DYNAMIC 3D VISUALIZATION OF FLOODS: CASE OF THE NETHERLANDS

K. Kumar1*, H. Ledoux2*, J. Stoter3

1 3D Geoinformation, Delft University of Technology, Delft, The Netherlands - (k.kavisha, h.ledoux, j.e.stoter)@tudelft.nl
2 Commission VI, WG VI/4

KEY WORDS: Floods, 3D visualization, CityJSON, Cesium 3D webglobe

ABSTRACT:

In this paper, we review state of the art 3D visualization technologies for floods and we focus on the Netherlands since it has a long history of dealing with floods and developing information technology solutions to prevent and predict them. We discuss the most recent advances in using 3D city models for modelling floods and discuss future directions. We argue that 3D city models provide a more realistic interpretation and assessment of floods e.g. information about the height of the water level and the number of floors that will be flooded. We present our framework to dynamically visualize floods in 3D using the Cesium 3D webglobe. An open platform using 3D city models for interactive visualization of different flood simulations can serve as a hub to involve all relevant parties such as water experts, policy developers, decision makers, and public. We created a 3D terrain model with buildings of a study area in the Netherlands in CityJSON format. We implemented a software prototypes for converting 3D city models from CityJSON to Cesium specific gTIF format for rendering over Cesium. We propose using CZML (Cesium Language) to represent time dynamic properties, water levels in our case. The developed framework which uses only open data and open-source software can be supportive in real applications such as planning for a city or municipal corporation, or for decision making.

1. INTRODUCTION

Flooding is globally one of the most dangerous and widespread natural hazards in low lying regions. The Netherlands is one such country that has a long history of dealing with floods. About half of the country is below sea level and the majority of its inhabitants live there. Flood risk arises both from surges along the coast and also from extreme precipitation events. The Dutch approach in dealing with flood risks has been to build an extensive network of artificial flood defenses such as dykes, dams, and levees. Apart from building physical structures, there is also the need to develop tools that quickly and clearly inform people about the potential flood hazards. The European Union Floods Directive (The European Parliament and the Council of the European Union, 2007) requires the member states to assess the flood risk in their coastal zones in the form of flood hazard and risk maps. However, as Kovanen et al. (2018) pointed out, these maps are static in nature. Recent developments in 3D GIS technologies provide better opportunities to improve the management, analysis, and visualization of flood-related data within a dedicated 3D spatial framework. 3D visualization techniques can help decision makers and stakeholders predict the consequences of floods and make informed decisions. National European mapping agencies such as in Finland (Linturi and Simula, 2003), Germany (Over et al., 2010) and the Netherlands (Elberink et al., 2013) have created their 3D city models for use in different applications such as urban planning, noise mapping and flood management. 3D visualization is more realistic and detailed, and helps the user to better understand the seriousness of the flooding. Traditionally, the 3D GIS visualization scenes were distributed as exported movie animations. Recent advancements in web technologies have allowed these 3D scenes to be shared over web. A number of standards exists for rendering 3D scenes/models in a web browser such as VRML1, KML2/COLLADA3, X3D4, and gTIF5.

In this paper, we review existing 3D visualization standards and technologies for visualizing floods (Section 2.). We discuss the most recent advances in using 3D city models for visualizing floods and discuss future directions. In Section 3., we present our framework system to dynamically visualize floods in 3D using the Cesium 3D webglobe. In order to investigate the applicability in practice, we illustrate the use of our system for real-world data in a case study for an area in Delft, The Netherlands (Section 4.). We created the 3D terrain model with buildings of the study area in CityJSON format. CityJSON is the JSON-based encoding of the CityGML standard (see Section 3.1 for explanation). We compared the two standards, that is CityJSON & Cesium specific gTIF standard, and implemented a software prototypes for converting 3D city models from CityJSON to gTIF for rendering on Cesium (Section 4.). This paper has two main contributions: (1) a working system for flood visualization with dynamic properties and (2) technical contributions required to realize a working solution. The paper provides an overview of our proposed framework and its possibilities.

2. DEVELOPMENTS IN THE GEOSPATIAL DOMAIN

2.1 Popular web-based 3D visualization formats

A multitude of open 3D formats exists for visualizing 3D data in a web browser. The oldest and most widespread format for modelling dynamic and interactive 3D scenes is VRML (Virtual Reality Modelling Language). It was accepted as the first web based 3D standard by web 3D consortium back in 1995, and is still a widely supported format by popular tools such as FME, and

---

*Corresponding author

https://www.khronos.org/glTF/

1https://www.v3.org/MarkUp/VRML/
2https://developers.google.com/kml/
3https://www.khronos.org/collada/
4http://www.web3d.org/x3d/what-x3d
5https://www.khronos.org/glTF/
2.2.3 Unity3D engine

Unity3D is a game engine developed by Unity Technologies to build and deploy interactive 2D & 3D video games. It is also used to create high quality architectural visualization and 3D real-time animations. Reyes and Chen (2017) used Unity3D to simulate storm surge flooding. The Unity Terrain Engine was used to create the terrain model of the study area. The 3D buildings and other city features were generated from the OSM (Open Street Maps) data. Both terrain and the city features were put together in the Unity environment with built-in components to simulate the ocean and surging waves. Wu et al. (2012) realized web-based 3D visualization and analysis of dam safety monitoring based on Unity3D engine.

3. OUR FRAMEWORK FOR VISUALIZING FLOODS

3.1 Overview of data models and Cesium 3D webglobe

3.1.1 CityJSON

CityJSON\(^7\) is a format for encoding a subset of the CityGML data model using JSON (JavaScript Object Notation). CityGML (City Geography Markup Language) is an open 3D data model for representing 3D city models (Open Geospatial Consortium, 2012). Its implementation is based on GML (Geography Markup Language). CityJSON offers an alternative to the GML encoding of CityGML. A CityJSON file represents both the geometry, appearance (textures/materials) and the semantics of the city features of a given area, e.g. buildings, roads, rivers, vegetation, and bridges (see CityJSON snippet below) and is around ~8X compact than CityGML (CityJSON wiki, 2018). CityJSON also allows us to store ISO 19115 metadata associated with 3D city models such as the coordinate reference system used, the extent of the dataset and its author. CityJSON supports both the 5 LODs (Levels Of Detail) of CityGML and the improved LOD specifications for representing buildings in CityGML (Biljecki et al., 2016). CityJSON also allows extending its data model with new attributes and objects for different applications. We used CityJSON in this paper because it is simple to create, manipulate, and edit JSON-based 3D city models and we can readily convert them to CityGML format and vice-versa using available libraries such as citygml4j\(^8\).

```json
{
  "type": "CityJSON",
  "version": "0.7",
  "metadata": {},
  "CityObjects": {
    "id": 1,
    "type": "Building",
    "attributes": {
      "roofType": "gable",
      "geometry": {
        "type": "Solid",
        "lod": 2.1,
        "boundaries": [...]}
    }
  }
}
```

3.1.2 glTF

glTF provides compact representation of geometries, small file sizes, and less processing time for unpacking and using the models. A glTF asset is represented by a JSON based scene description file and a binary (.bin) file containing the geometry (vertices, indexes and/or animation data) of the asset. glTF supports all the basic elements for representing 3D scenes such as textures, materials, animations, cameras, and lights. The glTF format also allows extensions to add new functionality to

---

\(^7\)http://www.cityjson.org
\(^8\)https://github.com/citygml4j/citygml4j

This contribution has been peer-reviewed.

https://doi.org/10.5194/isprs-archives-XLII-4-W10-83-2018 © Authors 2018. CC BY 4.0 License.
the schema or to simply define commonly used properties. glTF does not have the concept of LODs in the original schema. However, the ‘MSFT_lod’ extension, developed by Microsoft, adds the functionality to specify LODs in a glTF dataset.

### 3.1.3 Comparison and challenges

A comparison of the two formats is given in Table 1. We elaborate here on the main differences that are relevant for the conversion of CityJSON to glTF:

- CityJSON supports the GML geometry types such as polygons, triangles, solids, composite solids, etc. whereas glTF geometry is purely triangles, which cannot be structured to complex geometric primitives such as composite solids.

- CityJSON has support for semantics of city objects while glTF lacks semantics.

- glTF itself does not support geographic coordinates. All vertices are provided in a local coordinate system as single precision 32-bit floating point numbers (Schilling et al., 2016).

### 3.1.4 Cesium & CZML

Cesium is an open-source JavaScript library to create 3D virtual globes as well as 2D maps in a web browser (Cesium, 2018). It includes extensive libraries which support 2D as well as 3D geometries. It has support for visualizing 3D models in industry standard formats like KML, glTF, GeoJSON, TopoJSON. It can integrate layer imageries from different sources, including WMS, TMS, WMTS, Bing Maps, Mapbox, Google Earth Enterprise, OpenStreetMap, ArcGIS MapServer, standard image files, and custom tiling schemes. It is most suitable for time-dynamic geospatial data visualization using CZML. CZML (Cesium Language) is a JSON-based format for describing a time-dynamic data, primarily for display in a web browser running Cesium (CZML, 2016). A CZML document has a single JSON array where each element in the array is a ‘CZML Packet’ which describes the graphical properties for a single object such as a Building (CZML, 2016) (see CZML snippet below). Like glTF, CZML can be extended with custom properties.

```json
[  
  // packet one  
  {  
    "id": "Building-1",  
    "position": { "cartographicDegrees": [-75.5, 40.0, 0.0] },  
    "model": {  
      "gltf": "building.gltf",  
      "scale": 2.0,  
      "minimumPixelSize": 128  
    }  
  }  
]
```

19https://github.com/KhronosGroup/glTF/tree/master/extensions/2.0/Vendor/MSFT_lod

### 3.2 Proposed framework

Figure 1 depicts our proposed framework for visualization flood using Cesium. The details of the implementation of this framework are explained in Section 4.

#### 4. IMPLEMENTATION & DISCUSSION

### 4.1 Study area and datasets

The study area is the city of Delft, The Netherlands. BGT (Basisregistratie Grootschalige Topografie) and AHN3 (Actueel Hoogtebe-stand Nederland v3) point cloud are used as input for generating the terrain model of the study area. BGT contains topographic data i.e. all the buildings, roads, railroads, rivers, lakes and canals, as well as other landscape elements like forests and most other artificial structures. AHN3 is the the national height model of the Netherlands and has point density >10 points/m². We used BGT and AHN3 to obtain the footprint and height of the topographic features, respectively.
4.2 Implementation

The open source 3D modelling software, 3dfier\(^{20}\) is used to generate the TIN (Triangulated Irregular Network) terrain model (Figure 2) of the study area in CityJSON format. We also generated a separate CityJSON dataset with the 3D model of the buildings of the study area.

![Figure 2: 3D TIN terrain model of the study area](image)

Later on, this terrain model is used to run hydrological models. We used ANUGA for generating the time series water depth data of the study area. ANUGA\(^{21}\) is an open source hydrodynamic modelling tool developed by the Geoscience Australia and the Australian National University to help simulate the impact of natural hazards such as flooding, storm surges and tsunami (Roberts et al., 2015). ANUGA takes a triangular mesh of the study area as input data along with values for frictional resistance, initial values for water level, boundary conditions and operators such as rainfall, stream flows, windstress or pressure gradients if applicable. Data for frictional resistance and initial values for water level are provided by the Dutch company Nelen & Schuurmans\(^{22}\).

A hydrological simulation is run on the terrain model with the aforementioned data. This is obviously not an actual simulation of a way how a real world hydrological simulation works, but is good enough for our visualization purpose. The simulation generated a series of time varying water depth maps generated by the simulation. These water depth values are exported to CZML for visualization over Cesium.

![Figure 3: Cesium visualization of flood](image)

We implemented a python prototype to convert CityJSON datasets to glTF format for rendering over Cesium. As the position of buildings won’t change, this is a static layer. Therefore, we exported the dataset directly to glTF using the developed prototype. Figure 3 depicts the final visualization of a flood over Cesium.

5. CONCLUSION & FUTURE WORK

Several countries especially those lying in low elevation coastal zones are vulnerable to flood risks from rising sea levels and changing rainfall pattern. Lack of public awareness and motivation among people to support changes in coastal landscape or to even protest necessary measures to improve safety against floods. Simulating floods and visualizing their effects in 3D can prove useful for water experts, policy developers and decision makers to assess the seriousness of the situation and take decisions accordingly. Survey conducted by Leskens et al. (2017) showed that 3D visualisations have an enormous added value over 2D. In this paper, we reviewed the existing web-based 3D standards and technologies for flood visualization, and presented our open-source framework to dynamically visualize floods in 3D. An open platform using 3D city models for interactive visualization of different flood simulations can serve as a key to involve all interested parties. For simplicity, we used only buildings and terrain model for visualization in this research. In future, we aim to extend our approach to include other city features in the visualization and add support to do simple online query analysis on the model, e.g. information about the buildings inundated by flood water at a particular level. We also plan to utilize 3D tiling schemes to stream massive 3D datasets over Cesium webglobe. The research is still a work-in-progress but can be of great relevance to visualize the impact of disasters like flood in 3D.

ACKNOWLEDGEMENTS

This project (Project# 13740) is supported by the Netherlands Organisation for Scientific Research (NWO), and which is partly funded by the Ministry of Economic Affairs. Special thanks to Nlen & Schurmans, The Netherlands for providing data inputs for flood modelling.

References


\(^{20}\)https://github.com/tudelft3d/3dfier
\(^{21}\)https://github.com/GeoscienceAustralia/anuga_core
\(^{22}\)https://www.nelen-schuurmans.nl/en/