

INTERACTIVE GEO-INFORMATION IN VIRTUAL REALITY – OBSERVATIONS AND FUTURE CHALLENGES

J.-P. Virtanen ^{1,2,*}, A. Julin ¹, H. Handolin ¹, T. Rantanen ¹, M. Maksimainen ¹, J. Hyypä ², H. Hyypä ¹

¹ Aalto University School of Engineering; P.O. Box 14100, FI-00076 AALTO, Espoo, Finland - (juho-pekka.virtanen, arttu.julin, hannu.handolin, toni.rantanen, mikko.maksimainen, hannu.hyypa)@aalto.fi

² Finnish Geospatial Research Institute FGI; Geodeetinrinne 2, FI-02430 Masala, Finland - juha.hyypa@nls.fi

Commission VI

KEY WORDS: Virtual reality, Game engine, Indoor model, Mesh model, Immersive visualization

ABSTRACT:

Visualization applications are an increasingly significant component in the field of 3D geo-information. In them, the utilization of consumer grade virtual reality (VR) head mounted displays (HMD) has become a topical research question. It is notable, that in most presented implementations, the VR visualization is accomplished by a game engine. As game engines rely on textured mesh models as their conventional 3D asset format, the challenge in applying photogrammetric or laser scanning data is in producing models that are suitable for game engine use. We present an example of leveraging immersive visualization in geo-information, including the acquisition of data from the intended environment, processing it to a game engine compatible form, developing the required functions on the game engine and finally utilizing VR HMDs to deploy the application. The presented application combines 3D indoor models obtained via a commercial indoor mapping system, a 3D city model segment obtained by processing airborne laser scanning data, and a set of manually created 3D models. The performance of the application is evaluated on two different VR systems. The observed capabilities of interactive VR applications include: 1) intuitive and free exploration of 3D data, 2) ability of operate in different scales, and with different scales of data, 3) integration of different data types (such as 2D imaging and 3D models) in interactive scenes and 4) the possibility to leverage the rich interaction functions offered by the game engine platform. These capabilities could support several use cases in geo-information.

1. INTRODUCTION

Visualization is an increasingly significant component in 3D geo-information, required for multiple applications such as education, cultural heritage preservation and decision making. Visualization of geo-information also has a significant role in the smart city paradigm (Daniel & Doran, 2013). In 3D visualization, the application of consumer grade virtual reality (VR) head mounted displays (HMD) (Avila & Bailey, 2014) has become a topical research question, attracting lots of attention during the recent years. While the currently available systems still contain a number of issues (e.g. in usability, see: McGill et al., 2015), a large amount of prototype applications utilizing them have nevertheless been introduced (e.g. Beattie et al., 2015; Froehlich & Azhar, 2016; Velev & Zlateva, 2017). It is notable, that in most presented implementations, the VR visualization is accomplished by a game engine, rather than existing GIS platforms (see, e.g.; Šašinka et al., 2019; Jamei et al., 2017). If VR HMDs are to be applied with real-world data, e.g. in landscape planning, urban planning or cultural heritage, the 3D reconstruction of the existing environment must therefore be solved so that the results are compatible with existing commercial game engines. After this, the features of the engine can be applied for developing the desired application functionalities (Jamei et al., 2017). This approach has been applied in geo-information already prior to interest in VR (e.g. Manyoky et al., 2014). Such applications that utilize game engines but fall outside the entertainment domain are occasionally referred to as “serious games” (Anderson et al., 2010).

3D reconstruction of real-world scenes can be accomplished by several techniques. In the context of geo-information, the two primary methods are photogrammetry and laser scanning. Both have been extensively applied and evaluated (Kersten et al., 2015; Grussenmeyer et al., 2008; Baltsavias, 1999). Kersten et al. (2015) conclude that both laser scanning and photogrammetry were able to produce detailed 3D data from the built environment. As a laser scanner operating on a single wavelength is unable to capture color information, it is common practice to combine image acquisition with laser scanning (e.g. Lindstaedt et al., 2011; Balsa-Barreiro & Fritsch, 2017). Additionally, separate instruments are available for 3D measurement of smaller objects (e.g. Bruno et al., 2010), and specific tasks, such as indoor mapping, where depth cameras have also been applied (Henry et al., 2012; Virtanen et al., 2018).

As illustrated by Kersten et al. (2015), environments can be reconstructed either as triangle meshes or more sophisticated CAD models. As game engines rely on textured mesh models as their conventional 3D asset format, this appears to offer a direct route. However, mesh models produced via laser scanning or photogrammetry commonly contain extremely high polygon counts: in Kersten et al. (2015), the photogrammetric model contained 6M polygons and the model formed by triangulation from laser scanning 10M. In Julin et al. (2019) the “hybrid model” produced by a combination of TLS and photogrammetry contained 694M polygons prior to downsampling operations. The mentioned polygon counts are vastly higher than those reported in game engine or other real-time rendered

* Corresponding author

applications, e.g. 0.9M polygons in Kersten et al., (2018a); 0.7M in Tschirschwitz et al., (2019a) and 0.5M in final application produced in Julin et al., (2019). The challenge in applying photogrammetric or laser scanning data for game engines is in producing textured mesh models whilst maintaining sufficiently low data volumes. Currently, the commonly applied solution is to first form a dense mesh model by triangulation, and then rely on algorithmic mesh decimation to obtain a less data intensive model. This method is also increasingly encouraged by the game engine community (see e.g. Lachambre et al., 2017). More optimized models can be created by manual modeling, but this easily requires a significant amount of working time (e.g. 26% of the project in Kersten, 2007).

Leveraging immersive visualization in geo-information therefore becomes a multi-step engineering problem, requiring the acquisition of data from the intended environment, processing it to a game engine compatible form, developing the required functions on the game engine and finally utilizing VR HMDs to deploy the application. In this interest, we present our work in using VR HMD's for visualizing various geo-information datasets with a commercial game engine. We present the equipment used, the data processing methods applied and finally the operation of the developed VR application.

2. MATERIALS AND METHODS

2.1 Textured 3D indoor model

The commercial indoor mapping system, Matterport, was applied for 3D mapping an interior of a daycare center located in Vantaa, Finland. The mapped interior consisted of three connected rooms with their furnishings. The data acquisition was performed by mounting the instrument on a tripod, using two alternative heights to minimize occlusions in the final data. In total, the mapping required a half of a day of working time from a single operator.

To produce a comprehensive dataset, the three-room interior was first measured with the doors connecting the rooms left open. Thus, it was possible to obtain a dataset covering the entire area, with 13 scan positions. After this, the rooms were scanned individually, with doors closed. The campaigns consisted of 12, 14 & 10 scanning positions for each of the three rooms. This was done firstly, to obtain models with closed doors, and secondly, to produce as high-quality models as possible, as the model detail level in Matterport system has been reported to vary according to the physical size of the mapped area (Virtanen et al., 2018). In total the models from the three rooms mapped individually consisted of 1.32 million triangles. For comparison, the mesh model containing all three rooms, mapped in a single project consisted of 336 671 polygons, resulting in a significantly lower polygon density (Figure 1).

The mesh models produced automatically in the Matterport cloud were post-processed by co-registering the models of individual rooms with the ICP method, using the larger dataset as reference. CloudCompare (Version 2.6.1) was applied for registration. After this, the co-registered mesh models of individual rooms were decimated to reduce polygon count, using Blender (Version 2.78) tool "Decimate modifier" with setting "Planar", effectively combining neighboring triangles with an angle difference of less than 3 degrees. This allowed merging polygons from planar regions, whilst maintaining the polygon density in curved regions (Figure 1). Table 1 gives the

polygon counts for mesh models before and after decimation operation. The decimated models were segmented in Blender according to bounds of automated texture mapping in Matterport cloud, producing a total of 18 individual mesh objects, each having a single texture map, but sharing the same origin and orientation. Finally, the models were edited manually to remove window surfaces that were poorly represented in Matterport mesh models.

	Original polygon count	Decimated polygon count (% of original)
Room 1	301 289	268 427 (89 %)
Room 2	606 471	409 759 (68 %)
Room 3	414 538	201 290 (49 %)
Total	1 322 298	879 476 (67 %)

Table 1. Polygon counts of indoor models prior and after decimation.

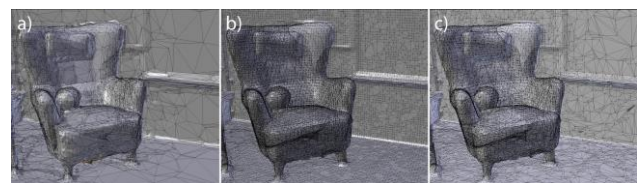


Figure 1. (a) Mesh from a scan project covering all rooms, (b) Mesh from a scan project of a single room, (c) Decimated mesh of single room scan.

2.2 Urban area 3D model

A segment of an urban environment in Otaniemi, Finland was modeled from airborne laser scanning (ALS) data using Terrasolid software package and Blender. The Terrasolid software package includes a suite of tools designed for processing point cloud and image data that run on top of Bentley Microstation. The ALS dataset was collected from a fixed wing aircraft using Teledyne Optech Titan multispectral airborne LiDAR system. First, the ALS point cloud was classified and filtered (for erroneous points) using various classification routines in TerraScan (version 016.022). The TerraModeler (version 016.009) was then used to produce a digital terrain model (DTM) from a specifically thinned group of points (model keypoints) representing the ground surface. The building models were automatically generated from the classified building roof points using Vectorize Buildings tool in TerraScan and then exported with Export City Model Tool in TerraPhoto (version 016.009). The total of 302 individual building objects were combined to a single file, and automatically optimized in Blender by writing a script (in Python) to analyze the models for duplicate vertices and recalculating surface normals to face outside. After this, the combined building mesh models were UV-textured from an orthogonal top projection to allow utilize an orthorectified image as texture for building roofs.

After this, the DTM was post-processed to produce a computationally lighter game engine model. This consisted of decimating the detailed DTM to form a low polygon count equivalent and then utilizing the original DTM to produce a normal map texture for the lower polygon count model (Figure 2). The original DTM consisted of 1 837 928 triangles, which was reduced to 55 136 using the decimate modifier in Blender. The building models were textured with a solar power production potential map obtained from the Espoo City Planning Office (2012). An overview of the processing steps is

illustrated in Figure 3. The final, textured model is shown in Figure 5a.

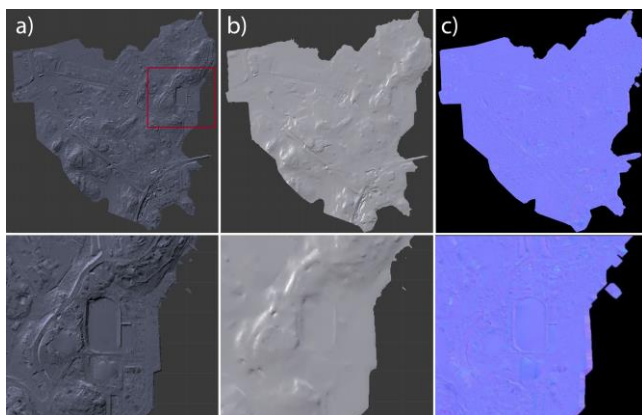


Figure 2. (a) The original high detail DTM, (b) A decimated version, (c) A false color normal map depicting surface normal differences between the detailed and simple version, allowing the simple version to be rendered with more detail. The lower row shows close-ups from the area marked in red for illustration.

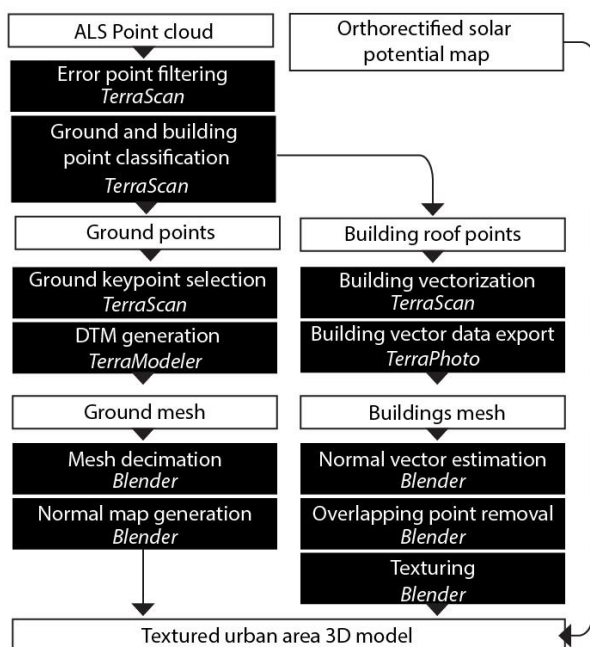


Figure 3. An overview of the processing steps for producing the urban 3D model. The datasets are denoted in white boxes, whereas the processing steps are in black.

2.3 Unreal Engine scene

A commercial game engine, Unreal Engine (Version 4.19) was applied for developing the VR application. Unreal Engine supports the input of mesh models and their textures in a variety of commonly used formats. For straightforward development of interactive applications, it features a visual programming system known as “Blueprints”.

The Unreal Engine template scene for VR environments was used as the starting point for development. This scene offers a

ready setup comprising of the software components required for utilizing the VR HMD and controllers. The scene assembly phase included importing all pre-processed assets to Unreal Editor, positioning them, inserting lights to the scene and developing the interaction functions with the Blueprint development tools of the Unreal Engine. The mesh models utilized included the 3D indoor model (as described in 2.1), urban area model (section 2.2) and a set of manually created 3D models depicting low-emission electricity production methods. As an example of expansive statistical data visualization, the atmospheric CO₂ map produced by NASA's Goddard Space Flight Center (2014) was also imported to the scene. For illustration, the global dataset was applied as a texture of a 3D sphere, in effect producing a simple virtual globe with a video texture. The technical properties of 3D assets generated with 3D mapping and manual modeling are given in table 2.

3D Indoor model			
3D Model	Polygons	Texture maps (resolution, px)	Model production method
Room 1	189 773	5 (2048x2048)	3D Indoor mapping
Room 2	295 581	7 (2048x2048)	
Room 3	320 819	6 (2048x2048)	
Blinds 1	1 140	0	Manual 3D modeling
Blinds 2	700	0	
Urban 3D model			
Otaniemi DTM	55 136	1 normal map (2048x2048)	Triangulation from ALS Building vectorization
Otaniemi buildings	54 382	1 (2048x2048)	
Global geo-information assets			
Planet sphere	3 968	1 video, (1280x720)	Manual 3D modeling
Individual 3D objects			
Wind turbine	682	0	Manual 3D modeling
Wind turbine blades	1 965	0	
Solar panel	10 205	0	
Hydropower plant	10 307	0	
Total	944 658	19 + 1 normal map + 1 video	

Table 2. 3D Assets used in scene.

To facilitate simple physics simulation of moveable artifacts in the scene, and limit user movement (e.g. preventing walking through wall surfaces), a set of primitive geometric objects were manually placed, following the rendering mesh geometry (Figure 4). The completed application was compiled and packaged to form a Windows executable.

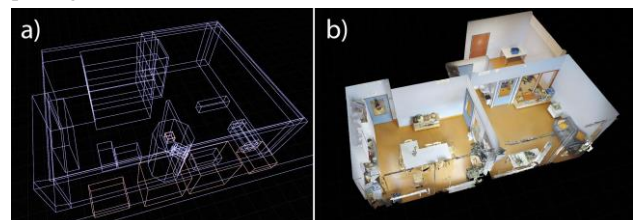


Figure 4. (a) The simplistic objects used as a collision model for physics modeling, (b) The rendered mesh.

2.4 Virtual reality systems

The resulting application was tested with two different VR systems: A stationary installation of HTC Vive Pro was used in combination with a desktop gaming PC. The HTC Vive Pro system utilizes six degrees of freedom (6DoF) tracking that follows the position and rotation of the headset and controllers using two external base stations.

As a portable system, the Dell Visor system, following the Windows Mixed Reality (WMR) architecture by Microsoft was used in combination with a gaming laptop. The Visor system performs the 6DoF tracking of HMD and controllers via a combination of HMD mounted cameras and acceleration sensors. This way, the tracking can be accomplished “inside-out”, without additional external units, increasing the mobility of the system. For the WMR architecture, applications can be developed either as “native” WMR applications, or via the SteamVR libraries, as in this case. (The SteamVR applications can be directly operated on WMR devices.)

Trough 6DoF tracking, both of the systems allowed “room scale VR”, making it possible for the user to freely move in the virtual environment within the bounds of the systems’ cord and HMD tracking range. The specifications of utilized systems are given in table 3. Both computers were running Windows 10 operating system, and Steam VR.

	Desktop VR	Portable VR
CPU	Intel Core i7, 3.6 GHz	Intel Core i7, 2.8 GHz
GPU	GeForce GTX 1080	GeForce GTX 1070
RAM	64 GB	16 GB
HMD	HTC Vive Pro	Dell Visor
Resolution (px / eye)	1440 x 1600	1440 x 1440
Tracking system	“Outside-in”, Two external base stations.	“Inside-out”, HMD’s integrated sensors & cameras
Platform	SteamVR	Windows Mixed Reality / SteamVR

Table 3. Tested VR systems.

3. RESULTS

The completed application allows the user to navigate the virtual environment consisting of the various data assets added to the digitized building interior, such as the urban area model with solar energy production potential, or the global CO₂ visualization (Figure 5).



Figure 5. Geo-information assets visualized in VR environment.

(a) The urban area model (mounted on the wall of one of the rooms) and (b) the virtual globe showing an animated texture.

Interaction functions of the application allow user to interact with the objects in the scene. Manipulatable objects consisted of

a number of moveable toy construction blocks and a target box for throwing them at, moveable flashlight (also manipulating a dynamic spotlight in the scene) and light switches linked with dynamic scene lighting, allowing the user to toggle lights on or off room by room. Additionally, the user was given a possibility to activate the global CO₂ visualization by touching it with a controller, and toggle the display of “holograms” depicting low-emission power production methods. Figure 6 illustrates the different interactions in the scene.

Out of the several game-like features in the application, one was being able to scale down the size of the character model of the VR user to a general kindergarten-aged child. With the feature, the user can view the scene from a child’s viewpoint, providing a novel approach to the environment and interaction. After diminishing the avatar’s size, all of the tasks in the VR environment become much more difficult. Reaching to the light switch demands physical stretching, throwing objects successfully requires more arm swing, and even navigating around the environment model becomes harder. Additionally, a set of commonly utilized VR functionalities were applied, such as the ability to navigate the scene with motion controllers using teleportation, a common locomotion method in VR applications. All lighting in the scene was dynamic and the VR application used the default mode of deferred shading for rendering the materials and lights in Unreal Engine.



Figure 6. User interaction in the VR scene: (a) User manipulating a construction block, (b) defining a location to teleport to, (c) user exploring the space with a flashlight after turning of the light, and (d) viewing the “holograms” showing energy production methods.

3.1 Performance evaluation

To assess the operation of the developed application on the two different VR systems, performance evaluation was carried out. MSI Afterburner -software (version 4.5.0) and Riva Statistics Server (version 7.1.0) were used to log a set of performance parameters: GPU usage, GPU memory usage, CPU usage (combined for all cores), RAM usage, frametime and framerate. Both of the HMDs require a rendering resolution slightly higher than the native panel resolution of the display. This is apparently to allow for compensation of barrel distortions on the HMD. The performance evaluations were performed with two alternative rendering resolutions, 100% and 20%. For the HTC Vive Pro, the 100% rendering resolution setting corresponds to a rendering resolution of 2016 x 2240 px / eye, and for Dell

Visor, 1593 x 1593 px / eye. In addition, a very low rendering resolution on 20% was used, resulting in 902 x 1002 px / eye in HTC Vive Pro and 712 x 712 px / eye for the Dell Visor.

The test consisted of launching the application, allowing it load (this was determined by waiting for the CPU load to stabilize), and navigating the scene. The experiment started with only the exterior sky light of the scene enabled, to simulate the lighting at nighttime. The experiment progressed by turning on the lights in the rooms one by one, in 30 seconds intervals. The experiment concluded with a total of four real time lights in the scene. The recorded framerates and GPU load percentages are shown in Figures 7 & 8. In all lighting configurations, the CPU loads and memory usages remained on a similar level. They were therefore assumed to be insignificant for performance variations in this case, even though the CPU side processing does contribute to the total framerate.

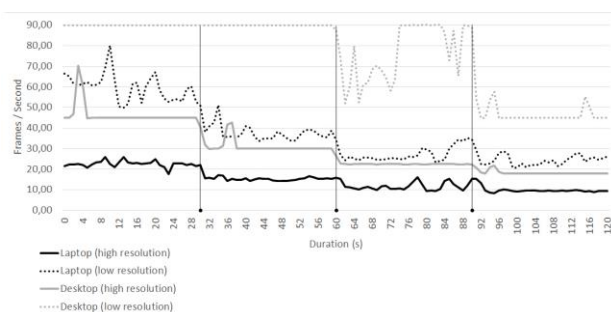


Figure 7. The attained framerate for the two different systems, on two rendering resolutions (100% and 20%). The approximate times for switching on additional lights in the scene are marked at 30, 60 and 90 seconds. As the amount of lights using dynamic shadows and illumination increases, the rendering performance drops.

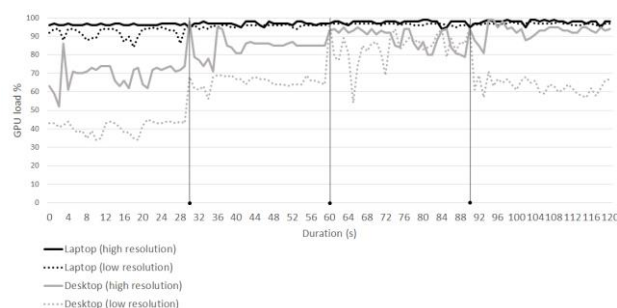


Figure 8. The GPU loads for the two different systems, on two rendering resolutions (100% and 20%). The approximate times for switching on additional lights in the scene are marked at 30, 60 and 90 seconds.

The highest recorded frame rates of 90 fps correspond to the maximum refresh rates (90 Hz) of the HMDs used. In this case, the system is able to render every drawn frame. This was only attained with the more powerful desktop system on low resolution, and maximum of two real time lights in the scene. As the amount of lights is further increased, the GPU load begins to approach 100% even on a fairly powerful system. The lowest recorded frame rates of approx. 10 fps are no longer usable and cause extreme discomfort for the user. Clearly, the most significant performance impact in application comes from the amount of real time rendered lights. Other variations in the performance most likely originate from the culling methods

(frustum & occlusion culling) for visibility and occlusion in the view frustum, that affect the number of polygons the system has to draw.

4. CONCLUSIONS

From the performance evaluation, it is apparent that contemporary room scale VR systems are relatively resource intensive, especially for the GPU. The HTC Vive Pro headset's rendering resolution of 2016 x 2240 px / eye requires a total of slightly over 9 million pixels to be rendered in 11 ms. This has to be taken into account in designing the game engine scenes, as "careless" utilization of dynamic real time lights may lead to insufficient performance even on high end platform.

Excluding the computationally heavy real time lighting setup, both of the VR systems were applicable to exploring the scene containing a multiple textured mesh models, video textures and interactive objects. By utilizing the Steam VR, it was possible to operate the same application on two different HMD systems. The laptop system with Windows Mixed Reality headset (Dell Visor) is especially interesting for mobile use cases: as the entire system consists of the HMD, handheld controllers and a gaming laptop, it can be transported easily. The desktop system with external trackers is more suited for semi-permanent installations. The HTC Vive system, in its different versions, has been extensively used in geo-information research literature, especially for cultural heritage work (e.g. Kersten et al., 2018a; Kersten et al., 2018b; Tschirschwitz et al., 2019a; Tschirschwitz et al., 2019b). Exploring more mobile systems, such as the Windows Mixed Reality, that are compatible with the same content production workflows (in this case, the Unreal Engine), is highly topical for increasing the mobility of VR applications. The fully mobile VR systems, realized via dedicated hardware, such as the Oculus Quest (Facebook Technologies, 2020) further promote the portability of VR systems compared with many current high-end systems, that are essentially PC peripherals.

As we can see by looking at the polygon counts, the models produced by 3D mapping methods are still quite large, amounting to a total of 915 691 polygons, compared with the 28 697 polygons for most of manually built 3D assets. Out of the total polygon count in the scene, the mesh models from 3D mapping amounted to more than 96% of polygons. Even though the optimization procedures applied had reduced the polygon counts to 66% (for the 3D indoor model) and approx. 3% (for the urban 3D model) from the original. Without the optimization procedures, using the 3D assets from mapping methods would have been unfeasible.

Other authors have also mentioned the polygon counts of 3D mesh models used in game engine mediated VR as a central aspect for maintaining sufficient performance, especially when utilizing 3D mapping methods for content production: In Kersten et al. (2018a), the model polygon count was reduced from 6.5M polygons first to 1.5M, and finally to 900k. In Tschirschwitz et al. (2019a), the final polygon count of 659k was concluded to result in sufficient performance. In terms of polygon counts, our experiments correlate with prior research. However, the results clearly indicate that other factors may also lead to loss of performance, even if polygon counts of models are within conventional limits. The model polygon count does not guarantee the performance of a game engine application, neither is it the only performance factor deserving consideration.

Additionally, the models produced with 3D reconstruction methods commonly encountered in geo-information have a number of properties that hinder the development of interactive game engine applications. These issues include the lack of object separation and hierarchy in 3D reconstructed models, presence of existing lighting conditions in photo textures, and lack of low-polygon equivalents that could be leveraged for e.g. physics simulation, necessitating manual modeling, albeit on a very simple level.

The capabilities of interactive VR applications include: 1) intuitive and free exploration of 3D data, 2) ability of operate in different scales, and with different scales of data, 3) integration of different data types (such as 2D imaging and 3D models) in interactive scenes and 4) the possibility to leverage the rich interaction functions offered by the game engine platform. The interaction functions also allow “gamification” and “serious gaming” approaches in applications. These capabilities could also support several use cases in geo-information, including the visualization of remote sensing data to support human interpretation, illustration of various scenarios in the built environment via 3D urban models, exploration of 3D indoor models, and joint visualization of datasets of several different scales (e.g. simultaneous visualization of urban model and global remote sensing data to better illustrate the relations of global phenomena and regional environments).

Several development needs can be identified from the presented work. Better support of point cloud data sets, improvement of semantic classification of indoor data sets and implementation of automated 3D reconstruction pipelines supporting multiple levels of detail in mesh models would all greatly simplify the future realization of immersive applications via game engines. This would allow the immersive visualization potential offered by current VR HMD systems to be increasingly utilized in geo-information.

ACKNOWLEDGEMENTS

This research project was supported by the Academy of Finland, the Centre of Excellence in Laser Scanning Research (CoE-LaSR) (No. 272195, 307362). Strategic Research Council at the Academy of Finland is acknowledged for financial support for project “Competence Based Growth Through Integrated Disruptive Technologies of 3D Digitalization, Robotics, Geospatial Information and Image Processing/Computing - Point Cloud Ecosystem (No. 293389, 314312)”. The Business Finland project “VARPU” (7031/31/2016) is acknowledged. The support from European Social Fund for the project S21272 and the European Regional Development Fund for the project “Soludus”, is acknowledged. The support from the “3D Studio” of Aalto University School of Engineering is acknowledged. The authors would like to thank Ordnance Survey GB (<https://www.ordnancesurvey.co.uk>) and 1Spatial (<https://1spatial.com/>) for sponsoring the publication of this paper.

REFERENCES

Anderson, E.F., McLoughlin, L., Liarokapis, F., Peters, C., Petridis, P., De Freitas, S., 2010: Developing serious games for cultural heritage: a state-of-the-art review. *Virtual reality* 14(4), 255-275.

Avila, L., Bailey, M., 2014: Virtual reality for the masses. *IEEE computer graphics and applications* 34(5), 103-104.

Balsa-Barreiro, J., Fritsch, D., 2017: Generation of visually aesthetic and detailed 3D models of historical cities by using laser scanning and digital photogrammetry. *Digital Applications in Archaeology and Cultural Heritage*, 57-64.

Baltsavias, E.P., 1999. A comparison between photogrammetry and laser scanning. *ISPRS Journal of photogrammetry and Remote Sensing* 54(2-3), 83-94.

Beattie, N., Horan, B., McKenzie, S., 2015: Taking the LEAP with the Oculus HMD and CAD-Plucking at thin Air?. *Procedia Technology* 20, 149-154.

Bruno, F., Bruno, S., De Sensi, G., Luchi, M.L., Mancuso, S., Muzzupappa, M., 2010: From 3D reconstruction to virtual reality: A complete methodology for digital archaeological exhibition. *Journal of Cultural Heritage* 11(1), 42-49.

Daniel, S., Doran, M.A., 2013. geoSmartCity: geomatics contribution to the smart city. *Proceedings of the 14th annual international conference on digital government research*, 65-71.

Espoo City Planning Office, 2012. Solar energy map of Espoo. https://hri.fi/data/en_GB/dataset/espoon-aurinkoenergiakartta (26 November 2019).

Facebook Technologies, 2020. Oculus Quest. https://www.oculus.com/quest/?locale=en_US (11 July 2020).

Froehlich, M. A., Azhar, S., 2016. Investigating virtual reality headset applications in construction. *Proceedings of the 52nd ASC International Conference*, 13-16.

Grussenmeyer, P., Landes, T., Voegtli, T., Ringle, K., 2008. Comparison methods of terrestrial laser scanning, photogrammetry and tacheometry data for recording of cultural heritage buildings. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 37(B5), 213-218.

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.

Jamei, E., Mortimer, M., Seyedmahmoudian, M., Horan, B., Stojcevski, A., 2017: Investigating the role of virtual reality in planning for sustainable smart cities. *Sustainability* 9(11).

Julin, A., Jaalama, K., Virtanen, J.P., Maksimainen, M., Kurkela, M., Hyypä, J., Hyypä, H., 2019: Automated Multi-Sensor 3D Reconstruction for the Web. *ISPRS International Journal of Geo-Information* 8(5).

Kersten, T.P., 2007. Virtual Reality Model of the Northern Sluice of the Ancient Dam in Marib/Yemen by Combination of Digital Photogrammetry and Terrestrial Laser Scanning for Archaeological Applications. *International Journal Of Architectural Computing* 5(2), 339-354.

Kersten, T., Mechelke, K., Maziull, L., 2015. 3D model of Al Zubarah fortress in Qatar-Terrestrial laser scanning vs. dense image matching. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40(5), 1-8.

Kersten, T.P., Tschirschwitz, F., Deggim, S., Lindstaedt, M., 2018a. Virtual reality for cultural heritage monuments – from 3D data recording to immersive visualisation. *Digital Heritage. Progress in Cultural Heritage: Documentation, Preservation, and Protection. Lecture Notes in Computer Science*, 74-83.

Kersten, T.P., Tschirschwitz, F., Lindstaedt, M., Deggim, S., 2018b: The historic wooden model of Solomon's Temple: 3D recording, modelling and immersive virtual reality visualisation. *Journal of Cultural Heritage Management and Sustainable Development* 8(4), 448-464.

Lachambre, S., Lagarde, S., Jover, C. 2017. Unity Photogrammetry Workflow. https://unity3d.com/files/solutions/photogrammetry/Unity-Photogrammetry-Workflow_2017-07_v2.pdf (26 November 2019).

Lindstaedt, M., Mechelke, K., Schnelle, M., Kersten, T., 2011. Virtual reconstruction of the almagah temple of Yeha in Ethiopia by terrestrial laser scanning. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(5/W16), 381-390.

McGill, M., Boland, D., Murray-Smith, R., Brewster, S., 2015. A dose of reality: Overcoming usability challenges in vr head-mounted displays. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 2143-2152.

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.

NASA's Goddard Space Flight Center, 2014. A Year In The Life Of Earth's CO₂. <https://svs.gsfc.nasa.gov/11719> (11 July 2020)

Šašinka, Č., Stachoň, Z., Sedlák, M., Chmelík, J., Herman, L., Kubíček, P., Šašinková, A., Doležal, M., Tejkl, H., Urbánek, T., Svatoňová, H., Ugwitz, P., Juřík, V., 2019: Collaborative Immersive Virtual Environments for Education in Geography. *ISPRS International Journal of Geo-Information* 8(1).

Tschirschwitz, F., Büyüksalih, G., Kersten, T.P., Kan, T., Enc, G., Baskaraca, P., 2019a. Virtualising an ottoman fortress–laser scanning and 3d modelling for the development of an interactive, immersive virtual reality application. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42(2/W9), 723-729.

Tschirschwitz, F., Richerzhagen, C., Przybilla, H.J., Kersten, T.P., 2019b: Duisburg 1566: Transferring a Historic 3D City Model from Google Earth into a Virtual Reality Application. *PFG–Journal of Photogrammetry, Remote Sensing and Geoinformation Science* 87(1-2), 47-56.

Velev, D., Zlateva, P., 2017: Virtual reality challenges in education and training. *International Journal of Learning and Teaching* 3(1), 33-37.

Virtanen, J.P., Kurkela, M., Turppa, T., Vaaja, M.T., Julin, A., Kukko, A., Hyypä, J., Ahlavo, M., von Numers, J.E., Haggrén, H., Hyypä, H., 2018: Depth camera indoor mapping for 3D virtual radio play. *The Photogrammetric Record* 33(162), 171-195.