.Color()
195 };
196
197
198
199 var foundParcels = [] ;
200
201

```

```

202 // An entity object which will hold info about the currently
      selected feature for infobox display
203 var selectedEntity = new Cesium.Entity();
204
205 var pickedEntities = new Cesium.EntityCollection();
206 var pickColor = Cesium.Color.YELLOW.withAlpha(0.5);
207
208
209
210 function changeColor(entity, color) {
211
212     for (var j = 0; j < entity._children.length; j++) {
213         entity._children[j].polygon.material = new
                Cesium.ColorMaterialProperty(color) ;
214     }
215 }
216
217
218
219
220 // Color a feature on selection and show metadata in the InfoBox.
221 var clickHandler = viewer.screenSpaceEventHandler.getInputAction(
        Cesium.ScreenSpaceEventType.LEFT_CLICK);
222 viewer.screenSpaceEventHandler.setInputAction(function onLeftClick(
        movement) {
223     // If a feature was previously selected, undo the highlight
224     if (Cesium.defined(selected.feature)) {
225         changeColor(selected.feature, selected.originalColor) ;
226         selected.feature = undefined;
227     }
228
229     // Pick a new feature
230     //var pickedFeature = viewer.scene.pick(movement.position);
231     // get an array of all primitives at the mouse position
232     var pickedPrimitives = viewer.scene.drillPick(movement.position);
233     //console.log(pickedPrimitives.length);
234
235     var pickedPrimitive ;
236     var pickedFeature ;
237
238     if (Cesium.defined(pickedPrimitives) && pickedPrimitives.length >
        0 ) {
239         //Update the collection of picked entities.
240         pickedEntities.removeAll();
241
242         //for (var i = 0; i < pickedPrimitives.length; i++) {
243
244             //pickedPrimitive = pickedPrimitives[pickedPrimitives.length -
                1].id;
245             pickedPrimitive = pickedPrimitives[0].id;
246             pickedFeature = pickedPrimitives[0].id._parent;
247
248             pickedEntities.add(pickedFeature);
249
250         //}
251     }
252     else {
253         clickHandler(movement);
254         return;
255     }
256
257     // Select the feature if it's not already selected
258     if (selected.feature === pickedFeature) {
259         return;
260     }
261     selected.feature = pickedFeature;
262
263     // Save the selected feature's original color (Note: here this
        is the color of the first primitive / child of the parent/
        feature)
264

```

```

265     if ( Cesium.defined(pickedPrimitive.polygon.material) ) {
266         Cesium.Color.clone(
                pickedPrimitive.polygon.material.color.getValue(),
                selected.originalColor);
267     }
268
269     // Highlight newly selected feature = parent of picked primitive
270     changeColor(pickedFeature, Cesium.Color.YELLOW);
271
272     //showInfoBox(pickedFeature);
273     showInfoBox(selected.feature);
274
275 }, Cesium.ScreenSpaceEventType.LEFT_CLICK);
276
277
278
279 var infoDiv = document.getElementById('onclickInfo');
280
281
282 function showInfoBox(pickedFeature) {
283     // (Re)set feature infobox title and description
284
285     //var featureName = pickedFeature._name;
286     var featureName = pickedFeature._id;
287     //var featureName = pickedFeature.parent._id;
288
289     selectedEntity.name = featureName;
290     selectedEntity.description = 'Loading <div class="
291         cesium-infoBox-loading"></div>';
292
293     viewer.selectedEntity = selectedEntity;
294
295     selectedEntity.description = pickedFeature.description._value;
296
297     infoDiv.innerHTML="";
298     infoDiv.innerText="";
299
300 }
301
302
303
304 function getOwners(url) {
305
306     var infoDiv = document.getElementById('onclickInfo');
307
308     //var params = 'featureid='+id;
309     var params = '';
310
311     GetFeatureWFS( url, params, function(validJson) {
312         //console.log(validJson);
313
314         var key;
315
316         if (!validJson || validJson.features.length == 0) {
317             handleError(url + params + ' no results ');
318             return false;
319         }
320         if (validJson){
321
322             var content = '<b>'+ validJson.features[0].id + '</b>';
323             content += '<table>';
324             content += '<tr>';
325
326             for(key in validJson.features[0].properties){
327                 //content += '<tr><td>'+ key + '</td><td>'+
328                 validJson.features[0].properties[key] + '</td></tr>';
329                 content += '<td>'+ key + '</td>';
330             }
331

```

```

332     content += '</tr>' ;
333     content += '<tr>' ;
334
335     for(key in validJson.features[0].properties){
336         content += '<td>'+ validJson.features[0].properties[key] +'<
            /td>' ;
337     }
338
339     content += '</tr>' ;
340     content += '</table>' ;
341
342
343     infoDiv.innerHTML=content ;
344     infoDiv.style.display = "block" ;
345
346 }
347
348 return true;
349
350 });
351
352 }
353
354
355
356 function getParcels(url) {
357
358     //var infoDiv = document.getElementById('onclickInfo') ;
359
360     var params = '' ;
361
362     GetFeatureWFS( url, params, function(validJson ) {
363         //console.log(validJson);
364
365         var key ;
366         var segment_parcel_nr ;
367         var foundEntity ;
368         var originalColor = new Cesium.Color() ;
369
370         if (!validJson || validJson.features.length == 0) {
371
372             handleError(url + params + ' no results ');
373             return false;
374         }
375         if (validJson){
376
377             if (validJson.features.length > 100) {
378                 console.log('Found more than 100 parcels; please limit your
                    search');
379                 // todo replace alert
380                 alert('Found more than 100 parcels; please limit your search
                    ');
381                 return false;
382             }
383
384             for (var i=0; i < validJson.features.length; i++) {
385                 segment_parcel_nr = validJson.features[i].properties['
                    segment_nr']+ '_' +validJson.features[i].properties['
                    parcel_nr'] ;
386
387                 foundEntity = viewer.entities.getById(segment_parcel_nr) ;
388
389                 // save current color of first child of this parcel
390                 originalColor = foundEntity._children[0].
                    polygon.material.color.getValue() ;
391                 foundParcels[i] = { segment_parcel_nr, originalColor } ;
392
393                 // change color to highlite color
394                 changeColor(foundEntity, Cesium.Color.RED) ;
395             }
396

```

```

397     }
398
399     return true;
400
401     });
402
403 }
404
405 function handleError(text) {
406     infoDiv.innerHTML=text ;
407     infoDiv.style.display = "block" ;
408 }
409
410
411 function tryParseJSON(jsonString) {
412     try {
413         var o = JSON.parse(jsonString);
414
415         // Handle non-exception-throwing cases:
416         // Neither JSON.parse(false) or JSON.parse(1234) throw
417         // errors, hence the type-checking,
418         // but... JSON.parse(null) returns 'null', and typeof null
419         // == "object",
420         // so we must check for that, too.
421         if (o && typeof o === "object" && o !== null) {
422             return o;
423         }
424     } catch (e) {}
425
426     return false;
427 };
428
429 function GetFeatureWFS(url, params, callback_handleJson) {
430
431     var xhr = new XMLHttpRequest();
432     xhr.onload = function() {
433         if (xhr.status === 200) {
434
435             var validJson = tryParseJSON(xhr.responseText);
436             if (validJson) {
437                 callback_handleJson(validJson) ;
438             } else {
439                 console.log('Response is not valid json. Response
440                 is: ');
441                 console.log(xhr.responseText);
442                 handleError(url + params + ' gives output: ' +
443                 xhr.responseText) ;
444             }
445         } else {
446             console.log('Request failed. Returned status of ' +
447             xhr.status + ' ' + xhr.statusText );
448             handleError(url + params + ' gives error: ' + xhr.status
449             + ' ' + xhr.statusText) ;
450         }
451     };
452     xhr.onerror = function() {
453         console.log('Request failed. Returned status of ' +
454         xhr.status + ' ' + xhr.statusText );
455         handleError(url + params + ' gives error: ' + xhr.status
456         + ' ' + xhr.statusText) ;
457     };
458     xhr.open('GET', encodeURIComponent(url + params));
459     xhr.send();
460 }

```



```

517         var heightString = cartographic.height.toFixed
518             (2);
519         labelEntity.position = cartesian;
520         labelEntity.label.show = true;
521         labelEntity.label.text =
522             'Lon: ' + ( ' ' + longitudeString).slice(-7
523                 ) + '\u00B0' +
524             '\nLat: ' + ( ' ' + latitudeString).slice(
525                 -7) + '\u00B0' +
526             '\nAlt: ' + ( ' ' + heightString).slice(-7)
527                 + 'm';
528
529         labelEntity.label.eyeOffset = new
530             Cesium.Cartesian3(0.0, 0.0,
531                 -cartographic.height * (scene.mode ===
532                     Cesium.SceneMode.SCENE2D ? 1.5 : 1.0));
533
534         foundPosition = true;
535     }
536 }
537
538 if (!foundPosition) {
539     labelEntity.label.show = false;
540 }
541 }, Cesium.ScreenSpaceEventType.MOUSE_MOVE);
542 });
543
544 function toBrisbane() {
545     Sandcastle.declare(toBrisbane);
546     viewer.camera.flyTo({
547         destination: initialPosition,
548         orientation: initialOrientation,
549         endTransform: Cesium.Matrix4.IDENTITY
550     });
551 }
552
553 var toolbarDiv = document.getElementById('toolbar') ;
554
555 var dbSearchContainer = document.createElement('div');
556 dbSearchContainer.className = 'cesium-viewer-dbSearchContainer';
557 dbSearchContainer.id = 'containerId' ;
558
559 var form = document.createElement('form');
560 form.setAttribute('data-bind', 'submit: search');
561 form.id = 'formId' ;
562
563 var textBox = document.createElement('input');
564 textBox.id = 'textId' ;
565 textBox.type = 'search';
566 //textBox.className = 'cesium-dbSearch-input';
567 textBox.setAttribute('placeholder', 'Enter name of owner...');
568 textBox.disabled = true ;
569
570 form.appendChild(textBox);
571
572 var searchButton = document.createElement('span');
573 //searchButton.className = 'cesium-dbSearch-searchButton';
574 searchButton.className = 'cesium-button';
575 searchButton.id = 'searchId' ;
576 searchButton.textContent = "Start search";
577 searchButton.disabled = true ;
578
579 form.appendChild(searchButton);
580
581 var focusButton = document.createElement('span');
582 focusButton.className = 'cesium-button';

```

```

581 focusButton.id = 'focusId' ;
582 focusButton.textContent = "Return to Brisbane";
583 focusButton.onclick = toBrisbane ;
584
585 form.appendChild(focusButton);
586
587 dbSearchContainer.appendChild(form);
588 toolbarDiv.appendChild(dbSearchContainer);
589
590
591 document.querySelector('#searchId').addEventListener('click',
    function (event) {
592
593     // first undo possible results of previous search
594     var parcel_id ;
595
596     for (var j=0; j < foundParcels.length; j++ ) {
597         parcel_id = foundParcels[j].segment_parcel_nr ;
598         //console.log( parcel_id ) ;
599         if (parcel_id) {
600             changeColor( viewer.entities.getById(parcel_id),
                foundParcels[j].originalColor ) ;
601         }
602     }
603     foundParcels.length = 0 ;
604
605     // then new search, todo: check if same query text is fired
        again, if so: return
606     var searchFor = document.getElementById('textId') ;
607
608     if (!searchFor.value.replace(/\s/g, '').length) {
609         //if ( !( /\S/.test(searchFor) ) )
610         // string only contains whitespace (ie. spaces, tabs or line
            breaks)
611         return ;
612     }
613     else {
614         //if (searchFor && searchFor.value.length > 0) {
615         console.log('search for parcels owned by ..') ;
616         console.log(searchFor.value) ;
617
618         var partyname = searchFor.value ;
619         //var typeName = 'geoweb:v_parcel_3d_owned_by' ;
620         var typeName = 'geoweb:v_parcel_owned_by' ;
621
622
623         var url_wfs='http://pakhuis.tudelft.nl:8088/geoserver/geoweb/ows?
            service=WFS&version=1.1.0&request=GetFeature&
            outputFormat=application/json' ;
624         url_wfs=url_wfs+'&typeName='+typeName ;
625         url_wfs=url_wfs+'&propertyName=objectid,segment_nr,parcel_nr' ;
626         url_wfs=url_wfs+"&cql_filter=partyname like '%" +partyname+"%'";
627
628         console.log(url_wfs);
629         console.log("") ;
630
631         getParcels( url_wfs ) ;
632     }
633
634
635 }) ;
636
637
638 Sandcastle.reset = function() {
639     //viewer.clock.clockRange = Cesium.ClockRange.UNBOUNDED;
640     //viewer.clock.clockStep = Cesium.ClockStep.SYSTEM_CLOCK;
641 };
642

```


D | USABILITY QUESTIONNAIRE

The usability questionnaire has been created online by means of Google Forms, which provides a fast way of creating a survey. The advantage of using this tool is also in the fact that the answers are automatically collected in a spreadsheet, so the process of data processing and analysis are significantly fastened. In the following images, a transcript of the official questionnaire is shown.

Usability questionnaire - "Web-based visualization of 3D cadastre"

Welcome to the usability test of the Brisbane 3D cadastre prototype! Thank you for agreeing in participating in this study, your help will be crucial to point out the limitations in this early stage of the prototype development and make subsequent design changes according to your feedback.

The purpose of this study is to determine the initial response in terms of usability of the 3D cadastre prototype built as a product of the MSc thesis research "Web-based visualization of 3D cadastre". The research starts with a number of requirements to be met in order to build a successful prototype, at this stage, only a part of the requirements has been implemented. Please, try to fulfil all the tasks and answer all the questions.

Disclaimer:

The cadastral parcels are part of the Queensland DCDB, but they are not guaranteed to be up to date, also the RRRs and the person/parties involved are fake to protect the privacy of third parties.

* Required

Please fill in your personal information:

1. **Name:** *

2. **Surname** *

3. **To which one of these user groups do you belong to? ***

Mark only one oval.

- Researchers in the field of 2D and 3D cadastre (including staff of the MSc Geomatics for the Built Environment);
- Students of Geomatics for the Built Environment at TU Delft;
- Professionals using cadastral applications (lawyers, notaries, engineers, architects, land surveyors, building managers etc.);
- Managers in the government and municipal authorities in charge of the maintenance of the cadastral system;
- Public and private entities, companies;
- General public, citizens;
- Other: _____

Figure D.1: Questionnaire for the assessment of the usability of the 3D cadastre prototype - page 1/7

4. **Did you ever used a Cesium application before or did you know about Cesium JS? ***

Mark only one oval.

- Yes, I am very familiar with Cesium JS.
- I am a little familiar with Cesium JS, I have used it a few times.
- I have heard about Cesium JS but I have never used it before.
- No, I do not know Cesium JS at all.

5. **Please insert information about the platform that you are using to test the prototype (i.e. computer, operating system, browser used). Note: the prototype may not work on Mac computers.**

6. **Would you like to receive the results of the questionnaire? Fill in your e-mail address below and you will receive the outcome by the 13th of July 2018!**

Section #1

Please perform the following list of tasks and give us your feedback. Note: each question involves a practical task to be carried out on the prototype itself, after that a few questions need to be answered.

Description:

Please make sure you have a working Internet connection. Open the following link on your web browser to start up the prototype: <http://pakhuis.tudelft.nl:8080/edu/Cesium-1.43/Apps/3dcad/>. Before testing more advanced functionalities, it is crucial to get familiar with the basic navigation tools and view controls.

Note: it is suggested to use a mouse.

7. **Task: Navigate to the Brisbane Airport and check where it is located with respect to the river. You can do this in two ways, if you know where the airport is approximately located just pan and zoom to the location, otherwise click on the magnifier icon and type "Brisbane Airport, Australia". ***

Mark only one oval.

- South
- South-East
- Brisbane does not have an airport
- North

Figure D.2: Questionnaire for the assessment of the usability of the 3D cadastre prototype - page 2/7

8. **Opinion: Can you easily navigate through the viewer? Are the controls intuitive?**

9. **Please, give a grade on a scale from 1 to 10 to the usability of this functionality. ***

Mark only one oval.

	1	2	3	4	5	6	7	8	9	10	
Extremely low usability	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Extremely high usability

Section #2

Description:

Once you are familiar with the web viewer you can start playing with the data available by changing the visibility of the layers. The prototype contains layers with cadastral parcels both in 2D and 3D. Depending on the application, you may want to visualize certain layers rather than others.

10. **Task: Which layers can be toggled on/off? ***

Mark only one oval.

- It is possible to toggle on/off the visibility of the 3D parcels only
- It is possible to toggle on/off the visibility of the 2D parcels only
- It is possible to toggle on/off the visibility both of the 2D and 3D parcels
- It is not possible to toggle on/off the visibility of the layers

11. **Opinion: Was it easy to understand which layers could be toggled on/off? If applicable, please explain the difficulties you encountered.**

Figure D.3: Questionnaire for the assessment of the usability of the 3D cadastre prototype - page 3/7

12. Please, give a grade on a scale from 1 to 10 to the usability of this functionality. *

Mark only one oval.

	1	2	3	4	5	6	7	8	9	10	
Extremely low usability	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Extremely high usability

Section #3

Description:

One of the advantages of a 3D cadastral system is the possibility to store and visualize 3D underground parcels. A limitation that many globe based web-viewers have is that the camera cannot navigate under the ground surface, making impossible the visualization of subsurface parcels. For this reason, the "dynamic elevation tool" has been implemented, so that the parcels can be shifted up of a defined amount to be able to see the ones under the earth surface.

13. Task: Suppose you want to take a look at the boundaries of the underground tunnel in our 3D Cadastre test area near the Brisbane city centre. In order to do that, you have to shift the ground surface and navigate around the parcels to visualise them in detail. What is the lowest z-value? (i.e. deepest point of the lowest 3D parcel below the surface)? Note: if you shifted the parcels you have to take in to account the amount of the shift. *

Mark only one oval.

- About -25 m below ground
- About -55 m below ground
- About -100 m below ground
- It is not possible to define the lowest value

14. Opinion: Were you able to navigate and see the details of the parcels from every angle? Could you see the coordinates on the screen?

15. Please, give a grade on a scale from 1 to 10 to the usability of this functionality. *

Mark only one oval.

	1	2	3	4	5	6	7	8	9	10	
Extremely low usability	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Extremely high usability

Section #4

Figure D.4: Questionnaire for the assessment of the usability of the 3D cadastre prototype - page 4/7

Description:

Cadastre is not only about volumes and parcel boundaries, but it is also about legal information. Every parcel has attached legal information about owner(s), ownership rights, and identification number.

16. Task: Who is/are the owner(s) of the underground 3D tunnel parcels? *

Mark only one oval.

- Mr. Smith
 Mr. & Mrs. Smith
 Mr. Senior
 Mr. & Mrs. Senior

17. Opinion: Could you visualize the information easily? Was it easy to understand what to look for?

18. Please, give a grade on a scale from 1 to 10 to the usability of this functionality. *

Mark only one oval.

	1	2	3	4	5	6	7	8	9	10	
Extremely low usability	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Extremely high usability

Section #5**Description:**

Cadastre is also about people and, in particular, owners. Even though the primary focus of the prototype is on visualization, it is important for the user to have access to the 2D and 3D parcels based on ownership of persons.

19. Task: Search for owner "Annie" (note fake names in database for privacy). Which is the highlight colour used to show the parcels owned by Annie? *

Mark only one oval.

- Red
 Yellow
 Blue
 Green

Figure D.5: Questionnaire for the assessment of the usability of the 3D cadastre prototype - page 5/7

20. Opinion: Was the search functionality intuitive?

21. Please, give a grade on a scale from 1 to 10 to the usability of this functionality. *
Mark only one oval.

	1	2	3	4	5	6	7	8	9	10	
Extremely low usability	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Extremely high usability

Additional remarks

Thanks for participating to the usability test questionnaire! Your contribution is fundamental for the development of the 3D cadastre prototype!

Now you have the opportunity of suggesting ways of improving the design and point out what you did not like about the existing prototype.

22. What is the issue that mostly limits the user-friendliness and usability of the 3D cadastre prototype?

23. What functionality did you like best? What do you think is the most useful functionality?

Figure D.6: Questionnaire for the assessment of the usability of the 3D cadastre prototype - page 6/7

24. Please choose 3 functionalities that, according to you, are crucial for the improvement of the 3D cadastre prototype.

Check all that apply.

- Cross-section view
- 3D measurement tools
- Object search
- Display of partly unbounded objects and 'complex' geometries
- Explode view
- Visualization cues
- Sliding
- 3D buffer
- Database support
- Support different models (vector/polyhedral, raster/voxel, point clouds)
- 2D overview map (orientation)
- Handling massive data and caching/tiling between server and client
- Ensure spatial validity (3D vector topology)
- Other: _____

For more information about the requirements see: Barbara Cemellini, Thompson Rod, Marian de Vries and Peter van Oosterom, Visualization/dissemination of 3D Cadastral Information, In proceedings XXVI FIG Congress, 6-11 May 2018:
http://fig.net/resources/proceedings/fig_proceedings/fig2018/papers/ts05c/TS05C_cemellini_rod_et_al_9591.pdf

Powered by
 Google Forms

Figure D.7: Questionnaire for the assessment of the usability of the 3D cadastre prototype - page 7/7

The detailed results of the tasks performed by the users are shown in the following diagrams:

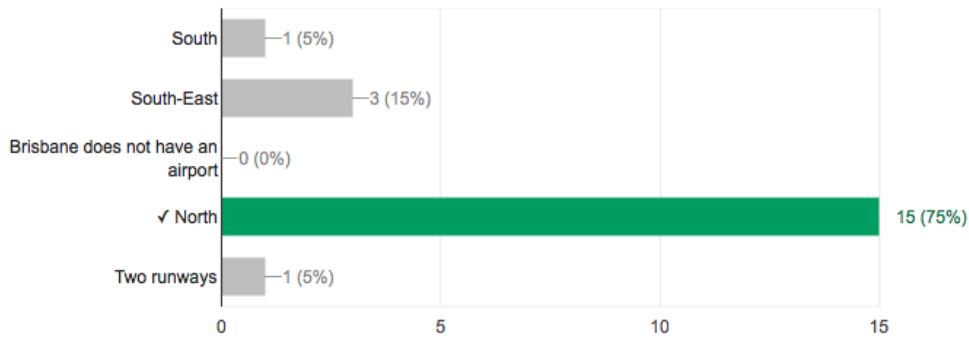


Figure D.8: Responses of Question n.1 - Navigate to the Brisbane Airport and check where it is located with respect to the river. You can do this in two ways, if you know where the airport is approximately located just pan and zoom to the location, otherwise click on the magnifier icon and type "Brisbane Airport, Australia".

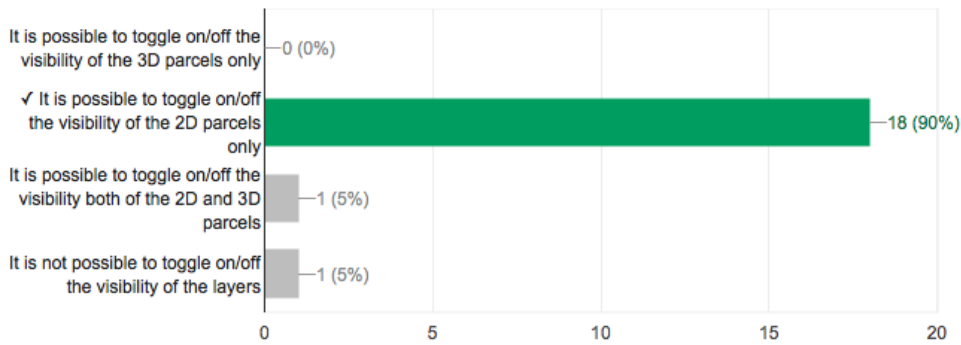


Figure D.9: Responses of Question n.2 - Which layers can be toggled on/off?

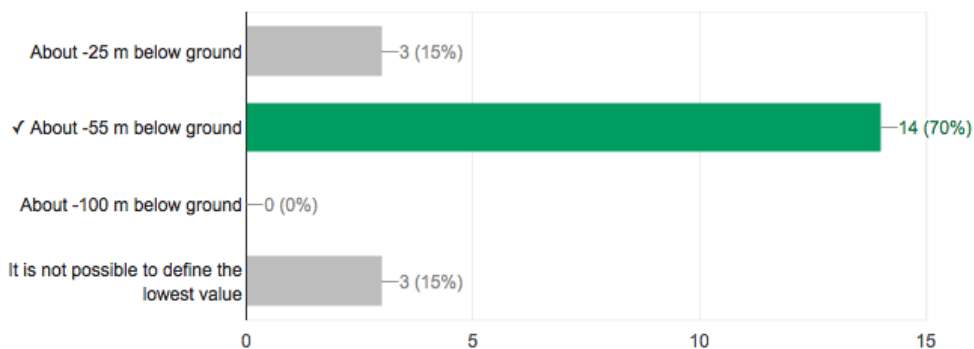


Figure D.10: Responses of Question n.3 - Suppose you want to take a look at the boundaries of the underground tunnel in our 3D Cadastre test area near the Brisbane city centre. In order to do that, you have to shift the ground surface and navigate around the parcels to visualise them in detail. What is the lowest z-value? (i.e. deepest point of the lowest 3D parcel below the surface)? Note: if you shifted the parcels you have to take in to account the amount of the shift.

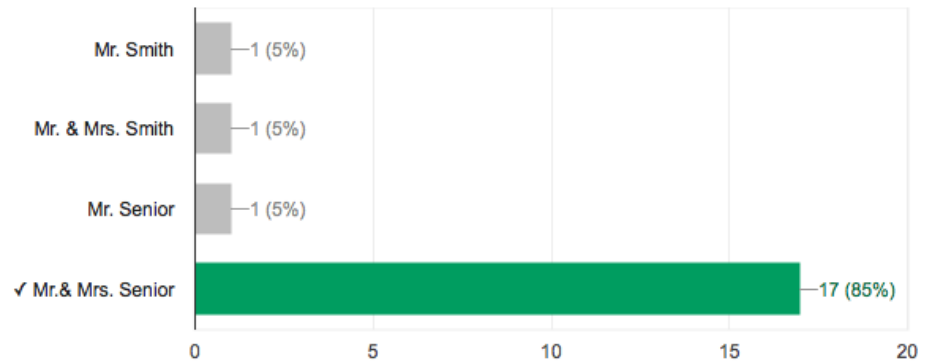


Figure D.11: Responses of Question n.4 - Who is/are the owner(s) of the underground 3D tunnel parcels?

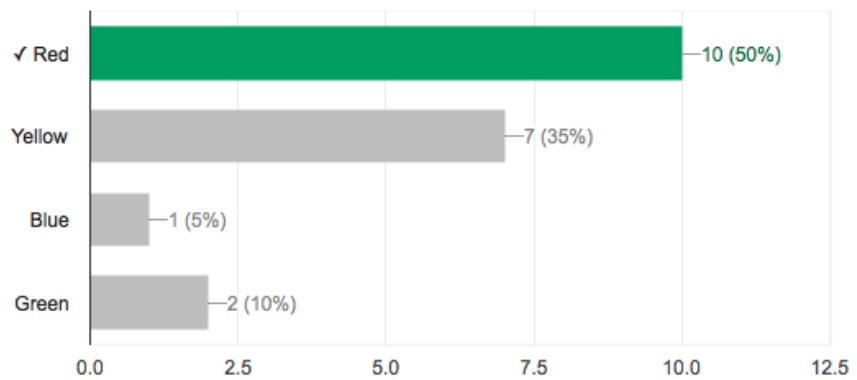


Figure D.12: Responses of Question n.5 - Search for owner "Annie" (note fake names in database for privacy). Which is the highlight colour used to show the parcels owned by Annie?

The graduation research "*Web-based visualization of 3D cadastre*" officially started in November 2017 and lasted for about 8 months, until June/July 2018. The preliminary research on the topic actually started earlier, with the development of a *research assignment* aimed at investigating the requirements related to 3D visualization on a web-viewer and the selection of the most suitable [WebGL](#) platform for the creation of a prototype.

During the development of the research, I had the opportunity to collaborate with my supervisors in the writing and publication of two conference papers related to the topic: the *7th International FIG workshop on the Land Administration Domain Model* held in Zagreb, Croatia in April 2018 and the *FIG Congress "Embracing our smart world where the continents connect: enhancing the geospatial maturity of societies"* held in Istanbul, Turkey in May 2018. These conferences were a good opportunity to present my work to researchers in the cadastre and 3D visualization domains and gain more insights into the [LADM](#) compliant storage schema developed for the Queensland cadastre, which is strictly related to the topic of my MSc thesis, although not in the scope.

In the initial phase of the thesis the research question has been defined, together with five subquestions addressing in detail different aspects of the main question. It must be pointed out that, not all the subquestions have the same importance, some of them are addressed from a technical point of view and some others just theoretically. The reason of this choice is a consequence of the numerous drawbacks encountered in the development of the prototype, in particular the issues related to the [KML](#) data format and its support by Cesium JS. These shortcomings did not allow the investigation of some other aspects related to the topic, examples are communication protocols, caching, tiling and their related issues.

The main aim of this MSc thesis is to define an optimal solution for the visualization and dissemination of 3D cadastral data, through the implementation of a web-based prototype. At first, a list of requirements was made to define the characteristics that every 3D cadastre prototype should have. Unfortunately, it was not possible to fully implement the whole list of requirements within the time frame set for the MSc research, therefore some implementation priorities have been defined, leaving the rest for future work. In addition, thanks to the research about Cesium JS capabilities, the reader is aware of the implementation possibilities offered (at the moment) by the platform even if they have not been physically developed.

Geomatics for the Built Environment is described as a field of study related to geographic data acquisition, analysis, management, visualization and dissemination, with the purpose of gaining knowledge to better understand built and natural environments.

This research primarily focuses on the *visualization* and *dissemination* aspects of geographic data, and uses the knowledge learned throughout the academic career: JavaScript development, [DBMSs](#), storage schemas, [ISO](#) and [OGC](#) standards, 2D and 3D cadastral systems, etc. For these reasons, the thesis fits into the Geomatics MSc program of TU Delft and contributes to broaden the knowledge and deepen the research on 3D cadastre visualization.

The final product of this research, the 3D cadastre prototype (of the Brisbane city centre), is a real world application within the field of Geomatics. This was possible thanks to the use of real data from the Queensland [DCDB](#) and their willingness to collaborate in this project. Despite being a real world application, it is still not complete and ready to use, but it represents a starting point for further development. Moreover, the legal framework of the state of Queensland already foresees the possibility of dealing with 3D parcels, so their visualization on a web-browser would represent a significant upgrade.

The relationship between this research and the wider social context can be seen in the advantages stemming from 3D cadastre visualization and in the relationship between this prototype and the contemporary solutions available in the field of 3D cadastre.

There are numerous benefits stemming from visualization of cadastre in 3D, rather than in 2D. In the first place, the possibility to support the rights of the owners of different strata of a building and the possibility to visualize the information about a single parcel without having issues of overlap. Other advantages are: the ability to differentiate between surface and subsurface parcels and the possibility to easily handle complex situations related to the growing complexity of the built environment in modern cities (for example, situations in which a road passes through a building). All these circumstances are (partially) solved by the 3D cadastre visualization prototype built for the purpose of this research.

Additionally, from the analysis of the existing prototypes, it turned out that the current solutions, developed in different countries of the world, are not fully equipped to accurately visualize 3D cadastral data, but they present some deficiencies that this work aims to solve.

Some gaps that have been filled are the availability of a plug-in free interface supported by all major browsers, the possibility to visualize underground parcels even if it is not fully supported by the platform and the handling of the main occlusion issues compromising the visualization of the 3D volumes.

COLOPHON

This document was typeset using \LaTeX . The document layout was generated using the `arsclassica` package by Lorenzo Pantieri, which is an adaption of the original `classicthesis` package from André Miede.

