FACULTY OF ENGINEERING
AND ARCHITECTURE

Pre-culling geometric linked building data for lightweight viewers

Linked Data

Master's dissertation submitted in order to obtain the academic degree of
Master of Science in de ingenieurswetenschappen: architectuur

Supervisor: Prof. ir.-arch. Paulus Present

Counselors: Ir.-arch. Jeroen Werbrouck
Prof. dr. ir. arch. Ruben Verstraeten

Philippe Soubrier 01702837 philippe.soubrier@ugent.be
Academic year: 2022–2023

GHENT
UNIVERSITY

Contents

List of Figures

List of Tables

Short abstract

This is my short abstract.

Abstract

This is my abstract

Chapter 1

Introduction

From 2D, to 3D and now to BIM. The evolution of the Architecture, Engineering and Construction (AEC) industry has been a long and complex one. The introduction of 3D modeling was the first major step in the industry's evolution, as it allowed for more accurate representations of buildings. No longer solely relying on 2D drawings, a 3D model of a building can be used to create various representations, from a simple 2D floor plan to a full 3D model. Following the adoption of 3D modeling, the implementation of Building Information Modelling (BIM) emerged as another significant milestone. BIM adds an extra layer of information on top of the 3D model. As the digital representation of a building's physical and functional characteristics, BIM serves as a repository for semantics originating from various applications throughout the design and construction processes, including cost estimation, energy analysis, and production planning.
-> needs refs to support point (about infustries in AEC)

However, as mentioned in Werbrouck, 2018, the next chalenge for the AEC industry is related to the ecosystem of current BIM softwares, which remains closed off to other disciplines. This data management challenge is currently being adressed by the Linked Building Data Community Group and other research entities, such as the University of Ghent, through the use of Web of Data technologies[1]. This emerging milestone will be discussed in this thesis under the term Linked Data BIM (LDBIM).

Proposal

Each of these evolutions has brought, and will continue to bring, a significant amount of data together. This volume is expected to grow exponentially in the future as the industry shifts towards a more digital approach and opens up to other stakeholders. The data graphs will not only expand in terms of semantics but also in geometry. This makes visual querying, or simply put, 3D exploration of models, an increasingly diffi-

[1]"Linked Building Data Community Group", n.d.

cult task. Especially when looking at newer devices used in the industry such as mobile phones, and tablet, which are becoming more and more powerful, but still have limited computational resources in comparision to office computers.

To bring this volume of geometric data in perspective, Table 1.1 shows the size of the test-models used in Johansson et al., 2015 , a study from 2015 on the performance of BIM viewers for large models with the following description:

" Although the Hotel model contains some structural elements they are primarily architectural models. As such, no Mechanical, Electrical or Plumbing (MEP) data is present. However, all models except the Hospital contain furniture and other interior equipment. " (Johansson et al., 2015)

Model	# of triangles	# of objects	# of geometry batches
Library	3 685 748	7318	11 195
Student House	11 737 251	17 674	33 455
Hospital	2 344 968	18627	22 265
Hotel	7 200 901	41 893	62 624

Table 1.1: Size of test-models in Johansson et al., 2015

These models demonstrate how basic BIM models can already contain a significant amount of data. LDBIM will not only bring together new stakeholders but also have the capability to keep track of multiple geometry versions for each object, should they occur. Therefore, this thesis proposes a new approach to visual querying of LDBIM models, wherein viewers will not have to load the entire model into memory. Instead, after filtering at the source, only the geometry needed for the visual tasks at hand will be loaded, while maintaining the original link to each resource for further processing and use cases. This filtering step is commonly referred to as culling in the computer graphics industry and is illustrated in Figures 1.1 and 1.2.

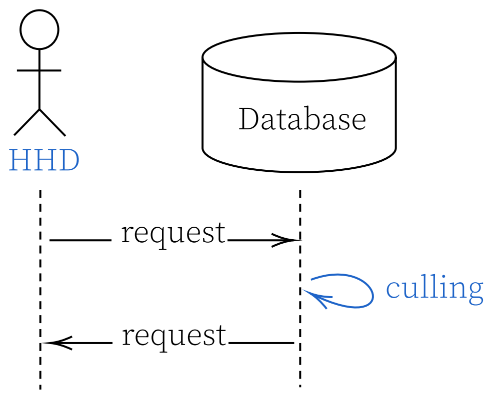

Figure 1.1: Sequence diagram - basic concept

Figure 1.1 illustrates the basic idea of this thesis, presenting an extra step in the communication between a user, represented here by a Hand Held Device (HHD), and a database storing the model. An HHD has been chosen to exemplify a low-powered device used in the field, which requires a lightweight 3D viewer to visualize and explore the digital twin of the building. The HHD is assumed to have no knowledge of the LDBIM model and only receives the geometry that needs to be displayed from the database. On the other hand, the database is assumed to possess, or have access to, all the knowledge of the model and the necessary semantics to perform the culling.

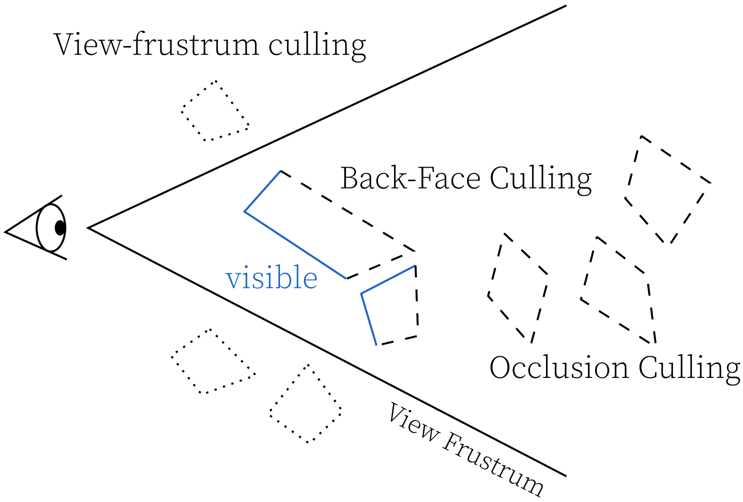

Figure 1.2: Illustration of culling principle, based on Cohen-Or et al., 2003

Figure 1.2 showcases multiple culling techniques to give an idea of what culling means. The first technique, *frustum culling*, is used to determine which objects are visible to the user. The second technique, *occlusion culling*, is used to determine which objects are occluded by or behind other objects. And lastly, *back-face culling*, is used to determine which faces, and not whole objects, are facing away from the user.

1.1 Linked Data

As mentioned in the Introduction, the evolution from BIM to LDBIM is an evolution of the data *management* layer. "Linked Data", as stated by the World Wide Web Consortium (W3C), is a collection of interrelated datasets on the Web, formatted in a standard way that is accessible and manageable by Semantic Web tools. The same applies to the relationships among them.[2] The following collection of Semantic Web technologies provides the required environment to achieve this goal.

1.1.1 RDF and triples

At the origin of this

[2] "Data - W3C", n.d.

```
@prefix fog: <https://w3id.org/fog#> .
@prefix omg: <https://w3id.org/omg#> .
@prefix bot: <https://w3id.org/bot#> .
@prefix rdf:      <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs:     <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf:      <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs:     <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd:      <http://www.w3.org/2001/XMLSchema#> .
@prefix flupke:   <http://flupke.archi#> .

flupke:room1 rdf:type bot:Zone ;
    bot:containsElement flupke:coneOBJ ;
    bot:containsElement flupke:coneOBJlit ;
    bot:containsElement flupke:cubeGLTF ;
    bot:containsElement flupke:cubeGLTF2 ;
    bot:containsElement flupke:cubeGLTFlit ;
    bot:containsElement flupke:IcosphereSTL ;
    bot:containsElement flupke:IcosphereSTLlit .

flupke:coneOBJ omg:hasGeometry flupke:coneOBJ_geometry-1 ;
    rdf:type bot:Element .

flupke:coneOBJlit omg:hasGeometry flupke:coneOBJlit_geometry-1 ;
    rdf:type bot:Element .

flupke:cubeGLTF omg:hasGeometry flupke:cubeGLTF_geometry-1 ;
    rdf:type bot:Element .

flupke:cubeGLTF2 omg:hasGeometry flupke:cubeGLTF2_geometry-1 ;
    rdf:type bot:Element .

flupke:cubeGLTFlit omg:hasGeometry flupke:cubeGLTFlit_geometry-1 ;
    rdf:type bot:Element .

flupke:IcosphereSTL omg:hasGeometry flupke:IcosphereSTL_geometry-1 ;
    rdf:type bot:Element .

flupke:IcosphereSTLlit omg:hasGeometry flupke:IcosphereSTLlit_geometry-1 ;
    rdf:type bot:Element .

flupke:coneOBJ_geometry-1 rdf:type omg:Geometry ;
    fog:asObj_v3.0-obj
    ↪  "https://raw.githubusercontent.com/flol3622/AR-Linked-BIM-viewer/
    ↪  main/public/assets/database_1/coneOBJ.obj"^^xsd:anyURI
    ↪  .

flupke:cubeGLTF_geometry-1 rdf:type omg:Geometry ;
    fog:asGltf_v1.0-gltf
    ↪  "https://raw.githubusercontent.com/flol3622/AR-Linked-BIM-viewer/
    ↪  main/public/assets/database_1/cubeGLTF.gltf"^^xsd:anyURI
    ↪  .
```

```
flupke:cubeGLTF2_geometry-1 rdf:type omg:Geometry ;
    fog:asGltf_v1.0-gltf
    ↪  "https://raw.githubusercontent.com/flol3622/AR-Linked-BIM-viewer/⌋
    ↪  main/public/assets/database_1/cubeGLTF2.gltf"^^xsd:anyURI
    ↪  .

flupke:IcosphereSTL_geometry-1 rdf:type omg:Geometry ;
    fog:asStl_v1.0-ascii
    ↪  "https://raw.githubusercontent.com/flol3622/AR-Linked-BIM-viewer/⌋
    ↪  main/public/assets/database_1/IcosphereSTL.stl"^^xsd:anyURI
    ↪  .

flupke:coneOBJlit_geometry-1 rdf:type omg:Geometry ;
    fog:asObj_v3.0-obj """# Blender 3.4.1
```

1.1.2 Ontologies and reasoning

-> what is an ontology
-> what is inference
-> what is a reasoner

1.1.3 Triplestores and SPARQL

-> what is a triplestore
-> what is SPARQL
-> what is a SPARQL endpoint

1.1.4 Complexity of the graph

-> new start

1.1.4.1 BIM geometry

The 3D model of a building consists of a multitude of sub-models, describing objects for all the different stakeholders participating in the project. Some describe very large objects, and some very small parts. Both can be defined in their most simple and abstract form or have an intricate and complex geometry. As a basic example, can a door simply be defined as a box, or up to the level of the screw-thread for the hinge system. The level of abstraction is here described as the Level of Detail (LOD), it is most of the time pre-selected for the needs of a BIM model, and is applied throughout a single model.

As shown in Figure 1.3, a standard BIM workflow goes through multiple phases each with their associated model and LOD. The LOD is a very important concept in the AEC

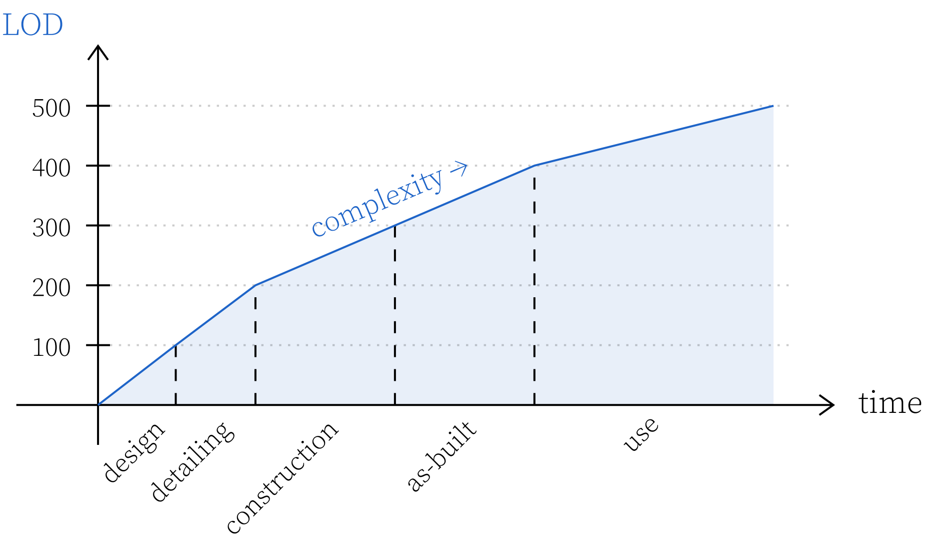

Figure 1.3: Evolution of LOD during the life-cycle of a building.

industry, as it allows for a very efficient workflow. Approaching the modelling step from a top-down perspective, starting with rougher geometries describing the rougher ideas of a concept model and evoluting to a more refined model for the construction phase. As last and longest standing model, can a higher LOD be used to describe subtle changes in the evolution of a building.

This amount of data, both accounting for the

-> +MEP + structural
-> bim model size in Johansson et al., 2015
–> old models
–> table with data
–> final conclusion about LOD here applicable

1.1.4.2 LDBIM geometry

-> Own definition of LDBIM
-> Why the need / What are the advantages of LDBIM?

The interconnectivity of semantics can also be applied to geometry descriptions. Which could allow the co-existence of multiple LODs in a single model. Besides storing the evolution of a single element's geometry, it allows the linking of the different LODs to each other. In contrary to standard BIM models, as explained in 1.1.4.1.

1.1.4.3 Computing power dilemma

The enrichment of the LDBIM-graph also comes with a cost. The amount of data that needs to be stored and processed is much larger than a standard BIM model. Viewers greatly suffer from enrichment as most standard applications require the full model to be loaded in memory.

Although most office computers used in the AEC industry are powerful enough to handle large models, the hardware used in the field is not. The most common hardware used in the field is a tablet or a laptop. Both of these are not powerful enough to handle the amount of data required to display a LDBIM model. This is a problem as these low-power devices are the only solution when it comes to mobility. Furthermore, the size of the models that are more and more enriched can, in some cases, pose a problem to the hardware of a standard office computer.

-> Johansson et al., 2015
–> refer to table scene size + fps => problem (cite fps sources)

1.2 Research questions

-> Taking advantage of database approach instead of file to avoid excess storage on high computational tasks
-> even before BIM viewer itself as in Johansson et al., 2015

It can be inferred from 1.1.4.3 that filtering is necessary in order to visualize the geometry of a LDBIM model, due to its complexity and size. This filtering step, as shown in Figure 1.1, is also known as culling in 3D computer graphics. Culling is the process of removing objects, or parts of objects, from the scene that are not visible to the user. This is done to reduce the amount of data that needs to be processed and stored. Implementing such technology in the context of LDBIM is the proposal of this thesis. The main goal being the introduction of similar algorithms within this context. As culling algorithms are not new and part of a field of research in continuous expansion.

The research questions are therefore focused on the feasibility of this introduction. And propose a set of possible solutions tailored to this specific problem, while highlighting possibilities for future research and specific use cases.

Figure 1.1 illustrates the basic idea of this thesis. Being the extra step inside the communication between a user, here represented as a HHD, and a database storing the model. A HHD has been chosen to illustrate a low-powered device used in the field, that requires a lightweight 3D viewer to visualize and explore the digital twin of the building. The HHD is assumed to have no knowledge of the LDBIM model, and only receives the geometry that is required to be displayed from the database. On the other hand, the database is assumed to have, or access to, all the knowledge of the model and the needed semantic to perform the culling.

1.2.1 To which extent can LDBIM geometry be culled to be streamed to lightweight viewers?

This thesis focuses on cumputing with data snippets or triples inside a LDBIM model, not wihtin. Meaning that the smallest unit of data that can be culled is the one described in one triple, in the most likely scenario: a single LOD of a single element. It implies that geometry is defined and separated at the object-level. It also implies that culling thechniques such as back-face culling will not be handled in this thesis, and will be left to the viewer itself, not the database.

Which snippets of data are needed by the viewer? Is part of the question. The basic needs of the viewer consists firstly of the geometry itself, selecting the right geometry format for the application as well as the additional visual information such as color, texture, etc. Secondly, the identifier of each element is of crucial importance to maintain the link to other semantic ressources in the graph. Enabling the viewer to retrieve those ressources for a multitude of usecases. Transforming it into a user-friendly visual query tool.

The impact of the culling on the performance of the viewer will greatly determine the mutl

-> complete

1.2.2 Can existing semantic and ontologies be used to feed possible culling algorithms?

–>Johansson et al., 2009
—> specitfic terminology also used in linked data
-> Can GIS ontologies be used too?

Chapter 2

State of the art

-> Introduction to the chapter, both tools and existing approaches to create overview of the state of the art

2.1 Related Technologies and Tools

2.1.1 Qoniq and LOD Streaming for BIM

-> Why I chose this one (why special)
-> LOD streaming principle
-> Probably present it as a goal but in the older framework (for effectiveness(esthetics and performance))
->In unity, the maximum amount of ram cannot exceed 2Gb[1]

2.1.1.1 Qoniq's approach to LOD streaming

(T. Strobbe, personal communication, November 25, 2022)
-> in contrary to Johansson et al., 2015

2.1.1.2 Advantages and challenges

2.1.1.3 Potential applicability to LDBIM

-> as stated before advantage of LOD existing

2.1.2 ld-bim.web.app

https://ld-bim.web.app/ -> Where does it come from
-> Detailed explanation of the features / capabilities

[1]"Unity - Manual: Memory in Unity WebGL", n.d.

-> Detailed fragmentation of missed opportunities / how this thesis positions itself to it

2.1.3 AEC related ontologies

-> as proposed in Johansson et al., 2015
–> using BIM semantics to introduce new culling Techniques (5)
–> CHC++
-> Present the following as related work but still in early stage, needs way more research (for computer scientists)

2.1.3.1 BOT

2.1.3.2 FOG and OMG

2.1.4 GIS related ontologies

-> Highlighting maturity and usefull data

2.1.4.1 geoSPARQL

-> 2D Limitations

2.2 Existing Approaches in BIM 3D Viewers and Visualization Techniques

- DDS CAD Viewer
- Tekla BIMsight
- Autodesk Navisworks
- Solibri Model Viewer

2.2.1 General Features

-> Johansson et al., 2015
–> table 3 about Acceleration techniques

2.2.2 Interoperability

2.2.3 Scalability

2.2.4 Collaboration and Data Sharing

2.2.5 Customization and Extensibility

Chapter 3

Culling approaches

3.1 AEC related ontologies

3.1.1 BOT

3.1.2 FOG and OMG

3.2 GIS related ontologies

3.2.1 geoSPARQL

Chapter 4

Setup

4.1 Participants

This is a diagram:

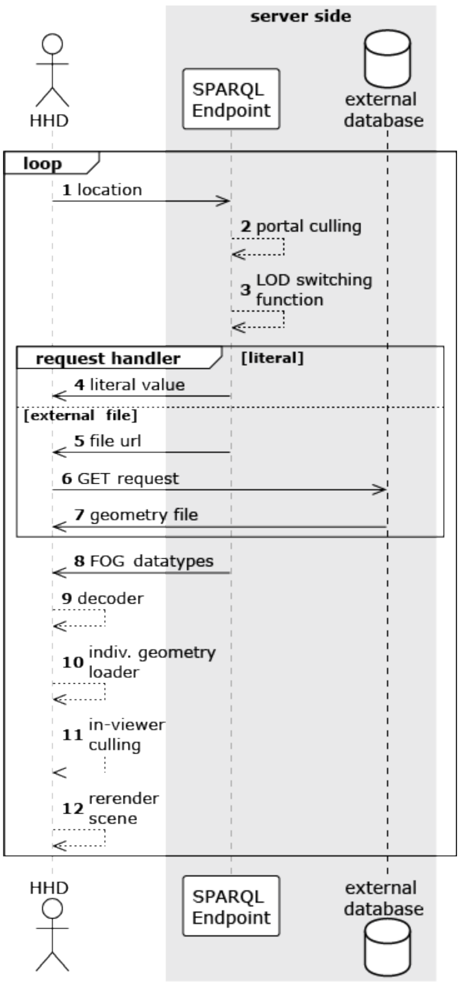

Figure 4.1: Sequence diagram

4.2 Framework

4.2.1 Nextjs

4.3 Querying

4.3.1 Front-end

4.3.2 Back-end

4.4 Rendering

4.4.1 Xeokit SDK

List of Acronyms

AEC	Architecture, Engineering and Construction	6
BIM	Building Information Modelling	6
BOT	Building Topology Ontology	
FOG	File Ontology for Geometry formats	
GIS	Geographic Information System	
HHD	Hand Held Device	8
LDBIM	Linked Data BIM	6
LOD	Level of Detail	10
OMG	Object Management Group	
SDK	Software Development Kit	
W3C	World Wide Web Consortium	8

References

Cohen-Or, D., Chrysanthou, Y. L., Silva, C. T., & Durand, F. (2003). A survey of visibility for walkthrough applications. *IEEE Transactions on Visualization and Computer Graphics*, *9*, 412–431. https://doi.org/10.1109/TVCG.2003.1207447

Johansson, M., Roupé, M., & Bosch-Sijtsema, P. (2015). Real-time visualization of building information models (bim). *Automation in Construction*, *54*, 69–82. https://doi.org/10.1016/J.AUTCON.2015.03.018

Johansson, M., Roupé, M., Johansson, M., & Roupé, M. (2009). *Efficient real-time rendering of building information models. vr for decision making view project virtual produktion planning with the help of bim and visualization, phase iii view project efficient real-time rendering of building information models*. https://www.researchgate.net/publication/220758081

Werbrouck, J. (2018). Linking data : Semantic enrichment of the existing building geometry. *Submitted in 2018*. https://lib.ugent.be/catalog/rug01:002494740

Referenced webistes

Data - w3c. (n.d.). https://www.w3.org/standards/semanticweb/data
Linked building data community group. (n.d.). https://www.w3.org/community/lbd/
Unity - manual: Memory in unity webgl. (n.d.). https://docs.unity3d.com/2023.2/Documentation/Manual/webgl-memory.html