3D Geographic Scenes Visualization Based on WebGL

"3D Geographic Scenes Visualization Based on WebGL" by R. Miao et al. [1] Investigates solutions to render 3D geospatial data. With the emergence of HTML5 (Hyper Text Markup Language 5) and WebGL (Web Graphics Library) these APIs (Application Programming Interface), R.Miao discusses the implementation of these libraries to render a digital city roaming system based on WebGL technology.

HTML5 is the latest standard of HTML (Hyper Text Markup Language), which is a markup language used to describe web pages. HTML5 expands with support for audio and video involving building audio visualisation and video editing [2]. Introduced earlier, WebGL is a JS (Javascript) API for rendering interactive 3D and 2D graphics within web browsers [3].

The authors discuss the network bottlenecks when drawing "*large amounts of data information (texture,lighting and coordinates*" [1]. They discuss present solutions to these issues such as mesh simplification which involves reducing the complexity of a surface model by removing insignificant details[4], however they lack optimization and cross-functionality. By studying scene methodology, model formats and 3D scene scheduling methods, the author's resulting roaming system showed an optimised interactive experience of 3D visualisation for users.

This paper introduces key optimizations that are relevant in improvement of OpenGL performance. The authors discuss the methodology in building their system. One of these proposed solutions discusses "*Occlusion Culling Technology*". This involves removing objects or parts when drawing a 3D scene. Another optimization implemented was glTF (GL Transmission Format) which would compress the model data and is closer to the GL interface.

They discuss the implementation of this optimization and others with the gathered results from simulation 3D scene data of New York City. They had found that there was a 22% decrease in memory usage after compression while the transmission response was less than 3s with the average response time being less than 1s. The average frame rate of the 3D city scene was 36 frames per second.

Though the authors prioritise real-time interaction there are trade-offs affecting the visual fidelity at closer zoom levels. Incorporating hybrid rendering, such as WebGPU alongside WebGL, could offer solutions to maintain both high visual quality and frame rate performance. Similarly, if not fine-tuned Occlusion culling can aggressively limit what is being rendered. In applications where visual continuity is prominent (large terrain or objects) this can reduce the overall quality of the 3D scene.

References:

[1] R. Miao, J. Song and Y. Zhu "3D Geographic Scenes Visualization Based on WebGL" , URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8046999 (Accessed 30/10/24)

[2] A. Malhotra "HTML5 Audio and Video", URL: https://tutorial.techaltum.com/html5-audio-video.html (Accessed 30/10/24)

[3] Mdn Web Docs "WebGL: 2D and 3D graphics for the web", URL:https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API (Accessed 30/10/24)

[4] B. Preim, C. Botha "Visual Computing for Medicine", URL: https://www.sciencedirect.com/book/9780124158733/visual-computing-for-medicine (Accessed 30/10/24)