Nordic Journal of Surveying and Real Estate Research 13:1 (2018) 54–76
submitted on 13 December 2017
revised on 7 August 2018
accepted on 22 November 2018

Browser Based 3D for the Built Environment

Juho-Pekka Virtanen[a, d], **Hannu Hyyppä**[a], **Matti Kurkela**[a], **Matti T Vaaja**[a], **Tuulia Puustinen**[a], **Kaisa Jaalama**[a], **Arttu Julin**[a], **Matti Pouke**[b, c], **Antero Kukko**[d, a], **Tuomas Turppa**[d], **Lingli Zhu**[d], **Timo Ojala**[b], **Juha Hyyppä**[d]
[a] Aalto University School of Engineering, Department of Built Environment
[b] University of Oulu, Center for Ubiquitous Computing
[c] VTT Technical Research Centre of Finland
[d] Finnish Geospatial Research Institute FGI
Contact: `juho-pekka.virtanen@aalto.fi`

Abstract. *Digital 3D models have become a central tool for geo-information. For many participatory and collaborative applications, distributing these models easily is essential. Several technical solutions exist for creating online systems that facilitate the study of 3D models in the context of the built environment. First, we provide an overview on browser based interactive 3D visualizations through presenting a set of existing systems applied in Finland, and discussing their common properties and differences. Second, we experiment with an online 3D application development platform in order to obtain first-hand experience. The systems studied show a high potential for browser based 3D applications: interactive visualizations with multi-user characteristics and dynamic elements can be built by leveraging the 3D web technologies. Finally, we suggest a list of essential capabilities for online 3D visualization, covering the spectrum of possibilities available in modern web-based 3D.*

Keywords: *3D, game engine, online, virtual, web tools*

1 Introduction

Digital 3D models have become a central tool for geo-information, being built to a wide range of cases in varying extents, structure and style. In the context of the built environment, 3D city models, combining geometric and semantic information, are a significant research theme (e.g. Gröger & Plümer, 2012; Kolbe, Gröger & Plümer, 2005). In Architecture, Engineering and Construction (AEC), the focus of modeling has often been an individual building or a site, rather than the whole urban area (e.g. Sulankivi, Mäkelä & Kiviniemi, 2009; Arayici et al., 2011). Outside urban environments, 3D models have been used in geomorphology (Alho et al., 2011), maritime safety (Goralski et al., 2011), and landscape planning (Wack & Stelzl, 2005), to name but a few examples.

One of the central use cases for 3D models is visualization. Several authors have discussed this in 3D city models (Biljecki et al., 2015), virtual geographic

https://doi.org.0.30672/njsr.67846

environments (Gong & Lin, 1999; Lin et al., 2013) and 3D GIS (Liu et al., 2016; Xiao, 2001; Zhu, 2004). In their review of over 100 applications for 3D city models, Biljecki et al. (2015) identified only five use cases not involving visualization, whereas it was a significant part of 24 use cases. In planning, visualizations help communicate the construction project being planned for stakeholders of the project (Bouchlaghem et al., 2005) and assist public participation and communication between the stakeholders (Wu et al., 2010). In their review of the benefits of Building Information Models (BIM) for various stakeholders in AEC, Azhar et al., (2012) list visualization as a benefit for all stakeholder categories (owners, designers, builders and facility managers). Compared to traditional representations (such as maps or 2D drawings), realistic renderings produced from 3D models can improve communication (Lewis & Sheppard, 2006; Black et al., 2007).

With digital tools, either interactive or static visualizations can be produced (van Schaik, 2010). Interactive ones allow the user to freely move within the virtual space (Yan et al., 2011; Berry et al., 2011), and may include other aspects (such as acoustics) in addition to visual appearance (Manyoky et al., 2014). The benefits of interactive visualization environments include the possibility of freely choosing the viewpoint (van Schaik, 2010) and being able to study the model at any scale (Wu et al., 2010).

In the professional domain, creating, manipulating and studying 3D models has commonly been accomplished with commercial software tools, depending on the discipline (e.g. CAD suites, GIS software, 3D visualization software). For a landscape visualization example, see Grêt-Regamey et al. (2013). However, these tools are often not feasible for distributing the models to many stakeholders coming from outside the professional domain, as they may be difficult to use and are often expensive to obtain. A partial solution to this has been the introduction of open source software (e.g. QGIS, 2018; Blender, 2018) and commercial tools offered for free non-professional use, embracing usability (e.g. Trimble, 2018b). The emergence of widely supported 3D file formats that can be transferred between different software such as OBJ, COLLADA, IFC (ISO, 2018) and CityGML (OGC, 2012) has further supported this. Nevertheless, file based storage and transfer, and local software installations limit the usability of these tools for model distribution.

Owing to the aforementioned restrictions, the World Wide Web (www) has become the de facto platform for distributing 3D content to large user groups in many application domains (Evans et al., 2014). This has been enabled by the development of WebGL technology (Khronos Group, 2018b), allowing plugin-free 3D visualization in browser. While online 3D systems have existed for over 20 years (see, for example, Brown (1999)), the WebGL technology has greatly accelerated their development and uptake. The potential offered by browser based 3D has been acknowledged in research, too (e.g., for city modelling applications, see Gaillard et al., 2015). For this purpose, software vendors have released browser based viewers (e.g., Esri, 2018a; Autodesk, 2018). In addition, there are specific products available for collaboration and communication, such as online participative urban planning (Agency9, 2018a) or collaboration in AEC industry (Trimble, 2018a). These products offer discipline specific tools for browser based collaboration and communication. In some cases, the entire software titles have been released as browser based applications (Trimble, 2018b; Esri, 2018b). With these tools, the models produced using commercial software can be distributed for viewing, commenting, or even editing online.

For use cases requiring customized functionality, software development is needed. For example, Rönneberg et al., (2014) presented the development of a touchscreen-based system for studying carthographic data. Alatalo et al. (2016) presented a WebGL application for multi-user interaction in a 3D city model. In both cases, the models or maps are prepared with professional tools, after which a separate application is developed for facilitating the specific use case. For developing browser based applications that utilize 3D models, several alternative solutions are available.

Game engines are well suited for creating interactive virtual environments, offering the tools for utilizing advanced 3D graphics, sounds, interaction etc. (Trenholme & Smith, 2008). As high fidelity interactive systems tend to be resource intensive, many of the commercial game engines (e.g. Unity, Unreal Engine) are still geared towards the development of standalone software. However, commercial game engines increasingly offer tools for porting the developed applications to JavaScript for publishing in a web browser (to obtain an indoor model example, see Virtanen et al., 2018). There are also fully browser based game engines available for the development of WebGL applications (e.g., babylon.JS, 2018). Dedicated file formats for transferring 3D data to online applications have also emerged, such as glTF (Khronos Group, 2018a) and GeoJSON (Butler et al., 2016). 3D game engines have been extensively used together with geospatial data sets and building models (e.g., Manyoky et al., 2014; Mól et al., 2008; Virtanen et al., 2018). A common limitation in using game engines with geo-information data sets is their lack of support for interfaces such as the Web map service, WMS (OGC, 2018a) and geographic coordinate systems. Dedicated software tools allowing the utilization of geospatial data sets in game engines have emerged, partially solving this issue (Mapbox, 2018; Google, 2018).

Through combining WebGL technology with discipline specific features, several platforms that allow custom application development have emerged. These include MAPGETS (FCG City Portal Oy, 2018), targeted for 3D city model applications and ViziCities (2016) for 3D geovisualization. These platforms have varying characteristics, ranging from a commercial service with a dedicated hosting and application store (FCG City Portal Oy, 2018) to a JavaScript framework for 3D geospatial visualization (ViziCities, 2016). In addition, some broader development toolkits also offer features for developing WebGL applications (e.g. OpenBIM, 2012). While Google Earth is perhaps the most well-known virtual globe available, (for an application example, see Johansson et al., 2016) there are several alternatives available, ranging from commercial products to open source projects (Keysers, 2015; Brovelli et al., 2013). CesiumJS is one of the open source libraries for generating online virtual globes (Cesium consortium, 2018) that can be leveraged for developing interactive applications for studying 3D models, and this has seen adoption in a number of applications (e.g. Müller et al., 2016; Li & Wang, 2017). The development of online systems displaying large data sets has also stimulated the research and development of file format specifications, especially glTF and solutions to transfer large model assets by streaming (Schilling et al., 2016).

To summarize, there are several tools for creating browser based systems that support visualization of 3D models. These tools range from model viewer utilities to game engines, where sophisticated interactive applications can be built. While case descriptions for applying individual platforms are available (e.g., Manyoky et al., 2014; Lewis & Sheppard, 2006; Wu et al., 2010), and comparisons within a category have been published (e.g. virtual globes, Keysers, 2015), tools that allow comparison of applications achieved with these differing systems are largely absent. This is particularly the case if they are to be used discussing applications created with radically different systems (e.g. a game engine and a virtual globe). Some directions for achieving this can be found from the literature discussing different visualizations: Lammarsch et al. (2008) present a comparison of different platforms for browser based interactive visualization, but do not include 3D data. Kobsa (2001) utilizes environmental data in empirical user tests of three different information visualization systems, but does not apply this to 3D geometric models. Keim et al., (2004) focus on geovisualization, but do not discuss 3D models either. A taxonomy by Chick et al. (2003) includes eight factors for classifying information visualizations, these being dimension (1D, 2D, etc.), representation (e.g. symbolic or photo-realistic), display format (e.g. chart, picture or virtual world), scale (e.g. linear, logarithmic), planar geometric projection (e.g. orthogonal), temporal dimension (none, discrete or continuous), time mode (fixed image, video, etc.) and interaction (e.g. navigation or interaction with model). While this taxonomy contains several applicable aspects, it is not completely suited for discussing interactive 3D visualizations as it aims to cover a broader scope.

3D models are increasingly applied in the context of the built environment, both as 3D city models (Biljecki et al., 2015) and the development of applications for BIM models (Azhar et al., 2015). The emergence of use cases for online 3D visualization, such as participatory planning and digital permit process, makes it increasingly important to be able to discuss the differences of systems that allow the development of these online 3D applications. This is relevant, as differing technical solutions have been applied for interactive 3D visualization in the Finnish 3D city modeling ecosystem, leading to technical fragmentation (Julin et al., 2018). Our aim is to provide an overview on the properties of browser based interactive 3D visualizations in the context of the built environment.

We present a set of existing systems applied in Finland, and discuss their common properties and differences. To obtain first-hand experience, we experiment with an online 3D application development platform in visualizing two different 3D data sets. Finally, we establish a list of essential capabilities for online 3D visualization platforms as a means of discussing these systems.

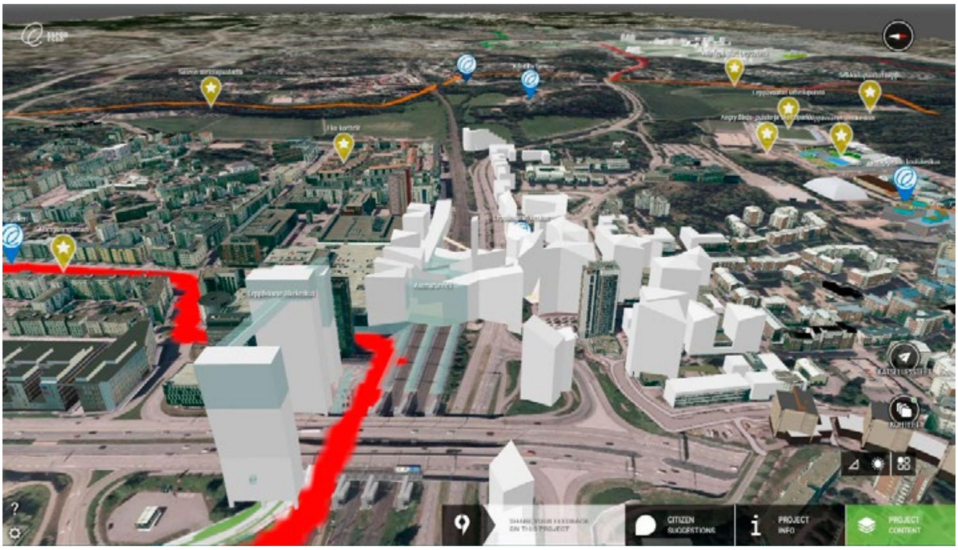

Figure 1. Online visualization of Leppävaara area showing the existing cityscape (textured models), proposed developments (white, non-textured buildings). The icons on map denote information points for existing city elements (yellow icon) and planned elements (blue icon). Image courtesy of the City of Espoo.

2 Browser based 3D systems in Finland

For this study, we selected five contemporary examples of browser based 3D systems applied in the context of the built environment and 3D city models. The case selection was based on finding cases from Finland which utilize a variety of platforms, and **was** limited to the systems that are openly available online.

2.1 Mission in Leppävaara

Tehtävä Leppävaarassa (Mission in Leppävaara), produced by the City of Espoo (Espoo, 2016), is a participative planning project realized using the Agency9 City Planner platform (Agency9, 2018b). The system allows the users to view the 3D model of the existing Leppävaara area (Espoo, Finland) and a number of models representing possible development projects, along with additional text and image material concerning them (Figure 1). Users can also add suggestions by marking an area from the 3D model, and see suggestions added by other users.

2.2 Otaniemi Lighting Simulator

The Otaniemi area lighting simulation (SitoWise, 2014), built using the Unity game engine, allows the user to explore the development project of Otaniemi campus in Espoo under different lighting conditions. The simulation allows the user to build different street lighting setups, and visualize them in the 3D model. It is possible to explore the model with different types of cameras, including a pedestrian avatar, a free flying camera and a camera offering a viewpoint from aboard a planned tram route. Real weather and time information is retrieved from a server and applied to the visualization. A street level view in daylight conditions is shown in Figure 2.

Figure 2. *Plans for the Otaniemi area visualized from the street level in Unity game engine. Image courtesy of SitoWise.*

2.3 Helsinki 3D+

The city of Helsinki has released several different 3D models, with two of them depicting the contemporary environment: a textured mesh model (Helsinki, 2018b) and a city information model (Helsinki, 2018a). Both of these models are viewable via browser based 3D tools and are offered for download (Helsinki Region Infoshare, 2016).

The Helsinki city information model is built utilizing a combination of airborne laser scanning, oblique aerial imaging (for textures) and existing registery information on buildings (Helsinki, 2018c). The model is maintained using technology from virtualcitySYSTEMS GmbH (virtualcitySYSTEMS, 2018a), featuring a Cesium based (Cesium consortium, 2018) online viewer (virtualcitySYSTEMS, 2018b) that allows the user to study the model online, access building information and perform queries using it, and download segments of the model in a set of commonly applied 3D file formats (Figure 3.)

The textured mesh model has been created using oblique aerial photos (Helsinki, How were the 3D models made? 2018; Bentley, 2018a), using, among other tools, the Bentley Context Capture software (Bentley, 2018b). A Cesium based viewer is provided to the model (Helsinki, 2018b), allowing the user to view the model in a browser (Figure 4).

2.4 Virtual Oulu

Built using the open source realXtend Tundra technology, the Virtual Oulu (Oulu 3D, 2018) offers a near photorealistic representation of the Oulu city center, allowing immersive street level navigation (Figure 5). The entity-component

Figure 3. *A view of Helsinki center in the online viewer of the Helsinki city information model. By clicking on buildings, the user can open the corresponding registry information displayed in an overlaid dialogue. The queried building is shown in red highlight. Image courtesy of the City of Helsinki.*

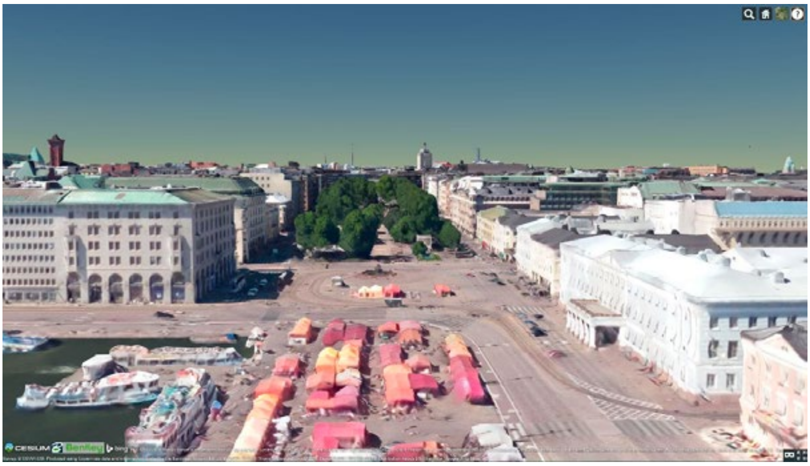

Figure 4. *In a view corresponding with Figure 3, the differences between the mesh model and city information model can be observed: the mesh model contains more geometric detail and includes large vegetation. However, no linking to registry information or discernible building objects are present. Image courtesy of the City of Helsinki.*

architecture of realXtend is presented by Alatalo (2011). RealXtend technology has been used to develop educational applications (Mattila et al., 2012), 3D cartographic applications (Virtanen et al., 2015a) and virtual city applications (e.g. Alatalo et al., 2016).

Figure 5. *Street level view from Virtual Oulu.*
Image courtesy of Oulu 3D & the City of Oulu.

Figure 6. *Hiukkavaara area development project plans visualized in MAPGETS. Image courtesy of FCG City Portal & the City of Oulu.*

2.5 SmartOulu

MAPGETS is a commercialized application development platform for city models (FCG City Portal Oy, 2018). MAPGETS accesses the Open Street Map (OSM) and city data through Web feature service (WFS) interface (OGC, 2018b), constructing the 3D city model automatically. Other data sources available over WFS, WMS or GeoJSON interfaces can be visualized on top of the model. MAPGETS can be used to develop and publish applications that utilize the 3D city model.

Built using the MAPGETS CityInfo application, the SmartOulu system allows the user to view a 3D model of the Oulu city, and view additional text, image and linked data concerning future city development (SmartOulu, 2018). In addition, the user can study the 3D models of future development projects, such as the Hiukkavaara area shown in Figure 6.

3 Application development platform for creating interactive 3D visualizations

To obtain firsthand experience in producing interactive visualizations of 3D models, we utilized an online application development platform for creating two different online scenes.

Meshmoon, a commercial development platform for virtual world applications, was applied to create the browser based 3D environment (Meshmoon, 2014a). Meshmoon combines commercial hosting services with a multi-user framework for creating online 3D environments that can also contain scripted additional functionality. For accessing Meshmoon scenes, either a standalone client or a browser based client can be used. In this case, we applied the browser based, plugin free Meshmoon Webrocket client. The Webrocket client is based on three.js (three.js, 2018) using WebGL. Meshmoon is based on realXtend technology, and has been applied for 3D map visualizations (Virtanen et al., 2015) as well as gamified applications utilizing indoor models (Alavesa et al., 2016).

Two different 3D models from the Keilaniemi area (Espoo, Finland) were used in the experiment. Keilaniemi is a predominantly commercial area, featuring several highrise office buildings. The area is undergoing a large transition with the construction of a new metro line, and residential development.

3.1 Keilaniemi area model

Open datasets provided by the National Land Survey of Finland were used for modeling the Keilaniemi area. The method described by Zhu et al. (2015) was applied for producing 3D models of buildings, road surfaces and terrain from the topographic database and the Airborne Laser Scanning (ALS) point cloud. An aerial orthoimage was used for texturing the Digital Terrain Model (DTM). The processing resulted in a set of 3D mesh objects retained in a geographic coordinate system (ETRS-TM35FIN).

The original building and road information contained in the topographic database was not kept with the generated models, apart from the object type (building, road, terrain) which was used to separate models into three respective categories. The models therefore contained no significant semantic information.

Two digital models were utilized to describe the residential tower development project in the area, and the new metro line. The model illustrating the metro line was produced with manual modeling in Blender, using a 2D map of the line as a reference. A 3D model in Trimble Sketchup format was utilized for the Keilaniemi Towers project. In addition, a set of 3D texts and arrows were created in Blender, illustrating the main directions of roads in the area.

3.2 Keilaniemi shore model

Manual game engine modeling methods were applied for producing a detailed 3D model of the Keilaniemi shore area. A set of commercial buildings were modeled using both ALS and Mobile Laser Scanning (MLS) data sets as a reference, using the results of automated building vectorization as the starting point for more detailed modeling.

The modeling was carried out manually in 3ds Max, following the conventions of 3D content production for the computer games industry. As the aim of the modeling process was to achieve a visually impressive, game engine compatible model, the geometric accuracy of the model was not verified. The model was constructed manually, aiming for a low polygon count. Texturing was extensively used, applying tileable (repeating) textures where possible to maintain small texture map sizes. Other bitmaps adding material properties such as reflectance, specularity and surface normal details were created. The modeling work required several weeks of working time from an experienced modeler. The scene was assembled from several manually crafted models in Unity 4 game engine.

The object structure of the model was determined by the modeler, following the geometric features of the environment. The model structure does not reflect real world object division (e.g. by keeping buildings as separate objects), nor do the models contain any semantic information.

3.3 Overview visualization

The 3D model of the Keilaniemi area described in 3.1 was applied to the creation of an interactive visualization of the area. The model was processed further utilizing the open source mesh modeling suite Blender. A coordinate transfer was applied to the model prior to importing, to enable operation at local, “short” coordinates. Having the model close to origin simplifies model editing (e.g. pivots) and helps maintain accuracy in systems that have limited floating point accuracy. During this stage, the digital terrain model was texturized with an aerial image, with orientation being carried out interactively. In addition, rendering materials were assigned to building and road objects.

The model of the planned construction project was imported to Blender and added to the scene. The position of the model was resolved by referencing the area plan. After this, the model was exported to a format compatible with Meshmoon Webrocket viewer (Ogre Mesh), and uploaded to the Meshmoon server. The server then converts the ogre meshes and materials into three.js format. A small application was developed to allow the user to show or hide the model describing the metro line or residential towers.

3.4 Pedestrian level visualization

The completed, assembled scene described in 3.2 was transferred from Unity to Meshmoon by converting it to a set of Meshmoon compatible objects using the Unity exporter plugin (Meshmoon, 2014b). The converted mesh models, material files and texture images were then uploaded to the Meshmoon server.

The avatar application (a pre-existing component of Meshmoon) was added to the scene. To make the model “walkable” by an avatar utilizing physics modeling, a set of geometric primitives had to be added. These are not rendered, but used by the physics engine to facilitate collision detection, providing the avatar with a surface to walk on. Figure 7 shows the model used for physics and the rendered model.

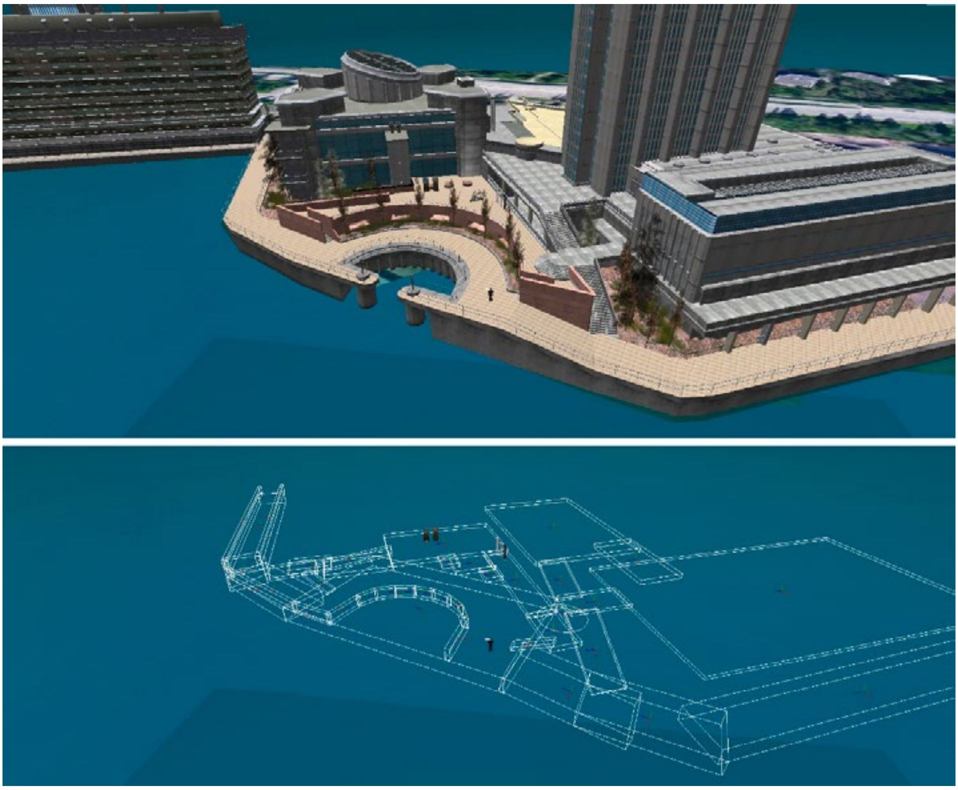

Figure 7. *The model visible to user (above), with the simplified model applied for physics modeling shown below from same viewpoint.*

4 Results

4.1 Keilaniemi visualizations

By using the described platform and processing methods, browser based interactive visualizations were created for both of the models depicting the Keilaniemi area (Figures 8 and 9). Google Chrome was used to access the visualizations. Even though operating on the same technical platfrom, the visualizations depict the environment on very different levels.

The overview visualization utilized an area model generated automatically from open geospatial data sets. It provides a simplistic visualization of the area, with simple 3D building and road surface models. Therefore, the detail level of the model is not sufficient for pedestrian level visualization. A rotatable camera application is provided for exploring the model. In addition, the overview visualization contains some thematic information not related to the geometry or visual appearance (Figure 8).

The pedestrian level visualization utilizes a manually created, significantly more detailed model is utilized. Instead of a moveable virtual camera, the applications provides a user with a 3D avatar for navigating the environment by

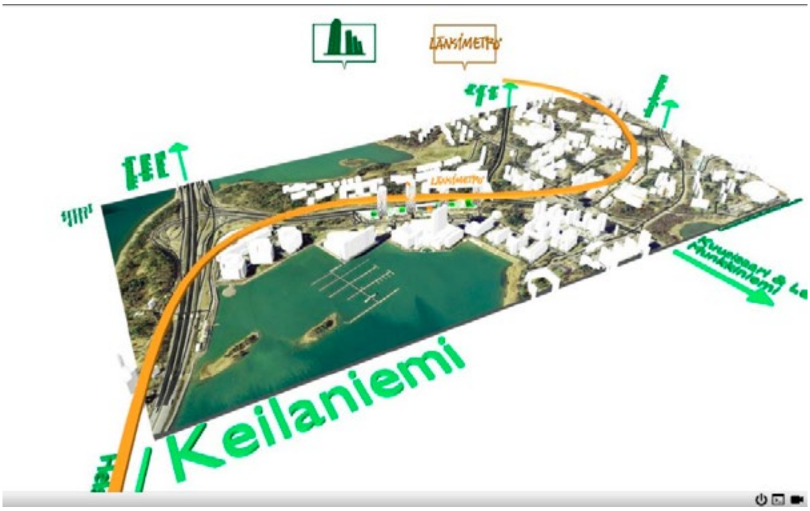

Figure 8. *The overview visualization of Keilaniemi area shown in browser.*

Figure 9. *User navigating the Keilaniemi shore model in the pedestrian level visualization.*

walking. The offered viewpoint is thus oriented around the pedestrian experience (Figure 9).

The technical properties for the visualizations (Table 1) reveal a number of differences. Firstly, the significance of texturing for model appearance: the amount of geometric data in the models is approximately the same, whereas the amount of texture data for the pedestrian level visualization is over tenfold, containing a considerable amount of texture images. Secondly, the number of objects is much higher in the pedestrian level visualization owing to a more detailed model

Table 1. *Technical properties of Keilaniemi visualizations,*

Case	**Overview visualization**	**Pedestrian level visualization**
Total polygons	328,261	340,989
Total vertices	984,783	1,022,967
Mesh data (MB)	17.2 MB	19.8 MB
Mesh objects	53	346
Texture data (MB)	0.9 MB	54.7 MB
Texture maps	24	84
Rendering materials	24	53
Total objects in scene	57	441
Draw calls	53	547

depicting the environment. Combined, these differences contribute to a radically different level of computational complexity observed in the number of draw calls raised. Rendering the detailed model requires exeedingly more memory for storing the textures, and computational power from the GPU to execute the drawing of the scene.

4.2 Essential capabilities for online 3D visualization platforms

Looking at the existing interactive 3D visualizations and the experiences obtained from the Keilaniemi cases, we can identify several differences between browser based interactive 3D visualizations. We utilized these observations to modify (and extend) the taxonomy of visualizations from Chick et al. (2003), resulting in a list of eight essential capabilities of online 3D visualizations (Table 2).

Coordinate space and extent. In the presented case, the visualized area ranged from a set of a few buildings in a local coordinate system (See 3.4, Keilaniemi pedestrian level visualization) to a city-wide visualization with global coordinates (Helsinki 3D+).

The extent of the visualization and coordinate space are heavily dependent on the platform applied: many of the visualizations with large areas have been realized utilizing virtual globes, and thus allow, for instance, the placing of objects in any global location. Typically, these systems also handle geographic coordinate systems. In comparison, game engine based visualizations typically only support limited scenes and local coordinates, due to, for example, limitations on floating point accuracy and user interfaces unsuited to manipulating large scenes. The extent of the system can naturally be expanded by dividing the scene into tiles, but this is cumbersome if the tiling features are not integrated to the visualization system.

For some applications, the visualization of limited areas is sufficient. However, larger areas are required in applications such as landscape planning (e.g., Manyoky et al., 2014; Lewis & Sheppard, 2006; Wissen et al., 2008).

Geometric complexity. The visualized models varied from representing buildings as very simplistic volumes (Mission in Leppävaara; Keilaniemi overview visualization) to the inclusion of detailed building models (Otaniemi; Keilaniemi

Table 2. *Essential capabilities for online 3D visualization.*

Factor	Constrained	Optimal
Coordinate space and extent	Limited extent, local coordinate system	Global extent, geographic coordinates supported
Geometric complexity	Model complexity limited	CityGML LOD 4
Texturing	No textures	Fully textured model
Non-geometric information	No object structure in model	Full semantic data
Temporal dimension	Only representing a fixed moment of time	Representing a continuum of time
Dynamic content	Static, pre-built model	Fully dynamic simulation possible
Interaction	Camera control only	Full model editing functionality
User count	Single user model	Multi-user environment

pedestrial level visualization). Naturally, there were differences in representing other urban elements as well: for example, the pedestrian level visualization from Keilaniemi contained a high amount of walkway and park structures modeled in high detail (Figure 9).

CityGML uses the concept of level of detail (LOD) to characterize models (Gröger & Plümer, 2012). While the majority of the discussion concerning the LOD has been around the representation of buildings (e.g. Benner et al., 2013; Biljecki et al., 2016), the LOD can be defined for other object classes as well (e.g. city furniture, see OGC, 2012). While the highest level of detail in CityGML includes interiors and furniture (OGC, 2012), models with higher detail are commonly encountered in gaming context. The concept of LOD is also present in game engines and mesh models commonly used in game engine content production, but without a single standardized definition (Luebke, 2003). This complicates the discussion on the geometric complexity of models.

The geometric complexity naturally depends on the data to be visualized, but also necessitates suitable rendering and loading performance from the platform used. Tiling schemes are commonly applied to manage large amounts of data in visualization.

Texturing. The Helsinki 3D+ city information model applies photographic textures to relatively simplistic building models, whereas in the Otaniemi Lighting Simulation, highly detailed building models are textured with procedural textures depicting different façade materials. The models can also be un-textured, such as some of the models in the SmartOulu application which only contain a set of fixed colors. For the Keilaniemi shore model, the texturing played a significant role in the appearance of the models.

Texturing is also dependent on the source data, and can be applied to geometric models of different levels of complexity. It also becomes a specific content creation task when material specific textures are required. However, texturing is also varyingly supported on different platforms, with game engines supporting multiple textures per object, and allowing the use of texture maps to also describe other appearance properties of surfaces, such as the reflectance of specularity.

Non-geometric information. For several applications, the objects' ability to contain non-geometric information is essential. This information may consist of semantic data and object specific metadata. For example, the city information model of Helsinki allows the user to access building registry information. The inclusion of non-geometric information extends the analysis potential of models, and allows their use for information visualization (Chaturvedi & Kolbe, 2015). However, both of these scenarios require the platform used to support inclusion of non-geometric information.

Temporal dimension. A visualization represents either a static state, a set of discrete states, or a continuous flow of time (Wenzel et al., 2003). This can range from the model displaying a static moment in time (Helsinki 3D+) to visualizing different times of the day (Otaniemi) or to objects having a position on a timeline (SmartOulu).

Dynamic content. In addition to the user, other elements in a browser based 3D application can also be dynamic. The Otaniemi Lighting Simulator includes some moving vehicles, and a lighting system visualizing different times of the day. For example, building and city models have been used to visualize energy consumption data from sensors (Kim et al., 2012). In this case, the model colors and bar heights are altered dynamically according to the data.

Unlike time series data, dynamic content may consist of real-time sensor observations, or programmed logic (Chaturvedi & Kolbe, 2015).

Game engines have been applied for creating extensive dynamic simulations (Wang et al., 2003). In addition, the dynamic properties have to be taken into account in the data (Chaturvedi & Kolbe, 2015). Realizing dynamic properties naturally places demands on the platform used.

Interaction. Functions in the systems allow the users to perform a variety of tasks, from simply studying the model and accessing information (Helsinki 3D+) to commenting (Mission in Leppävaara) or editing objects in the system (Otaniemi Lighting Simulation). Interaction functions can be realized by leveraging the tools offered by the platform (like in Keilaniemi shore), or by programming them manually (as in the Keilaniemi overview visualization).

A common interaction in planning applications is the comparison of different scenarios represented with different models (e.g., Wissen et al., 2008). The interaction tools and programming possibilities greatly depend on the platform used.

User count. The Virtual Oulu application can facilitate several users interacting, while model viewers are single user systems (Helsinki 3D+, Otaniemi Lighting Simulator). As a midway solution, many of the commenting and collaboration systems allow the users to leave comments, and view comments left by other users, but not interact directly (Mission in Leppävaara, SmartOulu).

The user count is dependent on the system used, with some systems natively offering support for multi-user interaction.

5 Discussion and conclusions

Recently, 3D models have become a central tool for professionals in the built environment. In a majority of applications, the visualization of these models is required. Digital systems are suited to producing interactive visualizations, that allow the user to study the models freely (unlike in images or videos). For online participatory activities, platforms that allow the development of customized functionalities are needed (Berry et al., 2011). Ideally, the platform used should be flexible enough to allow several different tasks to be accomplished with the same system (Döllner et al., 2006). If large focus groups are to be reached with these systems, they have to be browser-based and operate without any additional installations (e.g., plugins).

The creation of interactive visualizations may be performed using products intended for 3D model distribution, viewing and collaboration, or generic 3D game engines. In addition, there are emerging application development platforms that combine aspects from both game engines and collaboration environments.

While case descriptions of applying various technical systems are available in the literature, frameworks that allow the discussion of or the comparison of different setups are poorly available. By studying a set of contemporary Finnish examples of online 3D systems and the relevant literature, we suggest a list of essential capabilities for online 3D visualization in order to enable the discussion of online 3D visualization in the context of the built environment.

When discussing the different systems for browser based interactive 3D visualization, one of the challenges is the connection of the data sets visualized and the technical systems used. For example, when applying a game engine for visualizing simple 3D models, the rendering capability of the engine is much higher than what is required by the model. It also has to be noted, that some of the properties in the list are difficult to attain. For example, server infrastructure is needed for implementing the communication between users, or updating edited content to other users. However, this is not available for all platforms. Not only do these features require a platform with suitable capabilities, but also, in most cases, further software development. Choosing the right types of tools for the task, and identifying the features required becomes a central question when building interactive visualizations.

Looking at the platforms in more detail, a virtual world platform (Meshmoon) was applied for visualizing two different models from the Keilaniemi area. In the Otaniemi lighting simulation, we see the utilization of a game engine (Unity) for realizing a more complex application. These examples highlight the benefits of applying game engines: very different types of scenes can be built, as long as the models are available as mesh models in compatible format and polygon count. Visualizations of varying size, geometric complexity, dynamic content or even the user count become possible. Different types of camera controls, navigation methods and functionalities can be implemented as needed. Indeed, the high rendering capacity of a game engine is easily accessible. However, increasing the amount of interaction tends to increase the amount of work required: models have to be prepared for interaction functions (e.g., by adding elements for physics), and required additional components (camera control, etc.) have to be added, or developed from scratch.

The Mission in Leppävaara case applies a collaboration platform for city models. The implementation requires significantly less work than building a collaboration system from scratch using a game engine. At the same time, reaching the same amount of interaction as attained with a game engine would be very difficult. Specific platforms are not as flexible as game engines.

To summarize, the systems studied show a high potential for browser based 3D applications in the context of the built environment. Interactive visualizations with multi-user characteristics and dynamic elements can be built by leveraging the 3D web technologies. These aspects (e.g., high interaction, multi-user operation) that were earlier associated with stand-alone software (such as computer games), are now possible in a browser in a way that is conductive to high levels of distribution. While increasing the amount of interaction tends to increase the amount of case-specific programming work required, the emerging application development platforms may reduce the amount of software development needed. Professionals working with geo-information data should be aware of the different tools and their properties, and the whole spectrum of possibilities available in modern web-based 3D in order to be able to make the best choices.

Acknowledgements. The authors would like to thank the Finnish funding agency for the support of Varpu project (7031/31/2016). The authors are grateful to the Academy of Finland for their support of the Center of Excellence in Laser Scanning Research (CoE-LaSR; 272195) and the Strategic Research Council project "Competence Based Growth Through Integrated Disruptive Technologies of 3D Digitalization, Robotics, Geospatial Information and Image Processing/Computing – Point Cloud Ecosystem (293389/ 314312)". The European Regional Development Fund (ERFD) is acknowledged for the support of AKAI project (301130).

References

Agency9 (2018a), CityPlanner. Retrieved from: https://cityplanneronline.com/site/ [29.5.2018]

Agency9 (2018b), Dialogue project: Mission Leppävaara. Retrieved from: https://cityplanneronline.com/site/index.php/showcase/dialogue-project-mission-leppavaara/ [14.5.2018]

Alatalo, T. (2011). An entity-component model for extensible virtual worlds. Internet Computing, 15(5), 30–37.

Alatalo, T., Koskela, T., Pouke, M., Alavesa, P. & Ojala, T. (2016). VirtualOulu: collaborative, immersive and extensible 3D city model on the web. In Proceedings of the 21st International Conference on Web3D Technology. (pp. 95–103). ACM.

Alavesa, P., Samodelkin, A., Jääskelä, E., Tanskanen, R., Li, B., Ojala, T., & Kukka, H. (2016, June). Campus knights: situated pervasive display as a window into pseudo-immersive game world. In *Proceedings of the 5th ACM International Symposium on Pervasive Displays* (pp. 132–139). ACM.

Alho, P., Vaaja, M., Kukko, A., Kasvi, E., Kurkela, M., Hyyppä, J., Hyyppä, H. and Kaartinen, H. (2011). Mobile laser scanning in fluvial geomorphology: Mapping and change detection of point bars. Zeitschrift für Geomorphologie, Supplementary Issues, 55(2), 31–50.

Arayici, Y., Kiviniemi, A. O., Coates, S. P., Koskela, L. J., Kagioglou, M., Usher, C., & O'Reilly, K. (2011). BIM implementation and Adoption Process for an Architectural Practice.

Autodesk (2018), Autodesk Viewer. Retrieved from: https://viewer.autodesk.com/ [29.5.2018]

Azhar, S., Khalfan, M. & Maqsood, T. (2015). Building information modelling (BIM): now and beyond. Construction Economics and Building, 12(4), 15–28.

babylon.JS (2018) babylon.JS Retrieved from http://www.babylonjs.com [29.5.2018]

Benner, J., Geiger, A., Gröger, G., Häfele, K. H., & Löwner, M. O. (2013, November). Enhanced LOD concepts for virtual 3D city models. *In ISPRS annals of the photogrammetry, remote sensing and spatial information sciences. Proceedings of the ISPRS 8th 3D GeoInfo conference & WG II/2 workshop* (pp. 51–61).

Bentley, 2018a Retrieved from https://www.bentley.com/en/project-profiles/city-of-helsinki_helsinki-3d [14.5.2018]

Bentley, 2018b, Retrived from https://www.bentley.com/en/products/product-line/reality-modeling-software/contextcapture [14.5.2018]

Berry, R., Higgs, G., Fry, R. & Langford, M. (2011). Web-based GIS Approaches to Enhance Public Participation in Wind Farm Planning. Transactions in GIS, 15(2), 147–172.

Biljecki, F., Ledoux, H., & Stoter, J. (2016). An improved LOD specification for 3D building models. Computers, Environment and Urban Systems, 59, 25–37.

Biljecki, F., Stoter, J., Ledoux, H., Zlatanova, S., & Çöltekin, A. (2015). Applications of 3D City Models: State of the Art Review. ISPRS International Journal of Geo-Information, 4(4), 2842–2889.

Black, J., Arrowsmith, C., Black, M. & Cartwright, W. (2007). Comparison of techniques for visualising fire behaviour. Transactions in GIS, 11(4), 621–635.

Blender (2018), Blender. Retrieved from: https://www.blender.org/ [18.5.2018]

Bouchlaghem, D., Shang, H., Whyte, J. & Ganah, A. (2005). Visualisation in architecture, engineering and construction (AEC). Automation in construction, 14(3), 287–295.

Brovelli, M. A., Hogan, P., Minghini, M., & Zamboni, G. (2013). The power of Virtual Globes for valorising cultural heritage and enabling sustainable tourism: NASA World Wind Applications. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 4, W2.

Brown, I. (1999). Developing a virtual reality user interface (VRUI) for geographic information retrieval on the Internet. Transactions in GIS, 3(3), 207–220.

Butler, H., Daly, M., Doyle, A., Gillies, S., Hagen, S., & Schaub, T. (2016). *The geojson format* (No. RFC 7946).

Cesium Consortium (2018), CesiumJS, Retrieved from: https://cesiumjs.org [14.5.2018]

Chaturvedi, K., & Kolbe, T. H. (2015, November). Dynamizers-Modeling and Implementing Dynamic Properties for Semantic 3D City Models. In UDMV (pp. 43–48).

Day, A. (2002) Urban Visualization and Public Inquiries: The Case of the Heron Tower, London. Architectural Research Quarterly, 6, 363–372.

Döllner, J., Baumann, K. and Buchholz, H., 2006. Virtual 3D city models as foundation of complex urban information spaces.

Espoo (2016), Tehtävä Leppävaarassa. Retrieved from: http://www.espoo.fi/fi-FI/Espoon_kaupunki/Tehtava_Leppavaarassa(63395) [24.11.2017].

Esri (2018a), Esri CityEngine Tutorial 15: Publish web scenes. Retrieved from: http://desktop.arcgis.com/en/cityengine/latest/tutorials/tutorial-15-publish-web-scenes.htm [29.5.2018]

Esri (2018b), ArcGIS Online. Retrieved from: https://www.esri.com/en-us/arcgis/products/arcgis-online/overview [6.8.2018]

FCG City Portal Oy (2018), MAPGETS, Retrieved from: http://info.mapgets.com/en/ [18.5.2018]

Gaillard, J., Vienne, A., Baume, R., Pedrinis, F., Peytavie, A. and Gesquière, G. (2015). Urban data visualisation in a web browser. In Proceedings of the 20th International Conference on 3D Web Technology (pp. 81–88). ACM.

Google (2018), Google Maps Platform, Retrieved from: https://developers.google.com/maps/gaming/ [18.5.2018]

Goralski, R., Ray, C., & Gold, C. (2011). Applications and benefits for the development of cartographic 3D visualization systems in support of maritime safety. *International Recent Issues about ECDIS, e-Navigation and Safety at Sea: Marine Navigation and Safety of Sea Transportation*, 77.

Grêt-Regamey, A., Celio, E., Klein, T. M., & Hayek, U. W. (2013). Understanding ecosystem services trade-offs with interactive procedural modeling for sustainable urban planning. Landscape and Urban Planning, 109(1), 107–116.

Gröger, G. and Plümer, L. (2012). CityGML–Interoperable semantic 3D city models. ISPRS Journal of Photogrammetry and Remote Sensing, 71, 12–33.

Henry, P., Krainin, M., Herbst, E., Ren, X. & Fox, D. (2012). RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments. The International Journal of Robotics Research, 31(5), 647–663.

Helsinki Region Infoshare (2016), 3D Model of Helsinki, Retrieved from https://hri.fi/data/en_GB/dataset/helsingin-3d-kaupunkimalli [14.5.2018].

Helsinki (2018a), Helsinki city information model, Retrieved from: https://kartta.hel.fi/3d/ [14.5.2018]

Helsinki (2018b), Helsinki reality mesh, Retrieved from: https://kartta.hel.fi/3d/mesh/ [14.5.2018]

Helsinki (2018c), How were the 3D models made?, Retrieved from https://www.hel.fi/helsinki/en/administration/information/general/3d/how-were-the-3D-models-made/ [14.5.2018]

ISO, ISO 16739:2013, Retrieved from: https://www.iso.org/standard/51622.html [18.5.2018]

Jaakkola, A., Hyyppä, J., Hyyppä, H., & Kukko, A. (2008). Retrieval algorithms for road surface modelling using laser-based mobile mapping. Sensors, 8(9), 5238–5249.

Johansson, T., Vesterlund, M., Olofsson, T., & Dahl, J. (2016). Energy performance certificates and 3-dimensional city models as a means to reach national targets – A case study of the city of Kiruna. Energy Conversion and Management, 116, 42–57

Kaartinen, H., Hyyppä, J., Kukko, A., Jaakkola, A. & Hyyppä, H. (2012). Benchmarking the performance of mobile laser scanning systems using a permanent test field. Sensors, 12(9), 12814–12835.

Keysers, J. (2015). Review of digital globes 2015. Australia and New Zealand Cooperative Research Centre for Spatial Information.

Khronos Group (2018a), glTF, Retrieved from: https://github.com/KhronosGroup/glTF [18.5.2018]

Khronos Group (2018b). WebGL, Retrieved from https://www.khronos.org/webgl/ [18.5.2018]

Kim, S.A., Shin, D., Choe, Y., Seibert, T. & Walz, S.P. (2012). Integrated energy monitoring and visualization system for Smart Green City development: Designing a spatial information integrated energy monitoring model in the context of massive data management on a web based platform. Automation in Construction, 22, 51–59.

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

Kreylos, O., Bawden, G. W., & Kellogg, L. H. (2008). Immersive visualization and analysis of LiDAR data. In Advances in visual computing (pp. 846–855). Springer Berlin Heidelberg.

Kukko, A., Kaartinen, H., Hyyppä, J. & Chen, Y. (2012). Multiplatform mobile laser scanning: Usability and performance. Sensors, 12(9), 11712–11733.

Lewis, J.L. & Sheppard, S.R. (2006). Culture and communication: can landscape visualization improve forest management consultation with indigenous communities?. Landscape and Urban Planning, 77(3), 291–313.

Li, W., & Wang, S. (2017). PolarGlobe: A web-wide virtual globe system for visualizing multidimensional, time-varying, big climate data. International Journal of Geographical Information Science, 31(8), 1562–1582.

Lin, H., Chen, M., Lu, G., Zhu, Q., Gong, J., You, X., ... & Hu, M. (2013). Virtual geographic environments (VGEs): a new generation of geographic analysis tool. Earth-Science Reviews, 126, 74–84.

Liu, T., Zhao, D., & Pan, M. (2016). An approach to 3D model fusion in GIS systems and its application in a future ECDIS. Computers & Geosciences, 89, 12–20.

Luebke, D. P. (2003). *Level of detail for 3D graphics.* Morgan Kaufmann.

Manyoky, M., Wissen Hayek, U., Heutschi, K., Pieren, R. & Grêt-Regamey, A. (2014). Developing a GIS-based visual-acoustic 3D simulation for wind farm assessment. ISPRS International Journal of Geo-Information, 3(1), 29–48.

Mapbox (2018), mapbox, Retrieved from: https://www.mapbox.com/ [18.5.2018]

Mattila, P., Krajnak, J., Arhippainen, L., & Brauer, S. (2012). Education in 3D Virtual Learning Environment – Case Virtual-Mustiala. Paris, 26–27 November 2012 École Nationale Supérieure des Arts Décoratifs (Paris, France), 134.

Meshmoon (2014a), Meshmoon Documentation, Retrieved from: https://doc.meshmoon.com/index.html [16.5.2018]

Meshmoon (2014b), Meshmoon Unity Exporter, Retrieved from: https://doc.meshmoon.com/index.html?page=unity-plugin [16.5.2018]

Mól, A.C.A., Jorge, C.A.F. & Couto, P.M. (2008). Using a game engine for VR simulations in evacuation planning. IEEE computer graphics and applications, 28(3), 6–12.

Müller, R.D., Qin, X., Sandwell, D.T., Dutkiewicz, A., Williams, S.E., Flament, N., Maus, S. and Seton, M. (2016). The GPlates portal: cloud-based interactive 3D visualization of global geophysical and geological data in a web browser. PloS one, 11(3), p.e0150883.

Nebiker, S., Bleisch, S., & Christen, M. (2010). Rich point clouds in virtual globes – A new paradigm in city modeling?. Computers, Environment and Urban Systems, 34(6), 508–517.

Nüchter, A., Borrmann, D., Koch, P., Kühn, M. & May, S. (2015). A man-portable, IMU-free mobile mapping system. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences II-3 W, 5, pp.17–23.

OGC (2012), City Geography Markup Language (CityGML) Encoding Standard, Retrieved from: http://www.opengis.net/spec/citygml/2.0 [18.5.2018]

OGC (2018a), Web Map Service. Retrieved from: http://www.opengeospatial.org/standards/wms [18.5.2018]

OGC (2018b), Web Feature Service. Retrieved from: http://www.opengeospatial.org/standards/wfs [28.5.2018]

van Oosterom, P., Martinez-Rubi, O., Ivanova, M., Horhammer, M., Geringer, D., Ravada, S., Tijssen, T., Kodde, M. & Gonçalves, R. (2015). Massive point cloud data management: design, implementation and execution of a point cloud benchmark. Computers & Graphics, 49, 92–125.

OpenBIM (2012), OpenBIM The Open Toolbox for BIM, Retrieved from: http://www.openbim.org/ [18.5.2018]

Oulu 3D (2018),Virtual Oulu, Retrieved from: http://www.oulu3d.fi/virtual-oulu/ [29.5.2018].

Pu, S., & Vosselman, G. (2006). Automatic extraction of building features from terrestrial laser scanning. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 36(5), 25–27.

QGIS (2018), QGIS A Free and Open Source Geographic Information System. Retrieved from: https://qgis.org/en/site/ [18.5.2018]

Rusu, R. B., & Cousins, S. (2011). 3D is here: Point cloud library (pcl). In Proceedings of the IEEE Interational Conference on Robotics and Automation (ICRA) (pp. 1–4). IEEE.

Rönneberg, M., Sarjakoski, T. and Sarjakoski, L.T. (2014). Developing a Multi-Touch Map Application for a Large Screen in a Nature Centre. Nordic journal of surveying and real estate research, 10(1).

Shneiderman, B., (1996). The eyes have it: A task by data type taxonomy for information visualizations. In Proceedings of IEEE Symposium on Visual Languages (pp. 336–343). IEEE.

SitoWise (2014), Otaniemen alueen valaistussimulointi. Retrieved from: http://vrs3d.sito.fi/Otaniemen%20alueen%20valaistussimulointi/10 [24.11.2017].

SmartOulu (2018), SmartOulu, Retrieved from: http://smartoulu.mapgets.com/ [24.11.2017]

Schilling, A., Bolling, J., & Nagel, C. (2016, July). Using glTF for streaming CityGML 3D city models. In *Proceedings of the 21st International Conference on Web3D Technology* (pp. 109–116). ACM.

Sulankivi, K., Makela, T., & Kiviniemi, M. (2009, June). BIM-based site layout and safety planning. In *Proceedings of the First International Conference on Improving Construction and Use through Integrated Design Solutions* (pp. 125–140).

three.js (2018), three.js – Javascript 3D library. Retrieved from: http://threejs.org/ [29.5.2018]

Trenholme, D., & Smith, S. P. (2008). Computer game engines for developing first-person virtual environments. Virtual reality, 12(3), 181–187.

Trimble (2018a), Trimble Connect, Retrieved from: https://connect.trimble.com [18.5.2018]

Trimble (2018b), SketchUp, Retrieved from: https://www.sketchup.com/ [18.5.2018]

Vaaja, M., Hyyppä, J., Kukko, A., Kaartinen, H., Hyyppä, H., & Alho, P. (2011). Mapping topography changes and elevation accuracies using a mobile laser scanner. Remote Sensing, 3(3), 587–600.

Vaaja, M.T., Kurkela, M., Virtanen, J.P., Maksimainen, M., Hyyppä, H., Hyyppä, J. & Tetri, E. (2015). Luminance-Corrected 3D Point Clouds for Road and Street Environments. Remote Sensing, 7(9), 11389–11402.

Virtanen, J. P., Hyyppä, H., Kämäräinen, A., Hollström, T., Vastaranta, M., & Hyyppä, J. (2015). Intelligent Open Data 3D Maps in a Collaborative Virtual World. ISPRS International Journal of Geo-Information, 4(2), 837–857.

Virtanen, J.P., Kurkela, M., Turppa, T., Vaaja, M.T., Julin, A., Kukko, A., Hyyppä, J., Ahlavuo, M., von Numers, J.E., Haggrén, H. and Hyyppä, H. (2018) Depth camera indoor mapping for 3D virtual radio play. The Photogrammetric Record.

Van Schaik, P. (2010). Using interactive 3-D visualization for public consultation. Interacting with Computers, 22(6), 556–568.

virtualcitySYSTEMS (2018a), Helsinki 3D with the virtualcityDATABASE, Retrieved from: http://www.virtualcitysystems.de/en/news/407-helsinki-3d-with-the-virtualcitydatabase [14.5.2018]

virtualcitySYSTEMS (2018b), virtualcityMAP, Retrieved from: http://www.virtualcitysystems.de/en/products/virtualcitymap [14.5.2018]

VizeiCities (2016), A framework for 3D geospatial visualization in the browser, Retrieved from: https://github.com/UDST/vizicities [18.5.2018]

Wack, R., & Stelzl, H. (2005). Laser DTM generation for South-Tyrol and 3D-visualization. International Archives of Photogrammetry, Remote Sensing and Spatial Information Science, 36, 3-W19.

Wang, J., Lewis, M. & Gennari, J. (2003, December). Emerging areas: urban operations and UCAVs: a game engine based simulation of the NIST urban search and rescue arenas. In *Proceedings of the 35th conference on Winter simulation: driving innovation* (pp. 1039–1045). Winter Simulation Conference.

Wenzel, S., Bernhard, J. & Jessen, U. (2003, December). Visualization for modeling and simulation: a taxonomy of visualization techniques for simulation in production and logistics. In *Proceedings of the 35th conference on Winter simulation: driving innovation* (pp. 729–736). Winter Simulation Conference.

Wissen, U., Schroth, O., Lange, E. & Schmid, W.A. (2008). Approaches to integrating indicators into 3D landscape visualisations and their benefits for participative planning situations. Journal of Environmental Management, 89(3), 184–196.

Wu, H., He, Z. & Gong, J. (2010). A virtual globe-based 3D visualization and interactive framework for public participation in urban planning processes. Computers, Environment and Urban Systems, 34(4), 291–298.

Yan, W., Culp, C. & Graf, R. (2011). Integrating BIM and gaming for real-time interactive architectural visualization. Automation in Construction, 20(4), 446–458.

Zhu, L. & Hyyppa, J. (2014). The use of airborne and mobile laser scanning for modeling railway environments in 3D. Remote Sensing, 6(4), 3075–3100.

Zhu, L., Hyyppä, J., Kukko, A., Kaartinen, H., & Chen, R. (2011). Photorealistic building reconstruction from mobile laser scanning data. Remote Sensing, 3(7), 1406–1426.

Zhu, L., Lehtomäki, M., Hyyppä, J., Puttonen, E., Krooks, A., & Hyyppä, H. (2015). Automated 3D scene reconstruction from open geospatial data sources: Airborne laser scanning and a 2D topographic database. Remote Sensing, 7(6), 6710–6740.