COMPARATIVE 3D GEOGRAPHIC WEB SERVER DEVELOPMENT:

VISUALIZING POINT CLOUDS IN THE WEB

by

Kevin Mercy

A Thesis Presented to the
FACULTY OF THE USC DORNSIFE COLLEGE OF LETTERS, ARTS AND SCIENCES
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
MASTER OF SCIENCE
(GEOGRAPHIC INFORMATION SCIENCE AND TECHNOLOGY)

August 2020

Acknowledgements

I would like to thank Dr. Andrew Marx and Dr. Steven Fleming for direction and advising of this thesis. I would also like to thank my thesis committee, Dr. Swift and Dr. Chiang for their valuable inputs and assistance in shaping the study.

Table of Contents

List of Tables

List of Figures

Abbreviations

API	Application Programming Interface
EC2	Elastic Compute Cloud
GB	Gigabyte
GIS	Geographic Information System
ICT	Institute of Creative Technologies
Lidar	Light detection and ranging
MB	Megabyte
NOAA	National Oceanic and Atmospheric Administration
OGC	Open Geospatial Consortium
QML	Query Markup Language
S3	Simple Storage Solution
UI	User Interface
USC	University of Southern California
2D	Two-Dimensional
3D	Three-Dimensional
4D	Four-Dimensional

Abstract

GIS Capabilities are rapidly expanding into the web and cloud environments, but there is little research on the capabilities and performance of 3D web GIS exploitation systems. To evaluate current 3D GIS capabilities and performance within the web, Esri ArcGIS Enterprise Portal, Cesium JS, and Hexagon Geospatial Luciad RIA were all configured on a cloud-based Amazon EC2 instance to host and serve 3D tile datasets that implement adaptive tiled data structures. Using two different source point cloud datasets, a high-resolution photogrammetric dataset, and a lower resolution lidar dataset, resource loading time and resource memory was tested within each system with increasing overall tileset sizes and with three different levels of zoom. The results show that while Cesium JS is quickest, Esri ArcGIS Enterprise Portal performs similar and with more detailed visualizations for both datasets. Hexagon Geospatial Luciad RIA performed slower than the other two systems, but possesses the most photorealistic and detailed rendering of the systems. Performance differences between the servers can be seen in the level of library compression and number of libraries imported into the page. Cesium JS is generally quickest, but most compressed and lightweight server. The larger detail and loading time in Esri ArcGIS Enterprise and Hexagon Geospatial Luciad RIA can be traced to smaller levels of compression and more library imports to enhance detail of 3D data rendering. Overall tileset size and spatial resolution of data did not significantly impact performance while zoom level did significantly impact performance. Generally, higher resolution of zoom required more resources and loading time. Results indicated that difference visualization systems are best suited for different applications. Cesium JS would likely be most suited for complex analytic operations, while Hexagon Geospatial Luciad RIA would be best for detailed single scene visualization.

Chapter 1 Introduction

The ongoing migration to web and cloud technologies is occurring due to the larger amounts of space, users, and interoperability between systems that is available in the cloud (Collins 2019). It is also becoming more commonplace for companies to collect geographic data, in both imagery space and in 3D space due to increased accessibility of unmanned aerial systems (UAS) (Whitehead and Hugenholtz 2014). 2D data is more widely utilized and possesses good support in desktop, cloud, and web software, while 3D data is still being adopted and integrated into each of these software platforms. Because of the more recent availability of web and cloud-based GIS systems, there is much less research utilizing these systems than traditional desktop GIS systems. There is even less documentation on open source implementations of web and cloud-based GIS systems which are more intricate to assemble due to the large amount of programming knowledge that is required to create an open source web GIS system. The induction of open source Cesium 3D Tiles as the standard data structure for streaming massive 3D datasets on the web is shifting attention within the geospatial community from commercial software and proprietary data structures to more open and accessible software and data structure formats (Chen et al. 2018; Farkas 2017; Krämer and Gutbell 2015). As open source data formats and software become more common, there are several options available to create web GIS systems which can store and investigate large amounts of 3D data. This study evaluates three different web GIS exploitation systems and provides information on their overall performance and capabilities. The systems evaluated in this study are Esri ArcGIS Enterprise Portal, Hexagon Geospatial Luciad RIA, and Cesium JS. Performance and capability analysis between the different exploitation systems indicate that the level of compression used for JavaScript files, and the amount of library content loaded most significantly impact performance between servers.

1.1. General Objective

This research study aims to fill in academic gaps in comparative analysis of 3D web GIS data types and 3D web GIS platforms. The availability of large computing resources made accessible by cloud services is permitting easy access to visualization of large 3D geographic datasets. Previously, 3D point cloud study has been limited by the computing power of a desktop system, but with evolving capabilities in high performance and cloud computing there are now many evolving capabilities for 3D geographic data (Cura et al. 2017; Guan et al. 2013; Huang et al. 2013; Li et al. 2016). Because of the interoperability of the web and its ease of integration with cloud technologies, many 3D data structures are now being provided in more optimized web-based tile formats. The introduction of 3D Tiles and Esri i3s marks community adoption and use of 3D web streaming formats for 3D data; however, these data structure standards have been implemented fairly recently and there is very little academic study providing technical information on the potential applications, limitations, and performance of these new data types. Additionally, there are several web GIS platforms which integrate these standards which have not been assessed quantitatively against each other. This study analyzes performance and capability differences between 3D GIS streaming data types and exploitation systems, identifying how fundamental differences in how they process and visualize data are responsible for these differences.

1.2. General Methodology

This results from this research provide performance metrics and core 3D service capabilities for each data type and each web GIS system tested. Three different exploitation systems were tested which include Esri ArcGIS Enterprise Portal, Hexagon Geospatial Luciad RIA, and Cesium JS. These systems have received attention within the larger industry community and are ideal systems to test due to their support of 3D GIS data. The OGC has standardized both Esri i3s and Cesium 3D Tiles as data structure standards for large web-based 3D GIS datasets (Open Geospatial Consortium 2019; Open Geospatial Consortium n.d.). Due to their standardization by the OGC, Cesium 3D Tiles and Esri i3s data structures were the primary web-based data structures evaluated in the study. Both data structures are optimized for the web and for tile-based services which attempt to enhance performance of visualization by limiting the amount of data which is rendered on client devices based on zoom level in the virtual globe engines. Through their use of tiling, these data structures allow client devices to work with datasets which are very large.

To provide a controlled computing environment for gathering of performance metrics, all the exploitation systems (Esri ArcGIS Enterprise Portal, Hexagon Geospatial Luciad RIA, and Cesium JS) were installed and configured on a virtual Amazon Elastic Compute Cloud (EC2) instance. All source data and processed data was stored on the virtual EC2 instance to ensure that all server resources originated from the same computing environment. Despite its virtualization, EC2 instances permit dedicated storage volumes to store the operating system, and data of the EC2. A 700 GB dedicated storage volume was attached to the EC2 to keep all software and data resources on the same data storage volume. Testing was also performed on the same client device over a consistent network connection to limit potential error in testing. Using a controlled server environment and client environment ensured that performance results could be directly compared to each other.

To understand how the exploitation systems handled different sizes of data, and different types of data, one high-resolution photogrammetric dataset, and one lower resolution lidar dataset were processed into Cesium 3D Tiles and Esri i3s at increasing tileset sizes. Each dataset was processed into 1GB, 5GB, 10GB, 25GB, and 50GB tileset sizes to understand performance differences of each server framework for small and large datasets. Additionally, two datasets, a photogrammetric and a lidar dataset were used as the source datasets to understand the effect of lower and higher spatial resolution on the performance of the server. Both source datasets were processed into Esri i3s and Cesium 3D Tiles to observe any differences in performance that may have derived from data structure type. Processing of data from two different collection methods and therefore variant spatial resolution, and in two different data structures, and for five total tileset sizes allowed gathering of metrics from each server for a variety of input data conditions.

Performance metrics were calculated in several different manners. Using GtMetrix, an industry standard website analysis toolset, each exploitation system was evaluated for its basic performance in loading a 3D tiled dataset of 1GB (GtMetrix). GtMetrix calculates a website's Google Page Score, Yahoo YSlow Score, Fully Loaded Time, and retrieves the network waterfall which shows all network requests and resource loading times made by the page. The scores from Google Page Score, and Yahoo YSlow illustrate an A-F ranking on the webpage performance. The waterfall information provides more quantitative information from which the Google Page Score and Yahoo YSlow are based off of. In addition to the snapshot of performance information provided by GtMetrix, for each dataset, at each memory size, in each exploitation system, the resource loading time and the total memory of resources used were calculated for multiple zoom levels. These two attributes (resource loading time, and total resource memory) can be compared across each test to understand overall performance differences between each test. Due to the nature of tile datasets, tiles are sent to the client based off of the zoom level in the exploitation system. Therefore, multiple zoom levels were necessary to test in order to observe the performance metrics of each exploitation system for different sized tiles. To observe effects on performance for different zoom levels, performance metrics were calculated for the full extent of the dataset, at a medium zoom level, and at a detailed zoom level. The results provide useful information on the overall performance, complexity, and functionality of each system based on tileset size, zoom level, and data type.

The methodologies section will provide important setup instructions and configurations for each server architecture which can be used as a development guide. Each software is setup in a different way to be made accessible for use on the web. It is important to understand the configuration of each exploitation system to further understand differences in the performances between them. The results of this research are largely useful to developers, as well as to 3D point cloud researchers and corporations that use 3D GIS data. In addition to providing instructions for setting up each exploitation system in the methodology section, the results provide the core performance metrics and essential capabilities and limitations of each system. Additionally, some basic user experience results were also included to account for differences in the exploitation systems that could not be quantitatively analyzed. Both user experience and performance are important within software development since applications likely have different standards in terms of speed and user experience. Therefore, in the results both quantitative performance metrics and qualitive user experience results are discussed. This research is centrally focused on the scientific research and industry developer communities to assess the current state of 3D web service technology. Developers and scientists can utilize the architecture guide to simply create basic 3D exploitation systems which are capable of visualization of large 3D datasets or to develop and assist progression of new tools and software for 3D geographic data analysis. These systems are relatively light and there is little support for complex operations.

Building more analytic tools within these systems would help push the boundaries of 3D web GIS capabilities.

1.3. Motivation

3D GIS datasets are becoming more common and it is important to note the capabilities of current 3D web GIS systems. There are several options for exploitation systems which involve commercial and open source options. The largest difference between open source and commercial software platforms are the level of programming required to setup the system, the amount of customization that can be programmed into the system, and the auxiliary tools that are available. Because of the support for i3s and 3D tiles, and to have a distribution of commercial and open source web GIS systems, Esri ArcGIS Enterprise Portal, Hexagon Geospatial Luciad RIA, and Cesium JS were chosen as the testing systems in this study. Both Esri ArcGIS Enterprise Portal and Hexagon Geospatial Luciad RIA are commercial options to web GIS. Cesium JS is completely open source and available as a content delivery network. However, all of these tools require some level or programming to be operational. Therefore, it is important to note that managing and operating a 3D web GIS requires knowledge of JavaScript, and HTML/CSS in order to properly configure and optimize systems.

Esri ArcGIS Enterprise Portal is a commercial web-based GIS platform which is completely operational out of the box and includes ability for customization with a developer's license (Esri n.d.a). Web applications can be created through the Portal website, or with a JavaScript application which references layers stored within the ArcGIS Portal. Storage within Esri ArcGIS Enterprise Portal can be managed through ArcGIS Server which allows users to configure data stores that point to locations of either local or remote databases. Once a data store is registered, data uploaded into the store can be served and even analyzed with some basic GIS tools within their web system. Esri ArcGIS Enterprise Portal is also the only system which serves i3s data. Due to its support of i3s, Esri ArcGIS Enterprise Portal was chosen as an exploitation system to test. A testing application for i3s was built using data stored within the Esri ArcGIS Enterprise Portal ecosystem, and a web application built using the Esri ArcGIS JavaScript API.

Hexagon Geospatial Luciad RIA provides a commercial web GIS software development kit which can be used to build 3D web GIS applications. Hexagon Geospatial Luciad RIA has minimal setup and comes packaged with a variety of different applications that can be used as source code to build more complex applications (Hexagon Geospatial n.d.). The Hexagon Geospatial Luciad RIA is essential a sophisticated API which can be used to build complex 2D and 3D GIS web applications. A sample application was built using Hexagon Geospatial Luciad RIA to serve the test tile datasets. The sample application creates a simple virtual globe which serves the test 3D tile datasets. More complex applications could be developed with Hexagon Geospatial Luciad RIA, but the minimal components necessary to serve a 3D dataset were used in this study.

Open source solutions most often integrate a back-end database, which can be stored locally or within the cloud, and a web mapping or virtual globe library to build a functional UI (Cesium). These solutions require usually require the most extensive programming to build up to a point where data storage and visualization are achieved. The Cesium JS library has recently been published as a content delivery network and users can now build Cesium JS applications with very few lines of code. Additional analytical tools can be programmed in using the Cesium API or through pure JavaScript. In order to test Cesium JS, a testing application was created with CesiumJS which serves 3D Tile data on a virtual globe. The application was served with a basic node.js express server.

Cesium JS, Hexagon Geospatial Luciad RIA, and Esri ArcGIS Enterprise Portal have been selected as testing platforms essentially due to their large support for 3D geographic data. Cesium JS is an open source virtual globe library which can be turned into a 3D web application with a small degree of JavaScript programming. As OGC has accepted Cesium 3D Tiles as a community standard for massive streaming of 3D geographic data, Cesium JS is an unquestionable platform to test since its native data type is 3D tiles. Hexagon Geospatial Luciad RIA browser boasts it can render at least 100GB of Cesium 3D Tile data and with its support for 3D tiles it is an ideal software to be tested alongside Cesium JS (Coghe 2018). Esri ArcGIS Enterprise Portal is the last software that will be tested. Developed by Esri, the commercial leader in the GIS industry and the creator of the i3s data format, ArcGIS Enterprise Portal is another system that must be tested. Each of these platforms can be implementable configured for use in an integrated web and cloud environment and support large amounts of 3D GIS data, and therefore make ideal candidates to be tested in this study.

1.4. Potential Applications

The lack of academic information on recent innovations in web GIS data structures and server technology is the core motivation of this study. Advances in 3D data structures are shifting 3D GIS into the web and the cloud and therefore 3D Tiles and i3s are the datasets which are focused on in this study. More optimized web-based 3D data structures implement more efficient use of resources than large point cloud files, which have been an industry standard 3D GIS data format for many years. Due to their support of i3s and 3D Tiles, Cesium JS, Esri ArcGIS Enterprise Portal, and Hexagon Geospatial Luciad RIA are used as the exploitation systems to test performance and capabilities in this study. Integrated web and cloud GIS systems appears to provide the computing environment and data formats necessary to stream massive 3D

GIS datasets which are difficult to work with in traditional desktop GIS software. Quantitative analysis of the performance of these exploitation systems provides valuable information on the capabilities that can be expected from each web GIS visualization engine. The results of this study illustrate foundational information on the performance and limitations of these web systems. The results of this study indicate that each system is likely suitable for different types of work and that there are many areas for further development and growth for 3D web GIS. It appears that these systems excel in rendering large 3D datasets, but additional tools and development is necessary to produce more analytic capabilities within these systems.

Chapter 2 Related Works

Strategies for managing 3D geographic data are becoming more common, largely due to the changes in 3D GIS data structures, and the increasing accessibility of cloud technologies. The following chapter will outline the computing and data structure issues with 3D data in desktop GIS environments and the ongoing progression of 3D web GIS to more optimized web systems and data structures. Researchers have successfully exploited 3D GIS data in desktop systems, but applications and studies are sparse due to the complexity of working with point cloud data. The computing challenges created by large point cloud files have incentivized investigations of new 3D geographic data structures and new methods to manage and analyze 3D geographic data. The most common solutions have been adaptations to 3D geographic data structures and integration of 3D geographic data with either high performance or cloud computing platforms. Thus, in addition to discussion of recent advances in 3D geographic data structures, this chapter will also examine the current 'state of the art' in 3D web GIS systems.

2.1. Introduction to Point Clouds

The industry standard las file format commonly used to store 3D light detection and ranging (lidar) and photogrammetric data, is a binary 'point cloud' data structure that is difficult to analyze. Point clouds often contain 3D geographic and attribute information of millions of points within a single binary file. Even with the reduction of data size from storing the data in a compressed binary format, point cloud files still end up being very large. Because point cloud data is stored in binary, it is also a complex data structure to render, and therefore requires specialized software to interpret the data. The large size and complexity of point cloud files has created many issues in storage, processing, visualization, and exploitation of the underlying 3D point data (Auer and Zipf 2018; Richter and Döllner 2014). The point cloud file has served as the prevalent data structure for high fidelity 3D geographic data but possesses evident limitations due to its large size.

Although point cloud data has posed computing challenges, researchers have been able to successfully exploit point cloud data. Point cloud data has received attention from a diverse community of researchers, with interests in both the urban and the natural environment. Lidar point cloud data has several acquisition methods including aerial and mobile scanners (Esri n.d.c). Aerial lidar has proven to be an effective source of data to create realistic high-resolution 3D city models which are advancing initiatives to virtualize large cities (Jayaraj and Ramiya 2018). In a similar respect, the high resolution provided by 3D lidar point cloud data, has progressed the field of autonomous vehicle study. Mobile lidar is becoming an industry standard implementation in autonomous vehicle design and researchers have analyzed mobile lidar point cloud data to improve data registration, and object identification and segmentation algorithms for autonomous vehicles (Daraeihajitooei 2018; Józsa et al. 2013). Within the natural environment, scientists have shown that point cloud data can be used to predict wetland locations using known locations of wetlands and to accurately determine individual tree sizes (Leonard et al. 2012; Liew et al. 2018;). Ostensibly, point cloud data has served as an enabler to many areas of ongoing science including studies in 3D cities, autonomous vehicles, and ecology.

Point cloud data continues to push research forward in many novel and innovative applications. A groundbreaking study involving an international team of Maya archaeologists has pushed the boundaries of point cloud study by using lidar point cloud data as a reconnaissance asset to identify and map 61,480 new Maya structures which were previously unknown (Canuto et al. 2018). The new capabilities provided by lidar to map entire areas in 3D is completely transforming the field of Maya archaeology. The use of lidar as a reconnaissance asset is further demonstrated by researchers at the Massachusetts Institute of Technology Lincoln Laboratory who have used aerial lidar data in several disaster episodes including Hurricanes Harvey, Irma, Maria, and Florence to assist emergency management teams. The high resolution of the sensor permits mapping of flood zones, and identification of infrastructure damage and debris which has been used by the Federal Emergency Management Agency to inform emergency response and recovery operations (Foy 2018; National Research Council 2014). Point cloud data is continually evolving capabilities and practices of research teams and institutions that acquire it.

2.2. Point Cloud Data Limitations

Because of the complexity of point cloud data management, there is only a small number of standard tools available in most available GIS software. In comparison to the number of standard tools and capabilities available for manipulation and analysis of 2D geographic data, the tools and capabilities available for 3D geographic data are very minimum. Most of the tools available for 3D data include basic data management and manipulation tools for 3D geographic data such as tiling, changing coordinate systems, conversion to another data type, etc. In this regard, many researchers create tools to extend the capabilities and applications of point cloud data. For example, immersion specialists have integrated point clouds into virtual reality space and built a basic toolset for interacting with the data. The technology the researchers created is largely developmental and has not been implemented on large scales (Kreylos et al. 2008). Another group of researchers' setup a sophisticated point cloud server, which can be used to quickly compress, query, and extract points from point cloud data. Although the server is configured with open frameworks and detailed methodology, the server appears difficult to replicate (Cura et al. 2017). Point cloud data is even being collected by indoor photogrammetry for modeling of indoor building environments. There is adequate software available for reconstruction of indoor environments into 3D point clouds, but like other areas of point cloud science, there are only a few tools available to work with the collected data (Wang and Cho 2015). Many researchers are able to work with point cloud data and have created tools to provide new capabilities to point cloud data; however, most of these tools require extensive setup and configuration.

Another potential limitation of point cloud data is the presence or quality of attribute information available in the dataset. One of the defining factors of the point cloud data structure is the presence of classification fields, which can be used to annotate points as a number of different features. Since point clouds have largely been acquired by aerial collection, many of the classification fields involve features relating to the external environment. Examples of point cloud class fields include vegetation, buildings, water, rail, roads, tower, bridge, etc. (Esri n.d.b). In the wetland study, researchers were able to classify their point cloud data by using the known vector boundaries of wetland areas (Leonard et al. 2012). Many applications of point cloud data depend on classification information to be useful, and often classification must be derived using information from another dataset or with an automated ground or surface classification algorithm. Without additional GIS data to inform classification methods, automated filtering algorithms for point clouds are relatively lackluster (Meng et al. 2010). Because accurate classification of point cloud data relies on large amounts of corresponding data, it can be difficult to create attribute information for unclassified point clouds. The classification information of point cloud data is important to research, and the ability to exploit point cloud data is largely dependent on the classification information available in a point cloud or that can be attributed to the point cloud through other data sources. Thus, the quality of classification information defines the extent to which the point cloud data can be analyzed for scientific phenomenon.

The utility of point cloud data and enhanced point cloud analytical tools is apparent in multiple disciplines including archaeology, ecology, city planning, and hazard management. Volumetric data from point clouds is assisting hazard workers with determining volume of fill needed to fix roads damaged by disasters and locations of flood zones and infrastructure damage. Archaeology researchers are able to virtualize sites, and even discover new ruins with point cloud data. Ecologists can precisely measure individual tree size and volume, and city planners can investigate city environments including existing and potential buildings, inside and outside. Overall, there are vast applications for point cloud data, despite its limitations in computing, software tools, and overall data quality. Researchers have been able to work around the limitations of point cloud data to create novel systems and methods which have advanced research in several disciplines.

2.3. Data Structures

To alleviate the computational issues of point cloud data, researchers have investigated new ways to structure 3D geographic data. There have been relatively two different approaches towards restructuring point cloud data, data aggregation and conversion to ‘level of detail’ data structure. Data aggregation results in an overall reduction of data precision. ‘Voxelization’ is a common form of point cloud aggregation that bins points into larger 3D ‘voxel’ structures (Marx et al. 2019). Although voxelization is an aggregative approach that reduces resolution of the underlying data, it greatly reduces the overhead computational cost and makes distributed computing much simpler (Boerner et al. 2017). In applications where the data resolution can be reduced, voxelization presents a powerful method which simplifies the point cloud data structure and greatly reduces the amount of computational resources required to render the dataset.

Another approach to make point cloud data more accessible is the level of detail data structure. The level of detail data structure is an optimized tree structure which tiles data into small tiles which are accessible in a hierarchical tree. The data structure results in a composition of files which only render at run time dependent on the level of zoom in the client system. This data structure essentially partitions the point cloud into smaller elements and only renders portions of the point cloud that correspond with the map region and level of zoom at which the user is at. The level of detail data structure approach allows the resolution of the point cloud to be maintained, while greatly reducing the overhead computational cost of rendering (Richter and Döllner 2010). Implementing a tree-like data structure greatly reduces the overhead cost of rendering point clouds, but due to its file-based structure is often not compatible across other systems (Cura et al. 2017). Transforming point cloud data into a tree like level of detail structure is a sophisticated process that optimizes point cloud rendering speed; however, lacks interoperability and integration across systems due to its file-based structure.

In efforts to increase rendering capabilities of point cloud data researchers have investigated new data structures for point cloud data. Both voxels and level of detail data structures provide ways to reduce the computational costs of point cloud data but present some new minor technical concerns.

2.4. Shift to High Performance and Cloud Computing

Another approach to mitigate computational cost of point cloud data is through high-performance or distributed cloud computing. High-performance computing utilizes clusters of computation resources to store and analyze data. Due to the increased computing capacity rendered by additional computational nodes, high-performance computing provides solutions for much quicker data analysis than on a desktop system. When compared to desktop capabilities, high-performance distributed file systems have shown much greater performance for analyzing large stacks of lidar point cloud data (Guan et al. 2013). Several studies attest to the large increase in performance and capabilities that can be achieved by using distributing computing clusters to work with point cloud data. In one study, researchers compared 3D data management performance in PostGre SQL and in Hadoop, running off a distributed file system, and Hadoop greatly out-performed PostGre SQL (Li 2016). Another group of researchers was able to create a point cloud change detection system in a distributed computing environment using Apache Spark to manage the large stacks of 3D data (Liu et al. 2016). The ability of the researchers to setup a change detection system on a distributed cluster shows that distributed computing enables the ability to work with several large 3D datasets. On most desktop-based systems, point cloud study is limited to single scene visualization and exploitation, and therefore distributed computing provides the ability to extend potential point cloud applications by permitting investigation of multiple point cloud datasets. High-performance distributed computing environments provides the ability to greatly speed up management and analysis of point cloud data.

Distributed computing provides a quicker way to analyze point cloud data, but this requires the possession of a high-performance computing cluster. For some developers, this may not be an issue, but distributed computing is largely only accessible to those working with large amounts of computational resources. Cloud based infrastructure is a derivative of distributed computing; however, cloud services can be bought, and users do not need to possess the physical computing environment. Cloud computing platforms have been compared to high performance computing clusters, and it has been demonstrated that due to communication latency between servers in the virtualized cloud environment that high performance computing is quicker (Jackson et al. 2010). Even though cloud environments are not as quick as high-performance computing environments, they offer a distributed computing environment which is much more efficient and scalable than desktop GIS. Cloud services provide many developers a bridge into distributed computing by providing the ability to run high-performance services without owning any actual infrastructure.

2.5. Shift to Web GIS and 3D Data Streaming

Due to the interoperability of web-based technology and its ability to work with cloud technologies, web is becoming a more common platform for GIS tools and applications. Web applications utilize JavaScript which is an interpreted language that can generally be read by any browser. The standard interpretation and data transfer protocols of web makes data transferable and easily accessible across systems in web environments. Web systems can also link to cloud databases and therefore the increased performance and capabilities brought from cloud platforms can be integrated into web applications. The integration of web clients and back end databases provides a robust and powerful computing environment, which is rapidly being developed to increase 3D GIS capabilities. Recently, the OGC accepted Cesium 3D Tiles as a community standard for streaming massive 3D geographic datasets across the web (Open Geospatial Consortium 2019). Cesium 3D Tiles are a web based json level of detail data structure which are optimized for use in the cloud to stream massive 3D geographic datasets through the web (Cesium 2015). The introduction of 3D tiles, and other web streaming data types in the OGC is signaling a shift from desktop GIS to web GIS for intensive 3D GIS applications.

There are several commercial and open source web GIS platforms which can be used to ingest and work with 3D geographic data, but there is little study on the capabilities of these systems. There are two important studies performed which attempt to analyze performance and capabilities differences between web GIS data types, and web GIS front end frameworks. In the first study, researchers conducted a capabilities assessment between GeoJSON, JSON, CZML, and Cesium 3D Tiles data types for their use in service of four-dimensional (4D) data. The researchers point out that Cesium 3D Tiles are the only data structure capable of rendering large amounts of data, but it does not support temporal visualization or attribute selection. The other data types provide some enhanced abilities in low data volumes, but in terms of high data volume, Cesium 3D Tiles were the only data type that did not crash or cause out of memory issues in testing (Murshed et al. 2018). The other relevant study indicates that there is no existing research comparing 3D web GIS frameworks, and then attempts to perform a qualitative study on available 3D web GIS frameworks. The researchers compare Cesium JS, three.js, and X3DOM.js. In the analysis it is evident that three.js and X3DOM.js can be used to render 3D geospatial data, but due to its extensive geospatial support and data management that Cesium JS is the most well supported platform for 3D geospatial web application development (Krämer and Gutbell 2015). This study illuminates the issue that there is little academic study pertaining to capabilities and performance of 3D web GIS frameworks. These two studies provide preliminary analysis of 3D web GIS data types and frameworks.

3D web GIS is a rapidly evolving area of industry and there is a general lack of academic literature evaluating the new data types and GIS systems that are being developed for the web GIS environment. Cesium 3D Tiles has recently been adopted by the OGC as the standard for streaming massive 3D geographic data. Esri i3c is also an OGC standard, but a web standard for ‘large amounts of heterogeneously distributed 3D geographic information (OGC n.d.).’ The OGC standard definitions for Cesium 3D Tiles and Esri i3s are fairly similar, and there are no studies which implement and compare these two data types. With a lack of technical information about the rendering speeds and visualization results of these data types, it is unclear what similarities and differences there are between these 3D data types. There is also a fundamental lack of academic literature comparing the web services frameworks used to visualize these data structures. In ‘A Case Study on 3D geospatial applications in the web using state of the art webGL frameworks,’ the researchers do a good job of comparing the available open source frameworks for 3D web GIS (i.e. Cesium Js, three.js, and X3DOM.js), but the researchers neglect to analyze commercial web GIS platforms. There are several commercial providers which provide essentially fully operational web GIS systems, including Esri and Hexagon Geospatial. In the realm of 3D geographic data, there is a large amount of research on desktop computing and distributed computing, but relatively no comparative research on the software systems and data structures which are currently being made available for 3D web GIS.

The following literature analysis reveals the current novel ways in which point cloud data is being investigated and highlights the shift from desktop GIS to integrated web and cloud GIS. Scientists have found many successes in exploiting point cloud data, and hopefully the applications for 3D GIS only continue to grow as web and cloud GIS becomes more accessible. Distributed and cloud computing show evident rendering improvements for point cloud data. These platforms provide access to more computing power, which enables investigation of multiple scenes. The use of the cloud, especially, is propelling 3D geographic study and shifting it into the web. The web is a highly interoperable environment which can be backed up on the server side by the computing power of cloud databases. With the ability to link front end clients with back end cloud databases, the power of distributed computing can be made accessible to larger communities of developers. Because of the increasing accessibility to integrated web and cloud systems, 3D data structures have shifted from large point clouds into adaptive web-based level of detail structures which permit streaming of massive 3D datasets inside web clients. The data types and standards for streaming 3D geographic data are fairly new and therefore there is little study on the data types themselves or the platforms that render and visualize this data. It is therefore the intent of this study to provide a formal comparative analysis of web-based 3D GIS data structures and 3D web GIS systems in order to provide a foundational performance and capabilities assessment of these new technologies.

Chapter 3 Data and Methods

In this study, a comparative analysis of both open source and commercial 3D web streaming data types, and 3D web GIS systems was conducted. In the study, web service frameworks which include Cesium JS, Hexagon Geospatial Luciad RIA, and Esri ArcGIS Enterprise Portal were setup on a cloud-based Amazon EC2 instance. Two different 3D data types, Esri i3s and Cesium 3D Tiles, were processed from las data and stored on the same EC2 instance in increasing total tileset sizes. Datasets were processed into five different sizes (1GB, 5GB, 10GB, 25GB, and 50GB) to test server for performance differences for low and high data load. Two different source dataset types, a high-resolution photogrammetric dataset and a lower resolution lidar dataset were processed into Esri i3s and Cesium 3D Tiles in order to observer difference in server performance that may be due to spatial resolution of underlying data. Each server was tested for general performance with metrics collected through GtMetrix based on each sites ability to serve a 1GB tile dataset. The figures generated by GtMetrix include Google page score, YSlow Score, Fully Loaded Time, and the network waterfall which contains every http request, resource loaded, and resource load time performed by the client. GtMetrix is a commonly used software suite to analyze website performance in industry. Additionally, each 3D dataset, for every overall tileset size (1GB, 5GB, 10GB, 25GB, and 50GB) were assessed for resource memory and loading time in each exploitation system at three different zoom levels. These tests ostensibly illustrate performance difference of each server with different dataset sizes and zoom levels. Qualitative user experience information is also briefly discussed based on the visual outputs that were produced from each system. The performance tests provide rigorous metrics on the speed and resources used by each server, and the qualitative results briefly discuss the visualization differences between the server frameworks. These tests and user experience results highlight the essential differences between the server frameworks and data types.

3.1. Project Design

To standardize the computing environment for each 3D web GIS exploitation system, all the exploitation systems were setup on the same Amazon EC2 cloud instance. It has been noted in the literature, that due to changing nature of the hardware underneath the virtualized cloud instance, and minor changes in local network bandwidth that cloud performance will influx small changes over its use (Yue et al. 2013). To consider the changes brought about by the shifting hardware underneath the virtualized cloud instance and for minor changes in internet bandwidth, each test was repeated five times and then mean of each test was taken to standardize the changes brought about by the shifting virtualized hardware and internet bandwidth. With the tests run multiple times, and with all the server software and data on the same virtual system, the computing environment of the study can be considered controlled (Yue et al. 2013).

All the data was stored on the EC2 instance along with the software to run each 3D web GIS exploitation system. Figure 1 depicts the architectural setup of the computing environment. The source data within the EC2 virtual machine consisted of point cloud data of 1GB, 5GB, 10GB, 25GB, and 50GB from each source dataset (the singular lidar dataset, and the singular photogrammetric datasets). Each of these datasets were converted into Esri i3s and Cesium 3D Tiles and these processed datasets were also stored on the EC2 virtual machine. Thus, for every source dataset, there is a Cesium 3D tileset and an Esri i3c dataset which had been processed from the source dataset. Thus, there were a total of 10 source point cloud datasets stored on the server, and 20 processed datasets, 10 in i3s, and 10 in 3D Tiles. A dedicated storage volume of 700GB was attached to the EC2 virtual machine so that all the datasets could fit on the server.

Placing both the server software and all the data within the same virtual machine allowed access of all datasets to client applications and also a controlled environment for running the server.

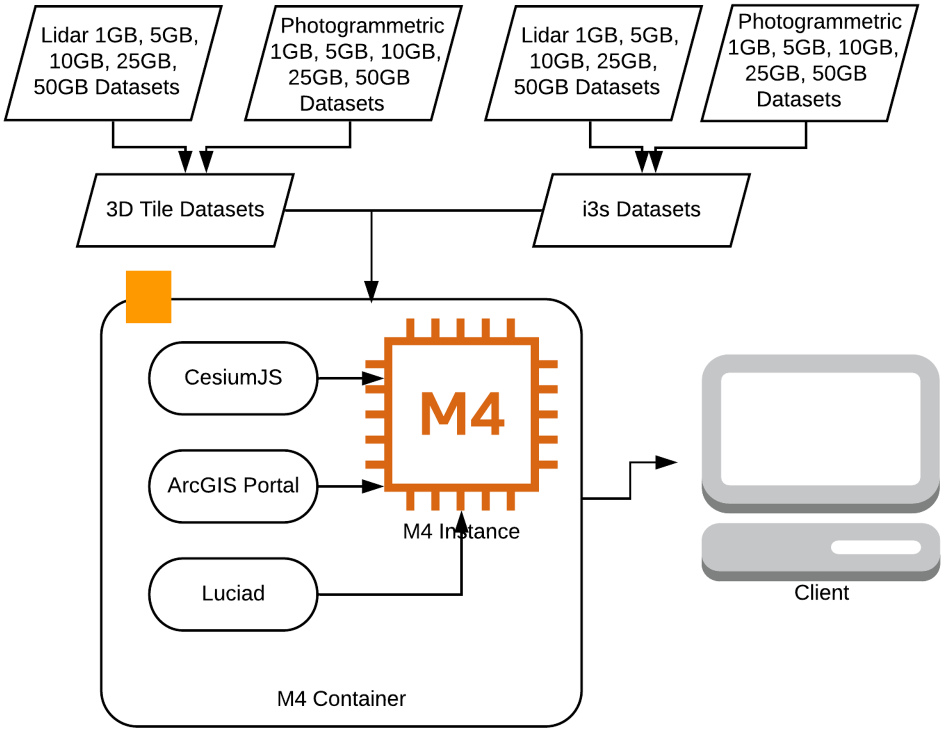

Lidar 1GB, 5GB, 10GB, 25GB, 50GB Datasets
Photogrammetric 1GB, 5GB, 10GB, 25GB, 50GB Datasets
Lidar 1GB, 5GB, 10GB, 25GB, 50GB Datasets
Photogrammetric 1GB, 5GB, 10GB, 25GB, 50GB Datasets
3D Tile Datasets
i3s Datasets
CesiumJS
ArcGIS Portal
Luciad
M4
M4 Instance
M4 Container
Client

Figure 1. Cloud Environment Configuration

The tests performed in this study are shaped from previous cloud computing studies. In a comparison between Map-Reduce and SQL for large data, researchers first tested a singular average query time between the two databases, and then they scaled up the number of queries and analyzed the differences in query load time (Jiang et al. 2009). Another study used open source software to measure CPU performance, and network bandwidth during cloud transactions (Huang et al. 2013) This study implements a hybrid approach. Each dataset was first tested on its overall rendering and resource load time for a singular dataset like in the map reduce study. Other metrics were gathered through GtMetrix of this singular transaction. GtMetrix is being used instead of CPU performance and bandwidth since this is a web application instead of a desktop application. GtMetrix provides performance metrics which report on the performance of web servers. These metrics include Google Page Score, Yahoo YSlow Score, Fully Loaded Time, and the network waterfall of http request and resources used. Fully Loaded Time and the network waterfall are metrics that can be quantitatively analyzed, while Google Page Score and Yahoo YSlow score are more arbitrary figures that rate the overall performance of the website. All these metrics provided a baseline mark of the overall server performance. Next, several tests were run on each exploitation system, for each dataset, where the data set increased in size from 1GB to 50 GB. In these tests, resource loading time and resource memory was recorded. These tests were also performed at three different zoom levels: the extent of the dataset, at a medium zoom level, and at a high zoom level. Esri i3s was tested in Esri ArcGIS Enterprise Portal, and Cesium 3D Tiles were tested within Cesium JS and Hexagon Geospatial Luciad RIA. It is important to run the singular transaction test in addition to the increasing load test (with the increase in dataset size), because some systems are noted to perform slower with small datasets and quicker comparatively with larger data (Jiang et al. 2009). The test of multiple zoom levels was also added to account for changes based on the zoom level. It was identified in preliminary testing that at the full extent only the tileset metadata, and a very low-resolution version of the whole dataset is loaded, while at higher zooms more intricate tiles that reflect the point data are loaded into the application. Because different types of tiles are loaded at different zoom levels, it is important to test each server at various zoom levels. These tests provide an evaluation of the overall server performance for a singular transaction, for increasing data size, for variant zoom level, and for variant spatial resolution of data.

In addition to the quantitative results, a basic qualitative discussion of the visualizations produced by each system was also recorded. The quantitative results provide empirical evidence of the speed of each framework; however, this discounts the visualization product which is output for display in each system. Therefore, it is useful to also include the basic data visualizations within each platform to illustrate core difference in data rendering.

3.2. Datasets

Two different datasets, a dataset collected with lidar, and a dataset collected through photogrammetry were selected as source data for the study to understand effects of spatial resolution on the performance of the exploitation systems. Photogrammetry is collected through matching of point from high-resolution imagery and usually possesses a much more dense and high spatial resolution than lidar data. Lidar data is collected more systematically, and points are sparser than in photogrammetric datasets. Table 1 shows information of the spatial resolution and extents of the datasets utilized in this study.

Table 1: Testing Datasets

Dataset	Spatial Resolution	Spatial Extent	Points	Source
USC Photogrammetry Data	4.50cm	0.37sqkm	2,337,671,969	USC ICT
NOAA Lidar Data	42.00cm	1,657.42 sqkm	1,920,458,367	NOAA

Datasets have been acquired from two different sources. The NOAA dataset was acquired from the NOAA data viewer (OCM 2015). This is an open source repository of las data provided by NOAA in order to analyze damage to New York City after Hurricane Sandy. To download the tiled las NOAA data, a text file with all the file urls was acquired from the NOAA site, and then a python script was created to fetch and download the las files found at each url into a folder on the local system. The USC photogrammetric dataset was acquired from the USC Institute of Creative Technologies. This dataset was re-projected by USC ICT staff from a local coordinate system into a geographic coordinate system and then copied from a remote server onto an external hard drive. Since the study employs testing the dataset at 1GB, 5GB, 10GB, 25GB, and 50GB sizes, each dataset was portioned into several different memory chunks that reflected the 1GB,5GB, 10GB, 25GB, and 50GB memory chunks that were needed to perform the increasing data load test.

Each dataset had to undergo processing into i3s and Cesium 3D Tiles format for ingestion and analysis within the web environment. The general data pre-processing chain is outlined below in Figure 2. Esri ArcGIS Enterprise ships out with an installation of ArcGIS Pro.

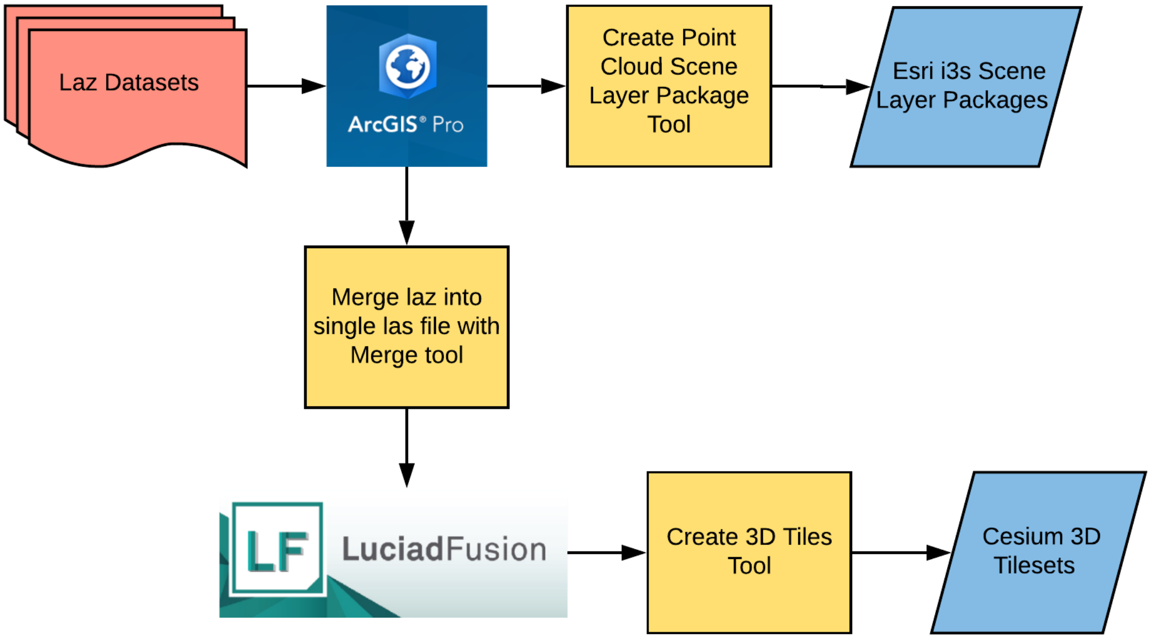

Laz Datasets
ArcGIS® Pro
Create Point Cloud Scene Layer Package Tool
Esri i3s Scene Layer Packages
Merge laz into single las file with Merge tool
LF LuciadFusion
Create 3D Tiles Tool
Cesium 3D Tilesets

Figure 2: Data Conversion Methods

Within ArcGIS Pro is a tool which converts a folder of las files into an Esri i3s data type. Each dataset was converted from las to i3s using the ArcGIS Pro Create Point Cloud Scene Layer Package Tool. Once each dataset was processed into i3s, the Share Package Tool was used in ArcGIS Pro to upload the package to the ArcGIS Enterprise Portal which was setup on the EC2 instance. Configuration of the software and datastore for ArcGIS Enterprise Portal will be delineated in the next section. Once the package was uploaded into the Portal ecosystem, the Publish tool on the Portal website was used to publish the 3D package as a scene service. It is necessary to publish the Portal object as a hosted scene service so that it can be used as an operational layer in web applications. Without publishing the layer as a hosted scene service, the layer is only available for downloading in the Portal ecosystem. In this step, it was noted that publication of a 50GB scene service layers consistently resulted in errors, and therefore no testing was able to be conducted for i3s at tileset size of 50GB. The error is not specific but suggests that Portal does not support tilesets of this size.

In order to create 3D tile services, Hexagon Geospatial Luciad Fusion was used to process all las datasets into 3D Tiles and to setup a datastore to serve them to the Hexagon Geospatial Luciad RIA and Cesium JS applications that were developed on the server. Luciad Fusion is a server software which possess processing and service capabilities. Cesium Ion can also be used to process las files into 3D Tiles, but Luciad Fusion was simply chosen since a license was already acquired with Luciad RIA for the Luciad Fusion software. With Luciad Fusion, all datasets were able to be processed into 3D Tiles and to be made accessible via http requests to the client applications. All data processing was performed on the EC2 instance. All processed data services for both i3s and 3D Tiles were setup on the EC2 instance. Using ArcGIS Enterprise Portal, services were setup for the i3s datasets, and with Luciad Fusion services were setup for the 3D Tile data.

3.3. Virtual Server Design

All server software for the exploitation systems was setup on an Amazon EC2 on demand m4.2xlarge instance. The m4.2xlarge EC2 possess 8 CPU cores, 32 GB of memory, and high network performance. It possesses more than enough computing power necessary to run a server capable of handling large 3D tile datasets. The m4 is a memory intensive server. This server instance was chosen since it has both high compute and memory resources. No GPU is needed on the server side since all 3D graphics are handled on the client device. The server needs high memory and CPU to run most effectively. A 700GB dedicated storage volume was added to host all source data, processed data, and software on the same virtual instance. The dedicated volume is a solid-state drive which also helps boost performance since there are no moving parts in the hard drive.

To setup the EC2 with ArcGIS Enterprise both Esri Cloud Builder and Amazon AWS Cloud Formation tools were attempted for instantiation of an Amazon EC2 with an Amazon Machine Image that contained Esri ArcGIS Enterprise; however, both tools failed. Because of this, creation of the EC2 was a very manual process. The EC2 builder within the Amazon AWS Management Console was used to select an m4.2xlarge EC2 instance with a 700GB storage volume. Next, an Esri ArcGIS Enterprise Amazon Machine Image was selected as the operating system to load onto the EC2 instance. Both Cesium JS and Hexagon Geospatial are libraries that can be downloaded and setup on the EC2; however, an Esri ArcGIS Enterprise Amazon Machine Image is necessary to setup ArcGIS Enterprise Portal since it is baked into the operating system of the EC2 instance. After the Esri Enterprise Amazon Machine Image was chosen, the EC2 production process started.

After the EC2 had been initialized, several configurations were required to setup the machine to act as a web server. The domain kevinmercythesis.com was purchased form Google Domains, and then using sslforfree was setup for secured http traffic. Setting up a secured http (https) connection was required to obtain a ssl certification for the domain kevinmercythesis.com. A ssl cert is required to run Esri ArcGIS Enterprise, and therefore the domain purchase and generation of a ssl cert to allow https traffic on the domain was required to setup Esri ArcGIS Enterprise. Once the ssl certification was obtained, web site forwarding was setup on the domain to point the domain site to the Amazon EC2 instance. To do this, an elastic IP address was created for the virtual amazon instance so that it would always possess the same public IP address. The elastic IP address created within the Amazon AWS Management Console was set as the forwarding address for the kevinmercythesis.com domain. Next, using an Amazon Cloud Formation template provided by Esri, a Virtual Private Cloud was created within for the EC2 Elastic IP address and attached to it. The IP address within the virtualized hardware does not match its external IP address and therefore a Virtual Private Cloud and an Elastic IP Address are required to allow proper trafficking of web applications from the EC2 instance to a secured external domain.

Next, a new user was created to allow the ArcGIS Enterprise Entity to make changes to the Amazon EC2 instance, and several changes were implemented within the Windows Firewall and the Amazon EC2 Security Groups to allow access of specific ports within the EC2 system to the outside web. With the purchase of a domain and a ssl cert, a webpage domain was setup which could receive web traffic from the EC2 instance. To correctly point the EC2 instance to the purchased domain, first creation of an elastic IP was initiated. Then a virtual private cloud was created and attached to the elastic IP address. Lastly, ports were opened within the windows firewall manage and within the Amazon security group for the EC2 instance to allow access of certain ports to the external web. These configurations established the frameworks necessary to run the web applications outside of the server and to begin setup of Esri ArcGIS Enterprise on the EC2 instance.

3.4. Comparative Program Architecture

Each 3D GIS exploitation platform possesses different software architectures and system configurations. Both Esri ArcGIS Enterprise and Hexagon Geospatial Luciad RIA are commercial software suites that must be licensed for use, while Cesium JS is an open source software package which requires no licensing and can be used simply by importing it as a content delivery network into a JavaScript web application. Because it is open source and requires no licensing, Cesium JS is the lightest and simplest to configure for use. Cesium JS can simply be imported within a JavaScript application by including these two lines of code within an html file. The two lines of JavaScript code import the main Cesium "library" and

```
<script src="https://cesium.com/downloads/cesiumjs/releases/1.66/Build/Cesium/Cesium.js"></script>

<link href="https://cesium.com/downloads/cesiumjs/releases/1.66/Build/Cesium/Widgets/widgets.css" rel="stylesheet">
```

Code 1: Cesium Import Script (Cesium n.d.)

the main library for the Cesium.js widgets. These two files are minified (or compressed) JavaScript files which handle API calls for all Cesium.js elements (i.e. the virtual globe, data sources on the globe, and all functions that are available in the API to programmatically interact with the globe module). Because of its library compression, and small number of library imports (only two modules), Cesium.js is a lightweight addition to a JavaScript application. Due to their licensing requirements and more extensive use of libraries, Esri ArcGIS Enterprise and Hexagon Geospatial Luciad RIA are more bulky applications than CesiumJS.

Esri ArcGIS Enterprise and Hexagon Geospatial Luciad RIA are both commercial software suites, but possess vastly different setups. Esri ArcGIS Enterprise requires a specialized Amazon Machine Image, where Esri ArcGIS Enterprise resources are basically baked into the operating system of the cloud instance. This is a vastly different configuration then either

Cesium JS or Hexagon Geospatial. Cesium JS and Hexagon Geospatial can be run in either cloud-based or non-cloud-based environments. Esri ArcGIS Enterprise requires a cloud-based virtual machine, and gets directly installed as core components to the cloud virtual machine. Esri ArcGIS Enterprise uses an array of applications to setup an environment which possesses a front-end website, a backend GIS server, and various other GIS data processing or management applications. Esri ArcGIS Enterprise is a sophisticated software suite that is geared towards hosting enterprise GIS resources for 2D and 3D data. In addition to providing the ability to host web services for 2D and 3D data, Esri ArcGIS Enterprise can also be used to create web applications, host web applications, and perform various geoprocessing computations. Esri ArcGIS Enterprise is a full-scale web GIS implementation. Because it implements many more functions and capabilities than Cesium JS or Hexagon Geospatial RIA, it is a much larger installation than either of these exploitation frameworks.

Hexagon Geospatial Luciad RIA, is essential a composition of JavaScript libraries which can be used for 2D and 3D web GIS. The focus of Hexagon Geospatial Luciad RIA is more on 3D data, but it possesses libraries for serving and interacting with 2D data sources as well. The Hexagon Geospatial Luciad RIA suite is also more of a developmental suite than a production software like Esri ArcGIS Enterprise. Hexagon Geospatial Luciad RIA comes packaged as a collection of JavaScript libraries. Creating a Luciad RIA application involves addition of core libraries which have been created for many general GIS functions, and then customization of those functions to create an application which meets the needs of the user. Examples of basic library functions include creating a view with a 3D virtual globe, adding a 2D raster to the globe, adding a 3D tileset to the globe, etc. These basic functions can essentially be imported into a new application and then customized to serve the user requirements. Hexagon Geospatial Luciad RIA resembles the architecture of Cesium JS more, but is much bulkier since it is a collection of libraries and not a singular library like Cesium JS.

Esri ArcGIS Enterprise is a collection of applications which function harmoniously to setup data service, hosting, and complex geoprocessing operations on a cloud instance. Cesium JS is a content delivery network, a single library, that allows users to create and interact with virtual globes and 3D data. And Hexagon Geospatial Luciad RIA is a composition of many JavaScript libraries which can be configured for working with 2D or 3D geospatial data and web maps or virtual globe interfaces.

3.5. Esri ArcGIS Enterprise Configuration

To instantiate an Amazon EC2 instance with the Esri ArcGIS Enterprise, Esri Cloud Formation templates were used. The “Create an Amazon VPC” cloud formation template and “Single-machine deployment” from the Esri Cloud Formation Templates site were used for Portal creation (Esri n.d.4). Originally the Esri Cloud Builder tool was attempted to create the EC2 with the ArcGIS AMI, but after a couple failed attempts, the Cloud Formation templates were used instead. With the “Create an Amazon VPC,” a virtual private cloud was successfully created. There were issues experienced in using the “Single-machine deployment” template, in the first attempt. To get around the errors from the “Single-machine deployment” template, the rollback resources on error option in Cloud Formation was turned off and then the template was rerun. Once the Cloud Formation template reached an error in the second trial, the created EC2 instance was still available and used to finish software setup. The EC2 was then logged into and configurations to the ArcGIS Server Site, ArcGIS web adopters, ArcGIS Portal, and data store wizard were conducted manually to finish installation and configuration of Esri ArcGIS Enterprise.

While the Cloud Formation Template ran, several resources and configurations for ArcGIS Enterprise were configured successfully. An Amazon EC2 with the Esri Enterprise software and a 700GB solid state drive was created successfully. The Portal software was also licensed correctly, and the ssl cert for https trafficking was installed. A virtual private cloud and elastic IP address were also created, allowing access of the EC2's public IP address to http, and https traffic. Manuel modifications required to finish setup included configuration of the ArcGIS Server site, ArcGIS web adaptors, ArcGIS Portal site, and the ArcGIS data store wizard, which are described below.

After instantiation and modification of elastic IP address, virtual private cloud, security groups, and users, several manual processes were conducted to properly setup ArcGIS Server and ArcGIS Enterprise so that ArcGIS Enterprise Portal could be used to create hosted feature services on the Amazon EC2 instance. First, the ArcGIS Server Authentication Manager was run in order to license ArcGIS Server. During this process a user called siteadmin was created which had permissions over ArcGIS Server resources on the Amazon EC2 instance. During Amazon Machine Image Instantiation, a user was already created for Esri ArcGIS Enterprise Portal. Next the Esri Web Adapter for Portal and Esri Web Adapter for Server were used to register the ArcGIS Enterprise Portal to the site https://kevinmercytheis.com/portal and the ArcGIS Server to https://kevinmercythesis.com/server. These web adopters also co-registered the ArcGIS Server and ArcGIS Portal software so that ArcGIS Server could be set as the hosting site for the Esri ArcGIS Portal. To initiate this change, the ArcGIS server site was selected as the federated server for the ArcGIS Portal site, and then this server was selected as the hosting server for the ArcGIS Portal. These changes were required in order to enable publishing hosted feature layers with the Portal site. Without federating the ArcGIS Server site and setting it as the hosting server, there is no way to publish scene services as operational web layers. The ArcGIS Server site and the ArcGIS Portal site were setup on the same instance so that the computing environment for the hosted Esri i3s layers would be the same computing environment as the Luciad Fusion 3D Tiles. After federation of the ArcGIS Server site, and the ArcGIS Portal site, the ArcGIS Portal could be accessed and used to publish operational scene service layers using the Amazon EC2 as the hosting server.

Once the Amazon EC2 instance was properly configured with ArcGIS Server and ArcGIS Portal sites, all the processed i3s data were published using the ArcGIS Portal site. As indicated earlier, the Portal site had issues publishing the 50GB scene layer packages and therefore they were omitted from testing.

Lastly, a web application was created out of port 3002 on the Amazon EC2 instance in order to create an application which served the 3D scene service layers for testing. Using node.js and npm, a basic npm express web server was used to open up port 3002 and attach the Esri JavaScript code to it. Node.js with npm had to be installed on the Amazon EC2 to run the web server out of port 3002. Using the Esri JavaScript API, an index.html file was created which created a web application that pulls an Esri globe and plots a scene service layer on it. Additional programming within the application was used through the study to customize the zoom level of the application and the dataset that was hosted from it. Throughout testing the zoom level of the application, and the rendered Scene Service Layer were modified to test the application for all specified tests. For each test conducted, the resource memory and the resource loading time were tested five times and the average of the tests was stored as the result.

3.6. Hexagon Geospatial Luciad Fusion Setup

Hexagon Geospatial Luciad Fusion license and software were obtained from a remote server. Setup of Hexagon Geospatial Luciad Fusion was much more straightforward than Esri ArcGIS Server and Portal. Once the Fusion files were transferred to the Amazon EC2 instance, a setup program was run which setup Fusion on the EC2 instance. Once the software was installed, the Fusion application was initiated. The Fusion application opened a console, a management portal, and started web services of the software on port 8081. A Fusion admin user was created and then processing of las data into 3D Tiles began. To process the 3D tiles, the tiles to process were selected and then a tool popped up to initiate creation of a 3D Tile service from those source tile. A service title, abstract, and data type were required for input. In this dialog, the data type is set as OGC 3D Tiles and a name was chosen for each service. Upon clicking the submit button, the software begins processing the source data into 3D Tiles, and when it is finished it sets up an entry point url from port 8081 which can be used by web applications to access the tile data. All source datasets were processed into 3D tiles in this fashion. The Luciad Fusion software created 3D tilesets of each dataset, and service urls on the Amazon EC2 that could be used by the Cesium JS and Luciad RIA web applications.

3.7. Hexagon Geospatial Luciad RIA

Similarly, to Hexagon Geospatial Luciad Fusion, license files and software were obtained from a remote server and then brought onto the EC2 instance. Setup of Luciad RIA was fairly simple. Most of the technical work in setting up Luciad RIA was in the programming of a JavaScript application to have Luciad RIA create a virtual globe and then display a 3D tile dataset. To license the Luciad RIA API, the license file is simply placed in a specific directory within the Luciad RIA file structure. Luciad RIA comes with a sample Jetty server that can be used to run Luciad applications out of port 8072. The sample server was used to run the RIA web application.

In order to create a Luciad RIA application, extensive customization from Luciad RIA sample applications was performed to create an application that simply created a 3D web GL virtual globe and then displayed a 3D tileset on it. A Luciad RIA sample application called firstsample was copied and renamed. Then its contents were modified to create a basic virtual globe interface. In order to generate a 3D tileset within the globe, another sample application called Data Types was used to program the 3D Tile display on the virtual globe. The Data Types application had a component called OGCLoader which was code to load a 3D tileset from a url and colorize it. This module was added into the ongoing Luciad RIA application. Another function, called addLayer was also taken from the Data Type application and then placed in the ongoing Luciad RIA application. With the OGCLoader module and the addLayer function, the ongoing Luciad RIA application could now be used to create a virtual globe, add a 3D tileset to it, and zoom to that location. Code was later found in order to customize zoom and navigation of the virtual globe. With all these components, a Luciad RIA application was created that could be used to zoom to different levels of a 3D tileset on a virtual globe. This application was then run from port 8072 using the Luciad RIA simple server and used to run all the performance tests.

3.8. Cesium JS Setup

Cesium JS is available as a content delivery network in JavaScript since it is open source. Therefore, no licensing is required to create a Cesium JS application. A basic program was created which retrieves the Cesium JS content delivery network and then creates a basic virtual globe interface which can be used to visualize 3D Tiles. A Cesium ion account is required in order to add a basemap and layers to the virtual globe. A Cesium ion account is free and provides a developer token that must be placed within the Cesium JS Application for it to run correctly. An ion account was created and then the key was placed in the application along with the code to create a Cesium virtual globe, and to add a tileset to it. With these customizations, a web application developed with Cesium JS was synthesized that could render a virtual globe and 3D tilesets. Additional programming could be added in to change the zoom level of the viewer as well. Similarly to the Esri ArcGIS Application that was setup with node.js and npm express to run out of port 3002, an npm express server was developed to run the Cesium web application out of port 3001 on the Amazon EC2 instance.

3.9. Testing and Results

After setting up all the software, three web applications were created which could be used to test each exploitation system. All Esri i3s datasets were published within the Esri ArcGIS Portal and available as scene services (i.e. https://kevinmercythesis.com/server/rest/services/Hosted/USC_TwentyFive_SLPK_Hosted/SceneServer). All 3D Tiles data had been processed with Luciad Fusion and made available as services out of port 8081 (i.e. http://kevinmercythesis.com:8081/ogc/tiles/noaa5gb/tileset.json). The Esri JavaScript API testing application was running out of http://kevinmercythesis.com:3002. The Cesium JS Application was running out of http://kevinmercythesis.com:3001. The Luciad RIA application was running out of http://kevinmercythesis.com:8072/web/samples/webapp/index.html. With all the i3s and 3D tile services running, and the testing web application setup on accessible ports, testing was able to be conducted for each server framework. First, each web application was configured to host a 1GB dataset and then the url of the application was placed in the GtMetrix tool in order to run tests for Google Page Score, Yahoo YSlow, Total Page Load, and to synthesize the network waterfall.

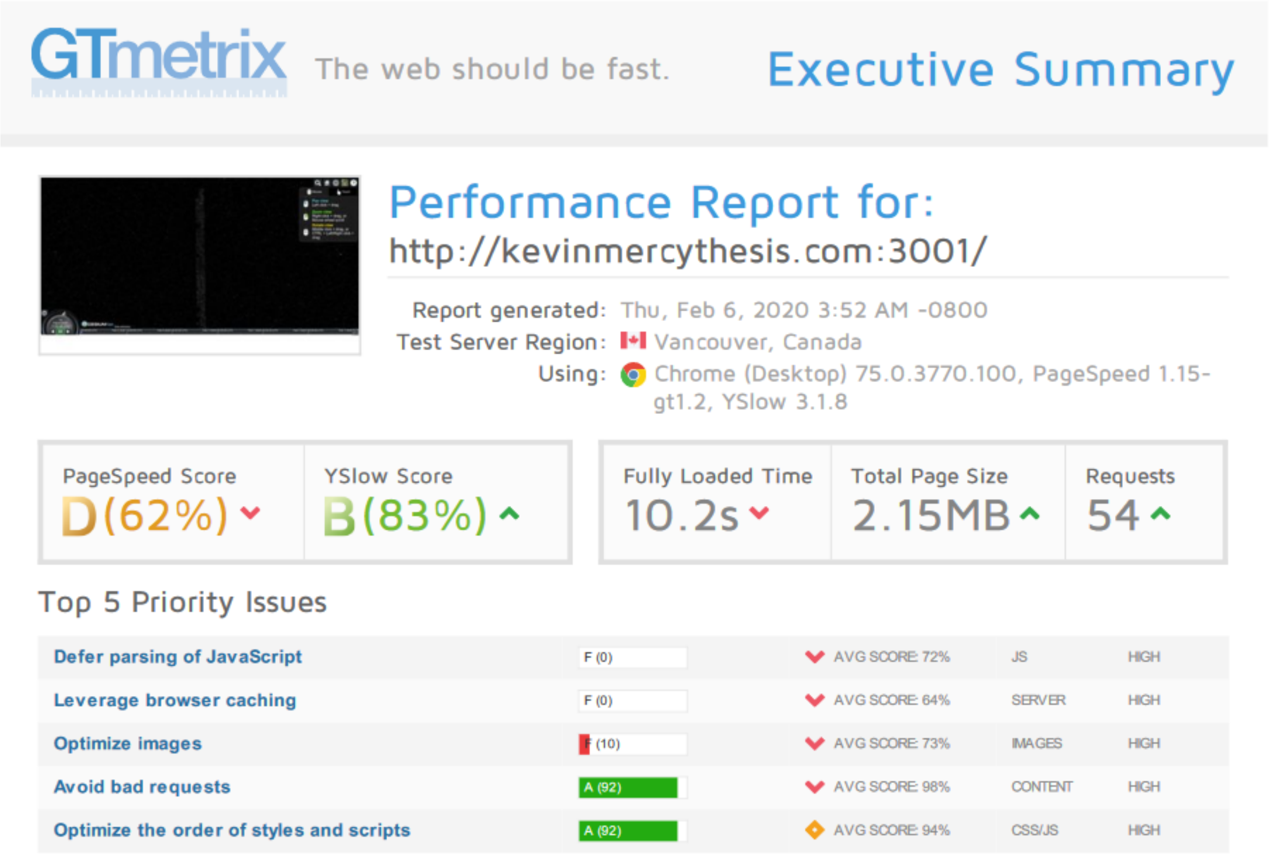
The reports from GtMetrix from each web application with a 1GB dataset were saved, and then testing of resource loading time and memory for variant dataset, tileset size, and zoom level were performed. For each exploitation system, first the photogrammetric dataset was tested at each memory size (1GB, 5GB, 10GB, 25GB, and 50GB) and at three zoom levels (dataset extent, medium zoom level, and high zoom level). For every test, the test was performed five times, due to the virtualization of the cloud hardware, and then the mean of the test was stored in a master sheet. Once all tests had been performed for the photogrammetric dataset, the same workflow was followed for the lidar dataset. While the testing was ongoing screenshots were captured of the visualizations that were rendered by each dataset at various zoom levels.

Chapter 4 Results

Reports from GtMetrix and from the increasing data load tests demonstrate that generally Cesium JS and Esri ArcGIS Enterprise Portal are quicker and use less memory than Hexagon Geospatial Luciad RIA. From simply the GtMetrix results it is difficult to determine if Cesium JS or Esri ArcGIS Enterprise are quicker. Cesium JS obtains a higher Yahoo YSlow score then Esri ArcGIS Enterprise; however, Esri ArcGIS Enterprise has a higher Google Page Score than Cesium JS. From the loading tests, it appears that for both datasets Cesium JS had the lowest loading time, but with larger datasets the difference in load time between Cesium JS and Esri ArcGIS Enterprise are basically negligible. For the most part, Cesium JS always uses the least memory, but at high zoom levels with lidar data Esri ArcGIS Enterprise used the least memory, but its load time was still slightly slower. In all cases, Hexagon Geospatial Luciad RIA was the slowest and uses the most memory. Although Hexagon Geospatial Luciad RIA was much slower and used more memory, it had the most photorealistic visualization compared to the other platforms. Even though Cesium JS for the most part was always quickest and lightest, it also rendered with much less visual detail than Hexagon Geospatial Luciad RIA, and Esri ArcGIS Enterprise. It does not appear overall tile set size influenced performance, as resource loading time and memory were generally consistent across tilesets. Resource loading time and memory used appeared more dependent on zoom level. At higher zoom levels, generally more time and resources were used. Dataset spatial resolution also did not seem to effect performance. Loading time and resources used were fairly consistent between the high-resolution photogrammetric dataset and the lidar dataset.

4.1. GtMetrix Results

Results from GtMetrix show that Cesium JS and Esri ArcGIS Enterprise server frameworks were generally quicker and used less memory than Hexagon Geospatial Luciad RIA. Figures 3 and 4 depict the results from GtMetrix for Cesium JS and Esri ArcGIS Enterprise.



GTmetrix The web should be fast. Executive Summary

Performance Report for:
http://kevinmercythesis.com:3001/

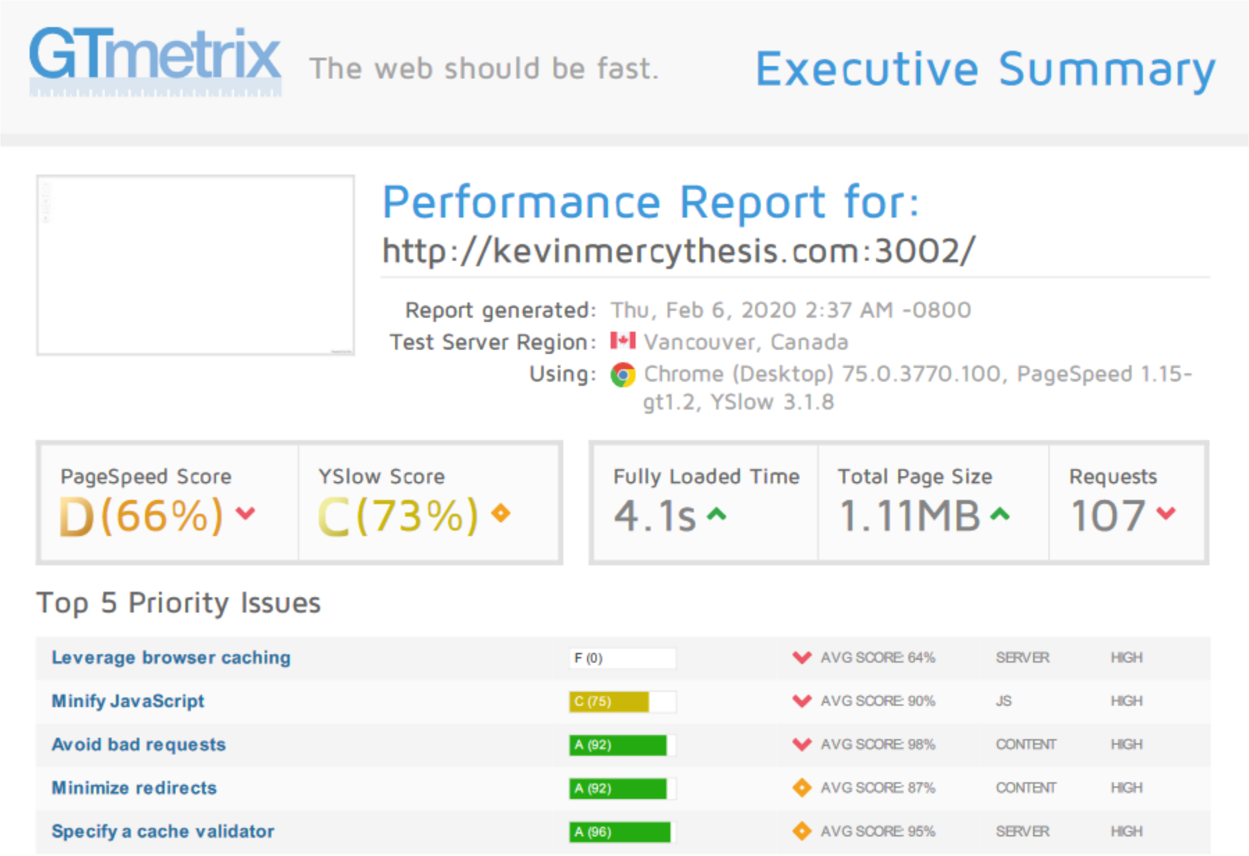
Report generated: Thu, Feb 6, 2020 3:52 AM -0800
Test Server Region: Vancouver, Canada
Using: Chrome (Desktop) 75.0.3770.100, PageSpeed 1.15-gt1.2, YSlow 3.1.8

PageSpeed Score	YSlow Score	Fully Loaded Time	Total Page Size	Requests
D(62%) ▼	B(83%) ▲	10.2s ▼	2.15MB ▲	54 ▲

Top 5 Priority Issues

Defer parsing of JavaScript	F (0)	▼ AVG SCORE: 72%	JS	HIGH
Leverage browser caching	F (0)	▼ AVG SCORE: 64%	SERVER	HIGH
Optimize images	F (10)	▼ AVG SCORE: 73%	IMAGES	HIGH
Avoid bad requests	A (92)	▼ AVG SCORE: 98%	CONTENT	HIGH
Optimize the order of styles and scripts	A (92)	◆ AVG SCORE: 94%	CSS/JS	HIGH

Figure 3: Cesium JS GtMetrix Results



GTmetrix The web should be fast. Executive Summary

Performance Report for:
http://kevinmercythesis.com:3002/

Report generated: Thu, Feb 6, 2020 2:37 AM -0800
Test Server Region: Vancouver, Canada
Using: Chrome (Desktop) 75.0.3770.100, PageSpeed 1.15-gt1.2, YSlow 3.1.8

PageSpeed Score	YSlow Score	Fully Loaded Time	Total Page Size	Requests
D(66%)	C(73%)	4.1s	1.11MB	107

Top 5 Priority Issues

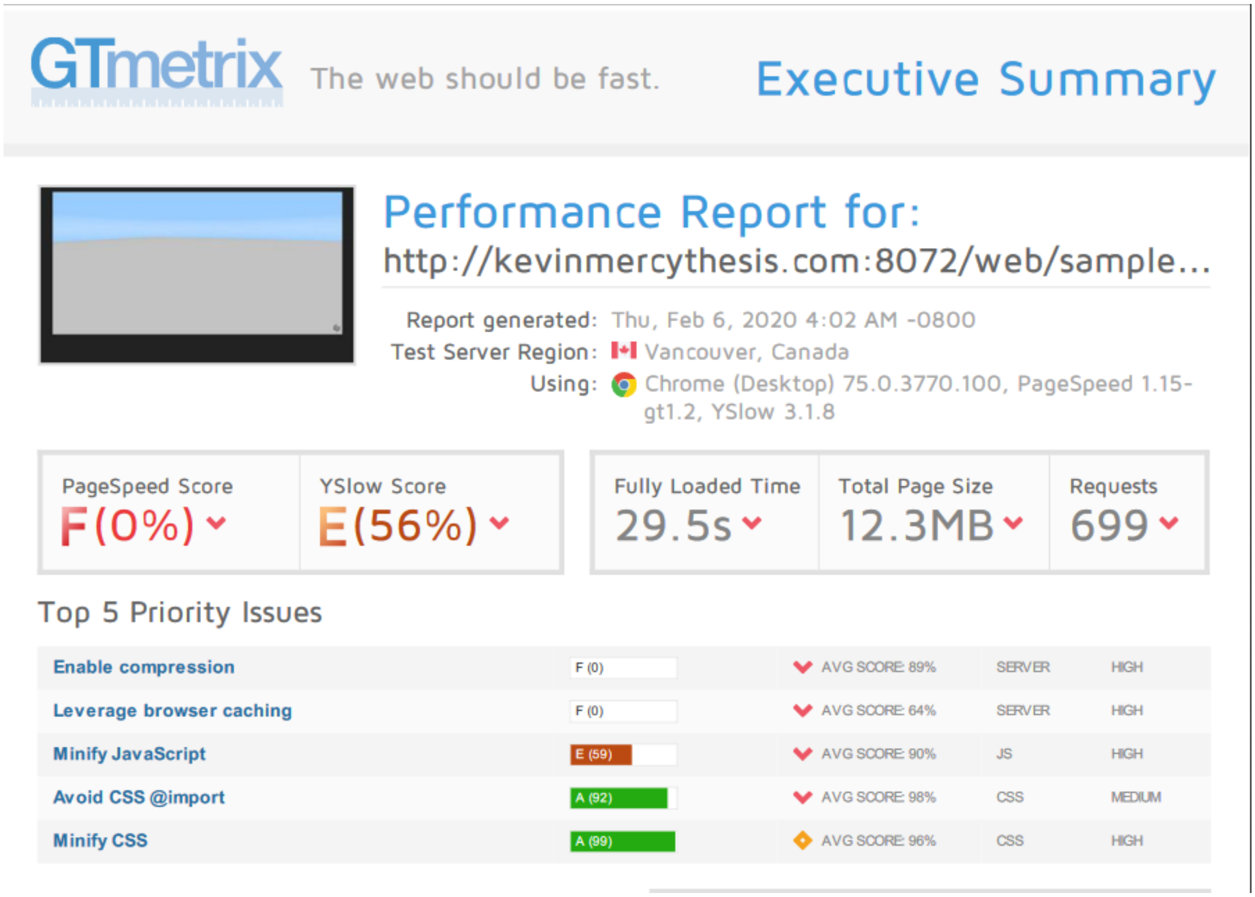
Leverage browser caching	F (0)	AVG SCORE: 64%	SERVER	HIGH
Minify JavaScript	C (75)	AVG SCORE: 90%	JS	HIGH
Avoid bad requests	A (92)	AVG SCORE: 98%	CONTENT	HIGH
Minimize redirects	A (92)	AVG SCORE: 87%	CONTENT	HIGH
Specify a cache validator	A (96)	AVG SCORE: 95%	SERVER	HIGH

Figure 4: Esri ArcGIS Enterprise GtMetrix Results

GtMetrix ranked Cesium JS with a higher YSlow score of 83% compared to a 73% score for Esri ArcGIS Enterprise, but ArcGIS Enterprise was ranked with a higher Google Page Speed Score of 66% compared to a 62% for Cesium JS. The Google Page Speed scores were similar. Empirically, Cesium JS had a higher fully loaded time of 10.2s, and total page size of 2.15, but a much smaller number of requests. Cesium JS only made 54 http requests compared to 107 for Esri ArcGIS Enterprise. Both the applications lacked browser caching, which is one of the critical issues with their generally low Google Page Score rankings. In their most minimal setup, Cesium JS and Esri ArcGIS Enterprise appear to perform very similar in overall ranked performance. Fully loaded time and page size from GtMetix suggest Esri ArcGIS Enterprise to be smaller and more lightweight then Cesium JS. Both servers appear to be slow overall though.

Implementing more browser side caching for both of these servers would likely greatly lower their load time and page size, and thereby increase their Google Page Score and Yahoo YSlow scores.

The GtMetrix results for Hexagon Geospatial Luciad RIA show it to be slower and using much more memory then Cesium JS or Esri ArcGIS Enterprise. Figure 5 shows the results of the Hexagon Geospatial Luciad RIA GtMetrix results.



Executive Summary

Performance Report for:
http://kevinmercythesis.com:8072/web/sample...

Report generated: Thu, Feb 6, 2020 4:02 AM -0800
Test Server Region: Vancouver, Canada
Using: Chrome (Desktop) 75.0.3770.100, PageSpeed 1.15-gt1.2, YSlow 3.1.8

PageSpeed Score	YSlow Score	Fully Loaded Time	Total Page Size	Requests
F(0%)	E(56%)	29.5s	12.3MB	699

Top 5 Priority Issues

Enable compression	F (0)	AVG SCORE: 89%	SERVER	HIGH
Leverage browser caching	F (0)	AVG SCORE: 64%	SERVER	HIGH
Minify JavaScript	E (59)	AVG SCORE: 90%	JS	HIGH
Avoid CSS @import	A (92)	AVG SCORE: 98%	CSS	MEDIUM
Minify CSS	A (99)	AVG SCORE: 96%	CSS	HIGH

Figure 5: Hexagon Geospatial Luciad RIA GtMetrix Results

GtMetrix ranked Hexagon Geospatial Luciad RIA with a 0% Google Page Score and a 56% for Yahoo YSlow. The Google Page Score was 0% due to the lack of compression utilized with the application. Both Cesium JS and Esri ArcGIS Enterprise implement compression. The lack of compression may be because Hexagon Geospatial Luciad RIA is largely a development library and the application is more developmental facing than production facing. Nonetheless, it appears that most of the extra time and resources used by Luciad RIA came from its large amount of library loading. 699 http request are made, which were mostly components within the Luciad RIA development library. The large overhead to call RIA components is likely one of the reasons Luciad RIA took longer to load and used more memory. Luciad RIA used significantly more memory, 12.3 MB, compared to 2.15 MB for Cesium JS, and 1.11 MB for Esri ArcGIS Enterprise, and more time 29.5 s compared to 10.2 s for Cesium JS, and 4.1 s for Esri ArcGIS Enterprise. Luciad RIA also did not implement browser caching. All three of these exploitation systems may have better performance with implementation of browser side caching.

4.2. Increasing Loading Tests

4.2.1. Resource Loading Time for Photogrammetric Data

For both photogrammetric datasets and the lidar datasets, it appears Cesium JS performed the quickest, and Hexagon Geospatial Luciad the slowest. At large tileset sizes, the difference between Cesium JS and Esri ArcGIS Enterprise is almost negligible. Figures 6 – 8 display the results from the resource loading time tests for photogrammetric data.

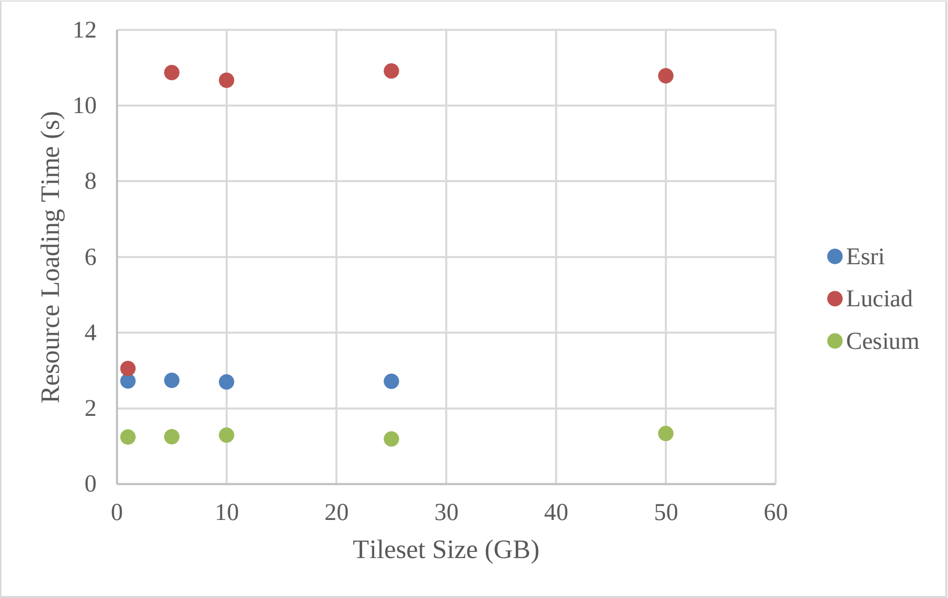


Figure 6: Resource Loading Time for Photogrammetric Dataset at Full Extent

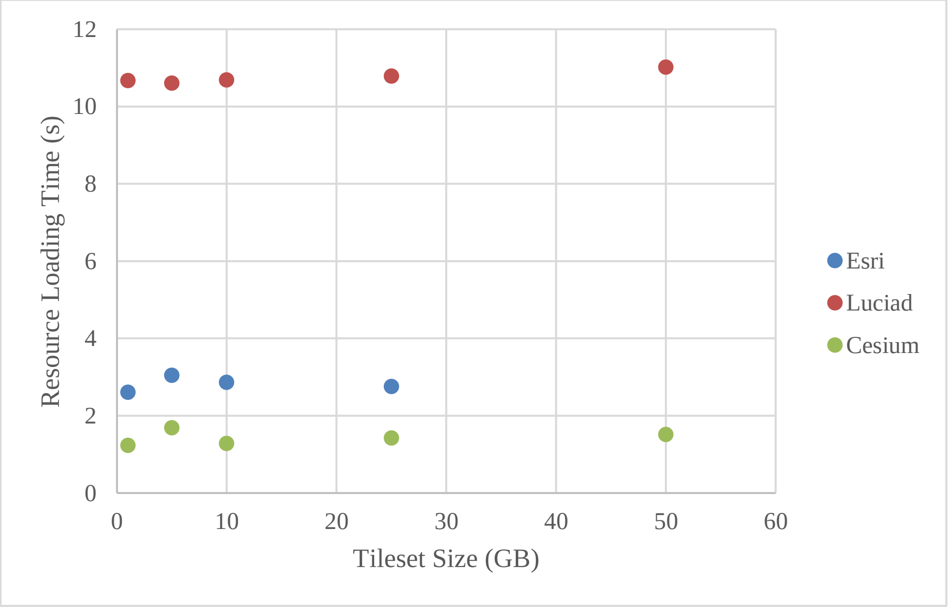


Figure 7: Resource Loading Time for Photogrammetric Dataset at Medium Zoom

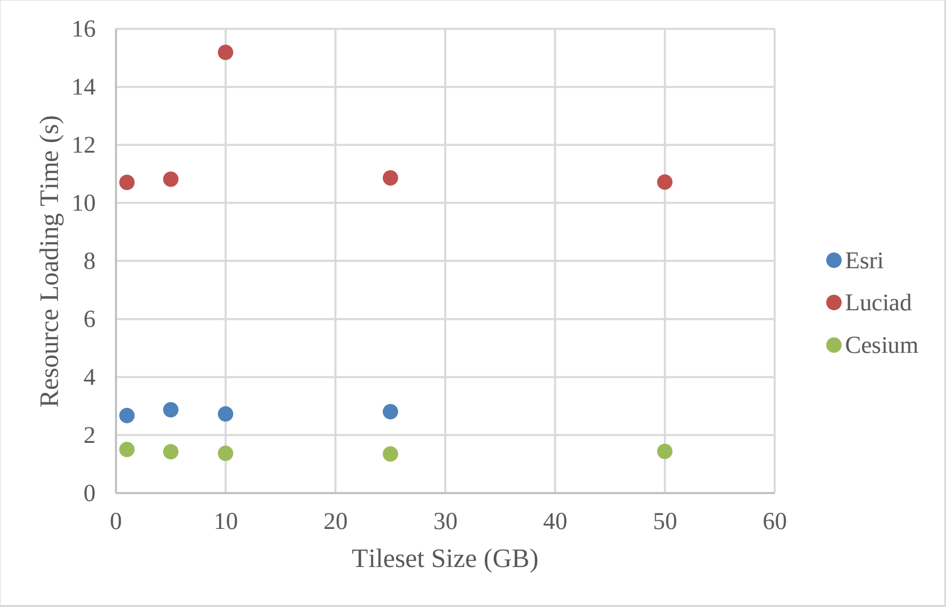


Figure 8: Resource Loading Time for Photogrammetric Dataset at High Zoom

At the full extent of the dataset, there is an anomalous result for the 1GB photogrammetric test of the Hexagon Geospatial Luciad RIA software. Since this data point is at the full extent, the quick load time may be due to a smaller number of library requests since tiles are not necessarily being in at this extent. Further testing is required though. Nonetheless, besides this data point, Hexagon Geospatial Luciad RIA consistently takes more than 10s to load resource at all zoom levels for all tileset sizes. Cesium JS generally loaded in just under 2s for all tileset sizes at all zoom levels. And Esri ArcGIS Enterprise took just above 2s seconds for all tilesets at all zoom levels. The resource time between Cesium JS and Esri ArcGIS Enterprise appear very similar.

4.2.2. Resource Memory for Photogrammetric Data

Tests for resource memory show that resource memory increased as zoom level increased. Hexagon Geospatial Luciad RIA was still slowest, and Cesium JS still used slightly less memory than Esri ArcGIS Enterprise. Tileset size did not seem to impact the resource sizes.

Figure 8 – 10 depict results of Resource Memory at the three zoom levels for the photogrammetric dataset.

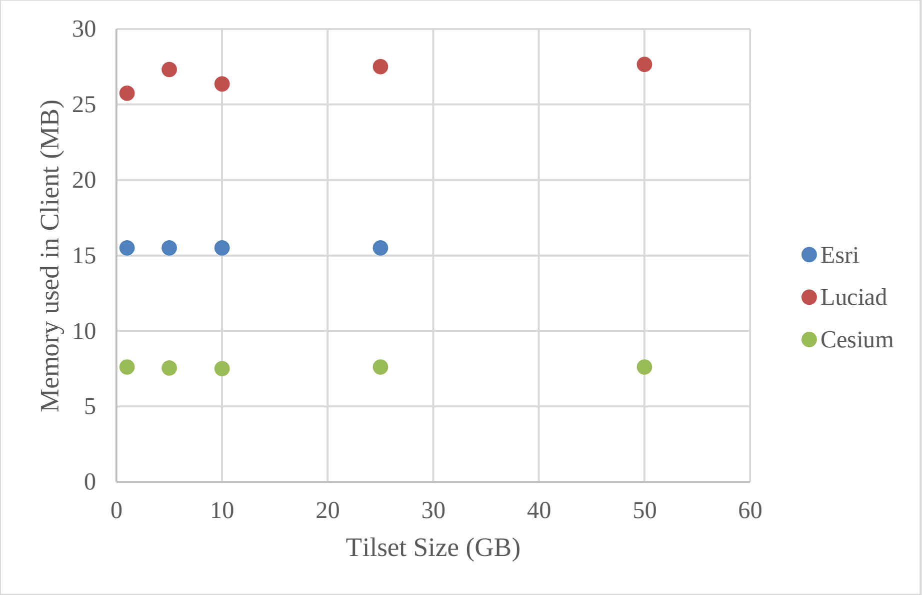


Figure 9: Resource Memory for Photogrammetric Dataset at Full Extent

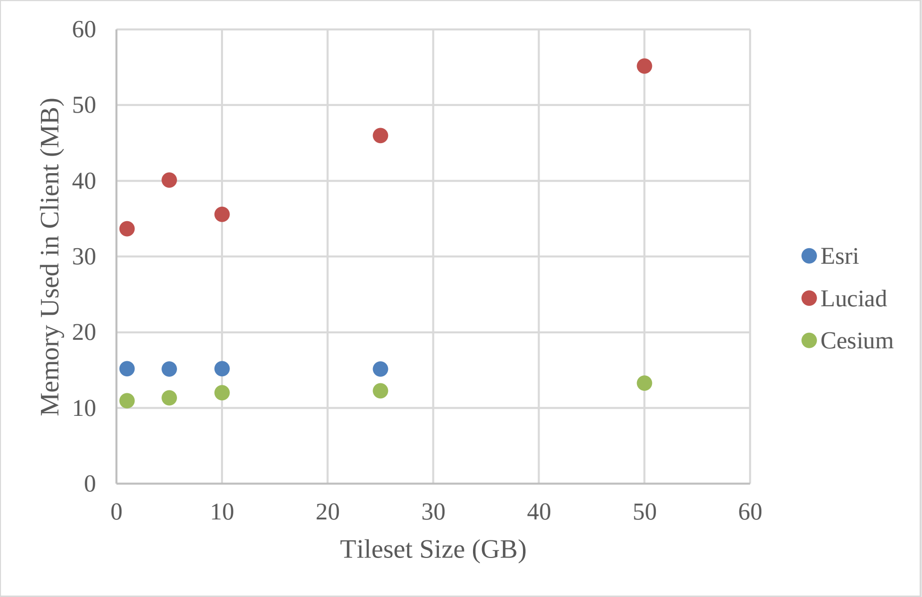


Figure 10: Resource Memory for Photogrammetric Dataset at Medium Zoom

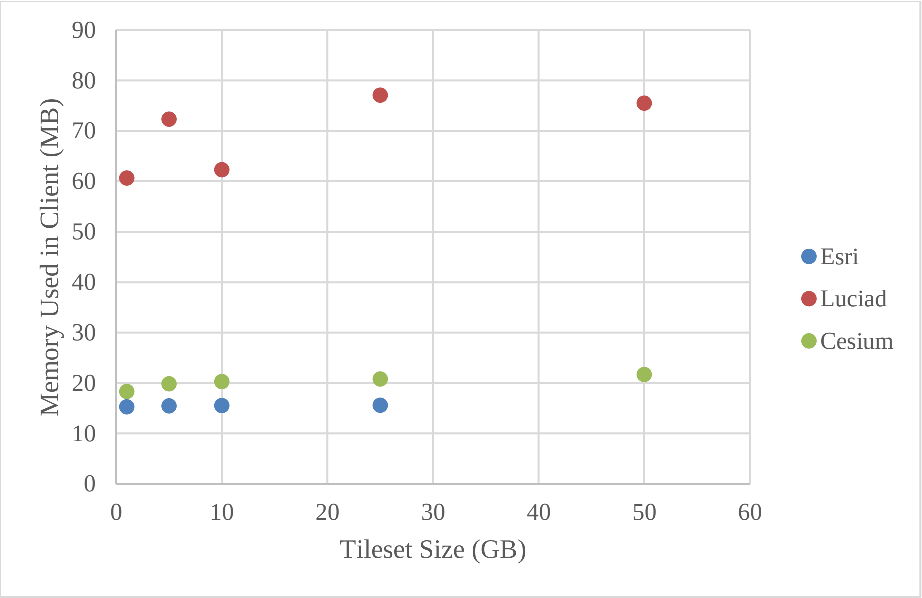


Figure 11: Resource Memory for Photogrammetric Dataset at High Zoom

As zoom increased Esri ArcGIS Enterprise and Cesium JS became very similar. At low level zoom, Cesium JS used very little resources, but as the zoom increased the memory used between Cesium JS and Esri ArcGIS Enterprise almost leveled out. Cesium JS still used the least memory; however, there is almost no difference in memory used between Cesium JS and Esri ArcGIS Enterprise at high zoom levels.

It is very curious the large rise in memory used for Hexagon Geospatial at the Medium Zoom levels across tileset size, since generally at the other zoom levels there is less variation in resources used across tileset size. The trend may arise much quickly in this zoom level compared to the other zoom levels, in that the location of the zoom may be in proximity to tiles which are existent in only the larger tilesets; therefore, causing a large increase in client memory usage. The trend may not be apparent at the full extent since generally a low resolution of the extent is loaded in the full extent of the dataset, and maybe the location in the high zoom loads tiles which are existent in each of the tilesets, and therefore no additional tiles are required to be loaded as the tileset size increases. The reason for the trend requires further testing in order to be completely understood.

4.2.3. Resource Loading Time for Lidar Data

Resource loading time for lidar data at three zoom levels for each tileset size appeared very similar to the results from the photogrammetric data. Resource loading times were still very similar, around 10s for Hexagon Geospatial Luciad RIA, 2s for Cesium JS, and around 2s for Esri ArcGIS Enterprise.

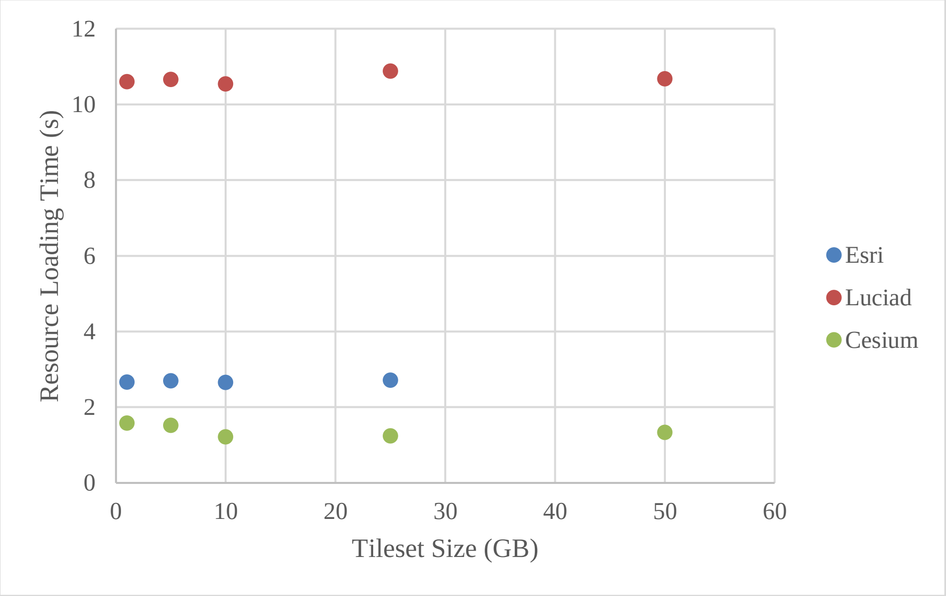


Figure 12: Resource Loading Time for Lidar Dataset at Full Extent

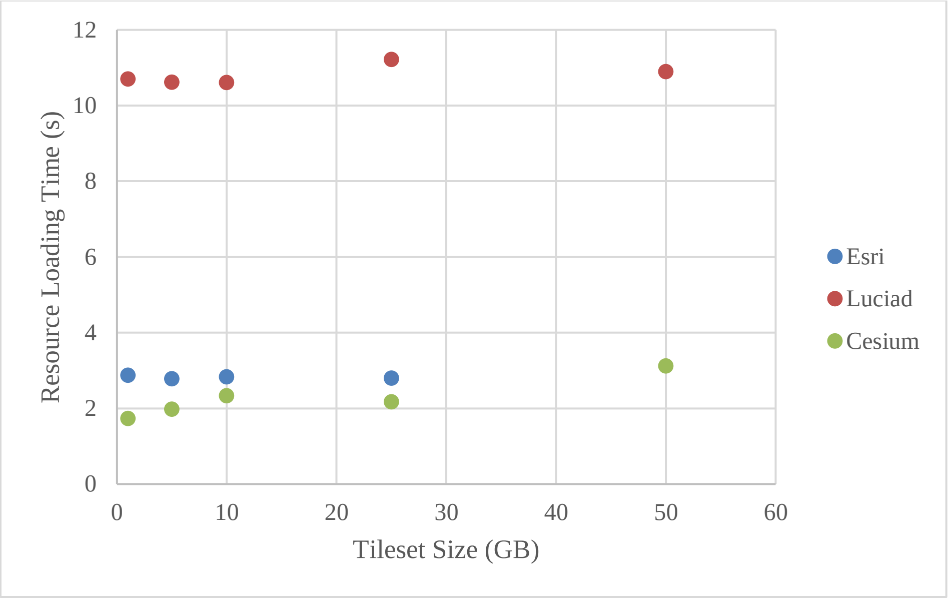


Figure 13: Resource Loading Time for Lidar Dataset at Medium Zoom

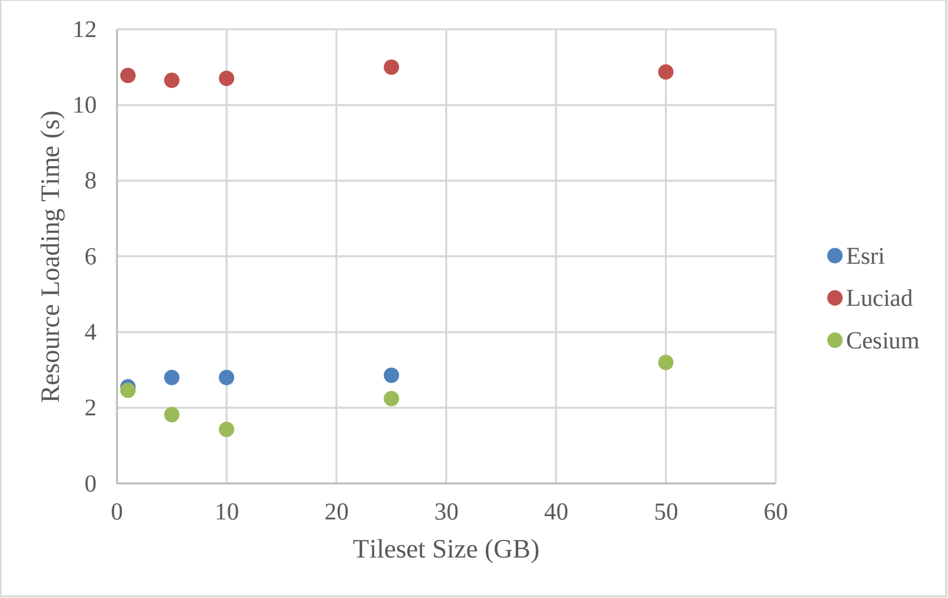


Figure 14: Resource Loading Time for Lidar Dataset at High Zoom

4.2.4. Resource Memory for Lidar Data

The resource memory results show some slight variation from the other tests. Instead of Cesium JS using the smallest resources in each case, at Medium Zoom resource memory between Cesium JS and Esri ArcGIS Enterprise were almost identical. And at high zoom level, Esri ArcGIS Enterprise actually used less memory (about 10 MB less) (Figure 14; Figure 15). It is curious though, despite using less memory, in overall load time Cesium JS was still quicker.

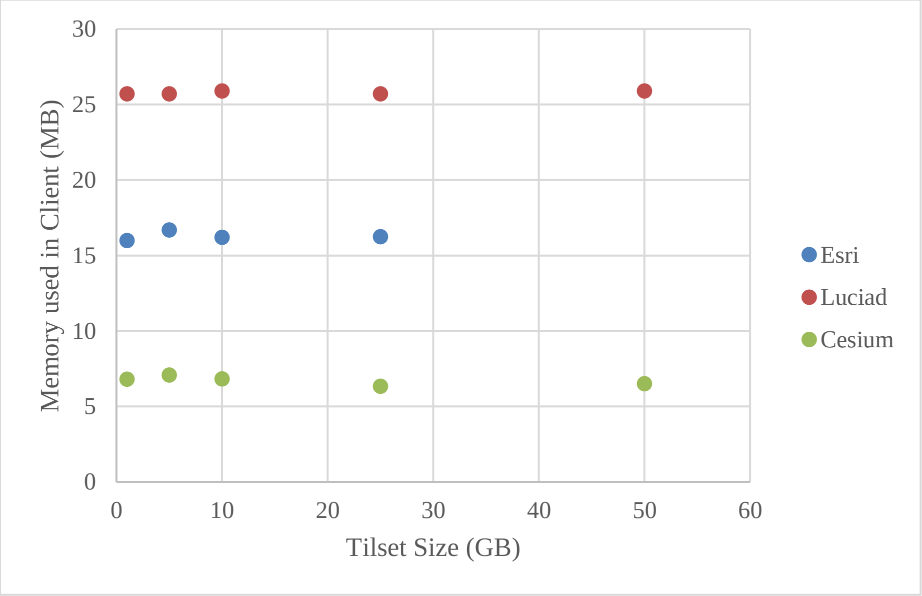


Figure 15: Resource Memory for Lidar Dataset at Full Extent

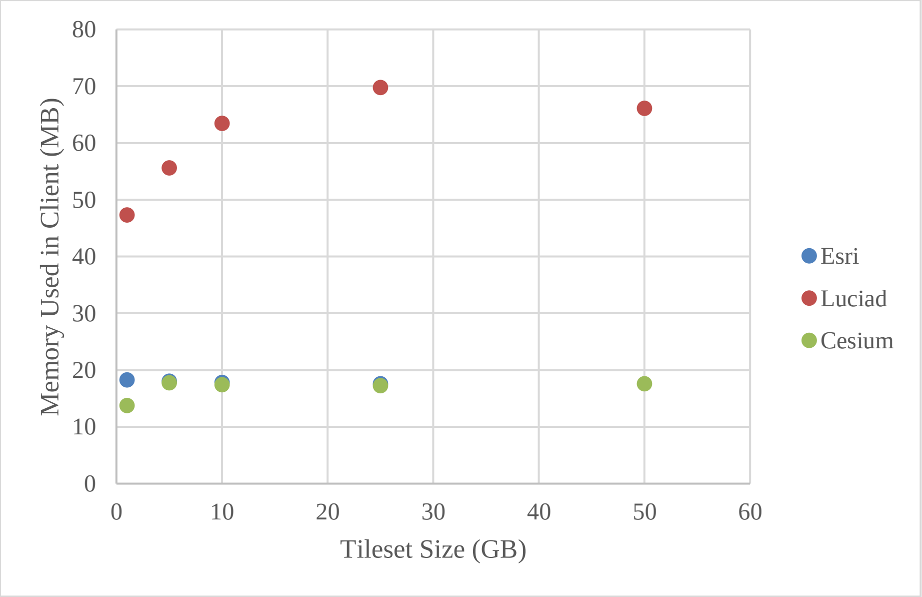


Figure 16: Resource Memory for Lidar Dataset at Medium Zoom

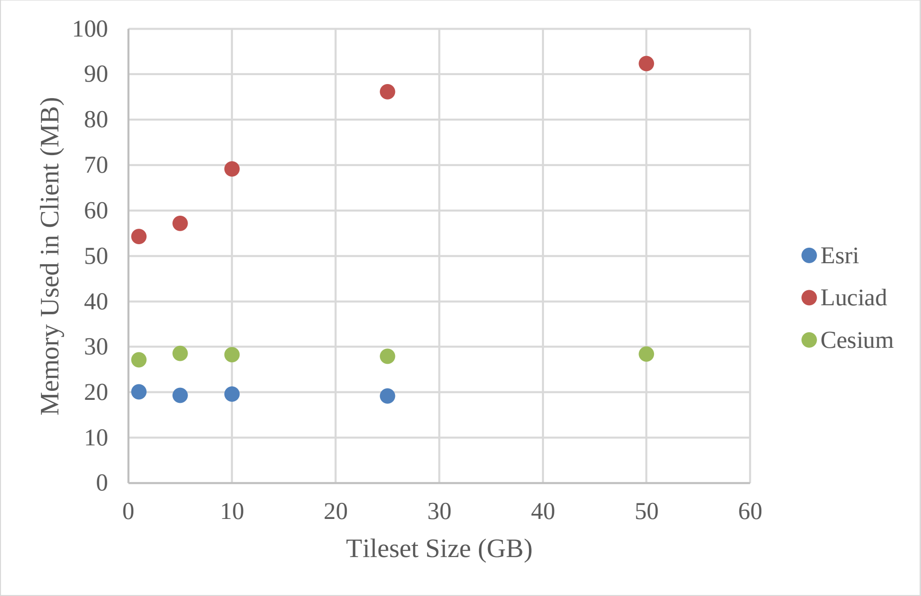


Figure 17: Resource Memory for Lidar Dataset at High Zoom

4.3. Qualitative Results

Although Cesium JS in most all cases had the quickest loading time and least memory used. It also produced the least detail in visualization compared to the outputs from Esri ArcGIS Enterprise and Hexagon Geospatial Luciad RIA. Figure 16 shows a screenshot of the Cesium JS

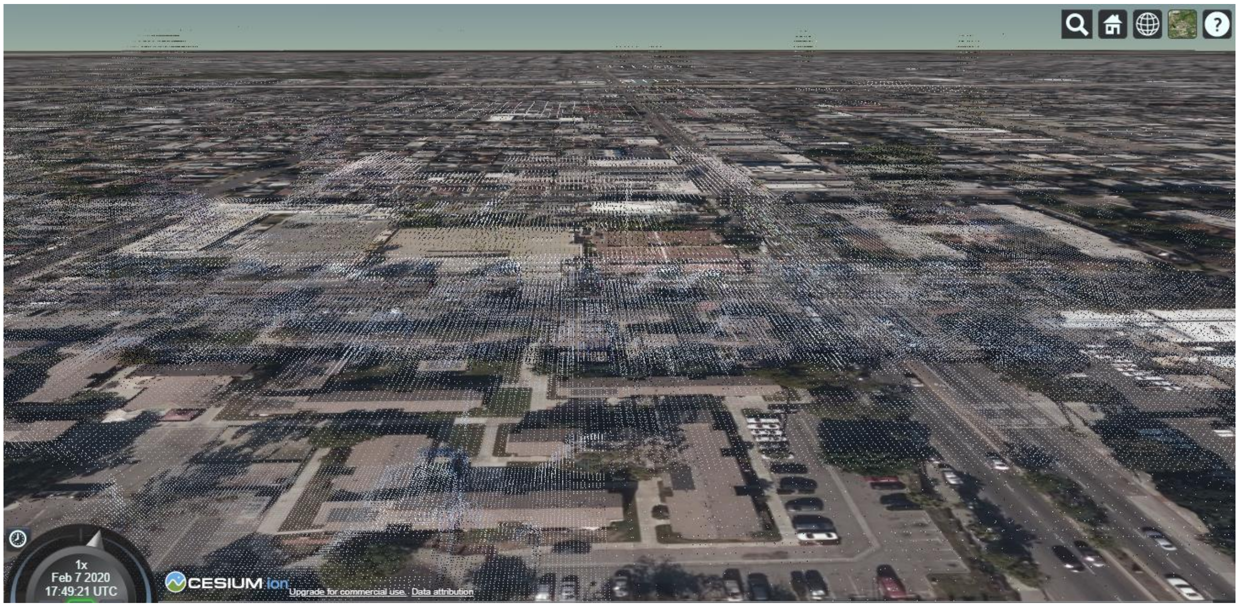


Figure 18: Cesium JS Zoomed in to Photogrammetric Dataset

system at a high zoom level. The points are so sparse that it is difficult to visually see much that is going on within the scene. In terms of loading time and memory Cesium JS appears more performant, but in its performance is a reduction in visualization within the client-side visualization application.

Hexagon Geospatial Luciad RIA performed much slower and used much more resources than the other platforms. However, its detail was much greater and more photorealistic than the other two exploitation systems. Figure 17 illustrates the photorealistic detail of the Luciad RIA platform. The larger use of resources is likely attributed to larger time spent generating color on top of the 3D tile points, and in loading the extensive libraries that Luciad RIA implements.

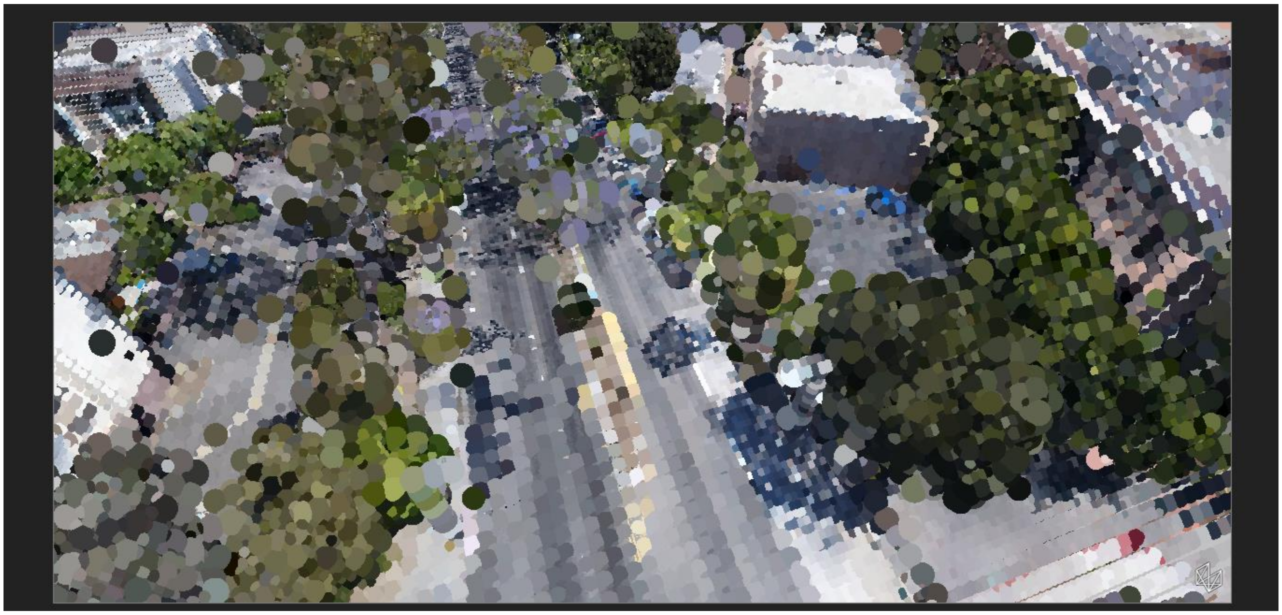


Figure 19: Photorealistic Output from Hexagon Geospatial Luciad RIA Application

Despite showing slightly quicker load times and slightly less memory in the loading test, Esri ArcGIS Enterprise implemented much more visual detail than Cesium JS. Figure 18 depicts

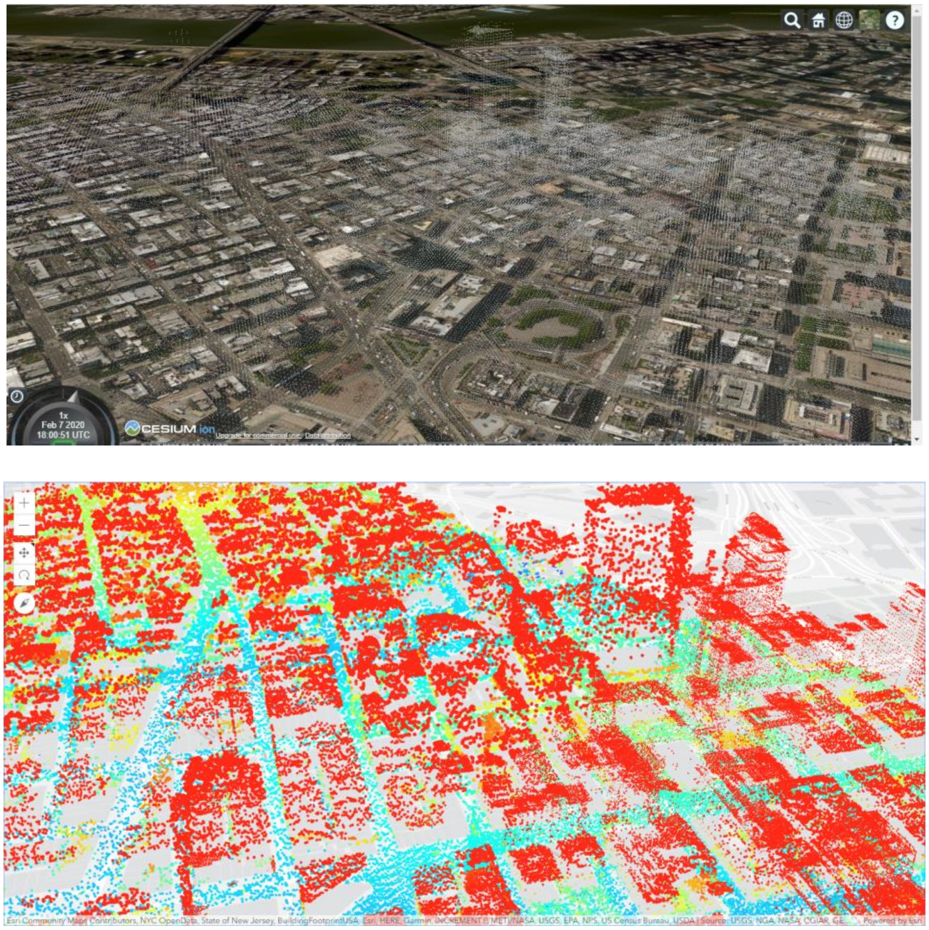


Figure 20: Detail difference in Cesium JS and Esri ArcGIS Enterprise. Cesium JS (top), ArcGIS Enterprise (bottom)

the visual differences between Cesium JS and Esri ArcGIS Enterprise in rendering the same scene. In a quantitative sense Cesium JS was slightly quicker than Esri ArcGIS Enterprise, but in a visual sense Esri ArcGIS Enterprise provided more detail.

Chapter 5 Discussions and Conclusions

The design of this study was constructed in order to understand the manners in which performance changes for adaptive web-based 3D geographic data structures at increasing tile set size, and zoom level. Adaptive web-based 3D geographic data structures implement a tile-based structure which changes resources loaded in the client application based on zoom level of the client device. Most past research with 3D data has been focused in the desktop environment with large point cloud files, with limited study on 3D web GIS capabilities and exploitation systems. 3D Web GIS capabilities are definitively growing, and this can be seen by the creation of adaptive web-based 3D geographic data structures. It was anticipated in the design of this study that overall tileset size would have a large impact on the performance of the exploitation systems; however, due to the tile structure of the datasets, there were very small changes in performance based on tileset size. It is likely performance does not vary greatly across tileset sizes since the browser is never fully loading the tileset and only specific tiles based on zoom level of the client. The biggest differences were observed instead with zoom level, and exploitation framework. The results from this study provide interesting insights into the behavior of adaptive web-based tile data structures and overall 3D web GIS capabilities. These web based systems and data types appear ideal for single scene visualization of very large 3D datasets, but much more extensive programming and customization is required to produce more analytic functionalities from these systems.

5.1. Performance Results

5.1.1. Qualitative and Quantitative Differences

The quantitative and qualitative results demonstrate general performance and visualization differences between each of the exploitation systems for loading various zoom levels of the tiled data. The performance results indicate that each system is likely suitable for different types of applications. Although the rendering time of Hexagon Geospatial Luciad RIA was slower compared to the other applications, it would be suitable for use in applications where single scene visualization is the most important element of the application. With its slow rendering time, it is likely unsuitable for applications which are more complex and analytic in nature. In these cases, Cesium JS and Esri ArcGIS Enterprise are more suitable choices for use. Both Esri ArcGIS Enterprise and Cesium JS had quick loading times. Cesium JS may be ostensibly the quickest framework from the loading tests; however, in cases where speed and detail of data matter, Esri ArcGIS Enterprise would be a more likely choice.

This study has investigated performance of loading various tileset sizes at various zoom levels within three different exploitation systems. The exploitation systems used in the study are web-based frameworks which are generally lightweight, and were configured with the most basic settings for this study. To optimize these systems, it would be necessary to implement browser side caching for each of the frameworks. As shown by GtMetrix, each of these platforms could benefit from browser side caching. Additionally, by using the base installment of these applications, single scene visualization was the core service investigated. With these exploitation systems setups, it would be interesting to build more complex analytic operations, or tools which integrate data from more than one scene. This study is limited in its performance analysis of only single scene load and visualization within each of the exploitation systems.

Each test was performed at increasing memory sizes, since it was thought that large tilesets would be much less performant in the systems than smaller tilesets. Within almost every test, there was very little change in performance across overall tileset size. These results indicate that the web and using level of detail data structures may be a quick way to perform analytic transactions. The i3s and 3D tile datasets have demonstrated to be very adaptive within each of the exploitation systems and further research would be useful to investigate the performance of more analytical transactions with these data structures.

5.1.2. Relations of Server Architecture and Performance Results

It appears that most performance differences between the three frameworks can be explained from the differences in library imports. In the GtMetrix results, Hexagon Geospatial Luciad RIA was ranked lowest in Google Page Score and Yahoo YSlow score due to its lack of compression and JavaScript miniaturization (Figure 5). In Cesium JS, the application is imported as a content delivery network which is a collection of two minified JavaScript files. These two compressed files load all the contents of the Cesium library into the application. Cesium JS also only creates 54 http requests to load in all components of the web page, while Hexagon Geospatial Luciad RIA create 699 http requests (Figure 3; Figure 5). Most of these additional http requests in Hexagon Geospatial Luciad RIA are calls to libraries within the Hexagon Geospatial Luciad RIA folder. The high number of library calls is responsible for the additional loading time and resources used by Hexagon Geospatial Luciad RIA. These libraries are also not compressed or minified like the Cesium JS files that are loaded in. Because Hexagon Geospatial Luciad RIA imports a vast number of libraries, it requires additional load time and resources compared to Cesium JS and Esri ArcGIS Enterprise.

Cesium JS and Esri ArcGIS Enterprise possess similar performance, with Cesium slightly outperforming Esri ArcGIS Enterprise. This performance difference can also be seen in the number of http requests and level of compression between the two servers. Cesium JS is 100% compressed, while Esri ArcGIS Enterprise scripts are only 75% compressed. Esri ArcGIS Enterprise created 107 http requests compared to only 54 by Cesium JS (Figure 3; Figure 4).

Cesium JS is the most compressed and lightweight library and therefore it is generally quickest of the three exploitation systems.

It should be noted that the number of http requests and therefore number of libraries imported appears to relate directly to the loading time and resources used in the page, but that capabilities within the page also shift greatly depending on library content available in the webpage. Hexagon Geospatial Luciad RIA may load slowest, and use the most library content, but it possesses the most photorealistic rendering of the data. The high-quality rendering of the data, can be traced to use of more extensive and sophisticated 3D color ramp libraries used by Hexagon Geospatial Luciad RIA to create a unique color map of the 3D Tile data. Similarly, Esri ArcGIS Enterprise performed slightly slower than Cesium JS, but its ability to color points was more advanced than Cesium JS. It is likely that additional library calls in Esri ArcGIS Enterprise provided more scripts for tile color rendering. Cesium JS may be the quickest, but it has the smallest number of library imports, and therefore the most minimal functionality available for coloration and display.

5.2. Summary

There are several useful ways in which researchers have been able to work with 3D point cloud data, and it is likely that development of more complex data structures and web GIS systems would assist in creation of more standardized tools and analytic capabilities for 3D point cloud data investigation. As documented, 3D geographic data has been useful by archaeologists to find new Maya sites, and by scientists to extract precise measurements of tree size and canopy (Canuto et al. 2018, Liew et al 2018). 3D data possesses more analytic value than 2D data since it can be used to provide volumetric properties that are not available with strictly 2D data. The issue with 3D data is its bulky data structure size. This makes it difficult to render and work with at large scales. To combat this, researchers have investigated data structure and computing resource changes. It has been demonstrated that adaptive level of detail data structures greatly increases performance of 3D data, and that through use of larger computing resources, either through the cloud or distributing computing clusters, that tools can be setup to analyze large point cloud scenes and also more than one point cloud scenes (Cura et al. 2017; Liu et al. 2016). Fairly recently new data structure standards have been adopted by the OGC for streaming 3D geographic data which are more adaptive, and web based. Although distributed computing provides the quickest platform, the integrated web and cloud environment provides the most accessible, scalable, and interoperable environment. Cloud resources can be purchased and easily scaled up to handle more and more data. Because of this, relatively new tools and systems have been constructed which are web-based and use adaptive tile-based data structures.

There has been limited study in the 3D web GIS environment. A group of researchers tested several different web data types and found 3D tiles the only capable candidate to render very large datasets. They note that visualization of large datasets was only possible with 3D tiles, but that it had limited analytic functionality (Murshed et al. 2018). With the adoption of 3D Tiles and i3s as OGC community standards for streaming massive datasets in the web it is important to understand their potential uses, and limitations. As it stands, there is limited study on the performance and capabilities of these data types, or even the platforms which can be used in the web to render them. This study was designed using an integrated web and cloud environment (due to its scalability for large datasets) and with focus on 3D web-based data types in hopes of identifying capabilities and understanding performance and limitations of these data structures. In order to test i3s and Cesium 3D Tiles in the web, Esri ArcGIS Enterprise, Hexagon Geospatial Luciad RIA, and Cesium JS were selected as server frameworks to render and serve 3D datasets.

Esri ArcGIS Enterprise was selected since it can host i3s data. The Esri ArcGIS Portal atmosphere can be used to publish 3D scene servers and to run 3D web applications which serve 3D data. Cesium JS was selected since it natively uses 3D Tiles and can also be used to create web applications which serve 3D Tile data. Hexagon Geospatial Luciad RIA was also selected since it can also be used to visualize 3D Tiles. These three systems were mainly chosen for their support of adaptive web-based data structures and their ability to author web applications which are capable of serving large 3D datasets in an integrated web and cloud environment.

It was the original intention to observe unique differences within the i3s and 3D Tile data structures, but each exploitation system only supported one type of 3D data, and therefore direct performance comparison between i3s and 3D Tiles was unable to be understood. The resulting tests provided useful information on the performances of the three exploitation systems tested, their limitations in 3D GIS capabilities, and some interesting insights in their potential applications with further development.

Although these results are useful, they still only provide a limited view of 3D web GIS performance. This study essentially quantified performance for single scene visualization of 3D geographic data with several different exploitation systems. Further research would continue to investigate performance using multiple scenes at once and for more analytical operations. It appears that in terms of rendering, using a browser and adaptive web-based data structures provide the foundation to visualize almost limitless amounts of tile data. The only ecosystem which could not render 50GB was Esri ArcGIS Enterprise Portal. Both Cesium JS and Hexagon Geospatial Luciad RIA were able to render scenes with 50GB tileset sizes, and with almost no performance changes compared to tilesets of smaller sizes. Although capabilities for analysis in the integrated web and cloud environment are minimal currently, this environment provides an ideal computing environment which is scalable and interoperable across devices. The next step in optimizing and enhancing 3D GIS capabilities would be further development of 3D tools using these data structures and exploitation systems. Each exploitation system tested is relatively lightweight and capable of visualization, but there are very little additional tools or capabilities available for 3D GIS in the web at this point.

5.3. Moving Past Single Scene Visualization

There is certainty many potential applications for extending the capacity of 3D geographic data analysis in the web, but the creation of a more standardized web toolset is necessary. As shown in the results of this study, performance for at least visualization of 3D point cloud data in the web is fairly standard across increasing tileset sizes since the web applications works with adaptive level of detail data structures. Even the spatial resolution of the data appears to have minimal impact on the performance of rendering tiles. Both the high-resolution photogrammetric data and lidar data possessed similar loading times. The integrated cloud and web environment with implementation of adaptive tile data structures appears to provide an environment could be scaled up for more complex computing operations.

These exploitation systems currently support mainly single scene visualization of 3D datasets, but it would be possible to develop on top of them to create more complex tools. As previously described in Chapter 2, researchers have already developed some novel cloud-based systems which permit visualization, editing, and classification of point cloud data (Cura et al. 2017). The issue is that these are carefully constructed systems with unique customizations that are built for a specific purpose. The research conducted by Cura et al. 2017 demonstrates that adaptive 3D data structures can be used to make quick and scalable analytic tools for point clouds, but these tools have not been developed at scale (Cura et al. 2017). The three exploitation systems investigated in this study all provide a way to visualize adaptive data and could maybe be used as entry points to develop more complex tools and capabilities.

The most useful applications for 3D geographic data study appear to include change detection and change monitoring applications. Satellite imagery is used frequently in change detection study, and with integration of 3D change detection, many industries including natural resources, defense, and local government could benefit. 3D data provides the ability to quantify volumetric change which cannot be done with 2D imagery. With the correct computing environment, change detection algorithms could be setup and ran with 3D data. It would be interesting to develop these algorithms within the integrated web and cloud environment and to compare their performance within the exploitation systems. The integrated web and cloud environment appears to be a good place that change detection for 3D data could be setup, the tools just do not exist to do it now. Visualization appears to be the most useful 3D GIS capability in the web. Although 3D visualization is useful, 3D web GIS would greatly increase in capabilities with development of more analytic tools, like 3D change detection.

Visualization is an important component to geographic data investigation, but often it is the production of analytical products which are the most useful capabilities to GIS systems. Desktop GIS systems possess many analytical tools which help researchers analyze and process various forms of vector and raster 2D data. With the exploitation frameworks provided by Cesium JS, Esri ArcGIS Enterprise, and Hexagon Geospatial Luciad RIA, 3D visualization of massive datasets is the primary capability, but more complex applications could be setup to perform change detection operations or other analytic transactions. Testing the performance of change detection for large point cloud datasets in the integrated cloud and web environment would be a very useful piece of research.

The next step in further evaluating 3D web GIS capabilities would be through the comparison of large data processing and analytic operations in the browser implementing both tiled data and point cloud data. Tiling data appears to assist the browser in being able to visualize large datasets since the dataset is tiled and only specific tiles are rendered based on zoom level. It would be interesting to test 3D data processing in the integrated cloud and web environment in addition to just basic visualization. There are desktop programming languages like C++ that will always function more efficiently than interpreted languages like JavaScript that are used in the web environment. However, in the integrated web and cloud environment programs may have access to more hardware resources and parallelization that may make processing and analysis in the web more efficient than in the desktop environment. It would therefore be interesting to note if analysis in the cloud is quicker with adaptive datasets or the source dataset since the transaction is performed on every item within the dataset. Further study of 3D GIS systems should integrate processing and analysis of 3D data with variant data structures using an integrated web and cloud environment.

5.4. Limitations and Areas of Future Research

A large difficulty in this study was setting up a computing environment that could be used to empirically test all datasets against each other. In order to standardize the computing environment all software and data was setup on an Amazon EC2 instance. One of research areas that was attempted to be covered which was missed was the direct comparison in performance between Esri i3s and Cesium 3D Tiles. None of the exploitation systems took in both Esri i3s and Cesium Tiles and therefore the two data structures could not be empirically analyzed against each other. The server frameworks Esri ArcGIS Enterprise Portal and Cesium JS in their rendering of a 3D dataset can be compared, but the objective difference due to data structure cannot be extracted. It appears from observing the two server frameworks that the data structures act very similar. The overall performance between Esri ArcGIS Enterprise and Cesium JS was generally similar across all tests. Hopefully one of the platforms will eventually support multiple data types, but as it stands now each platform implements a unique data 3D structure. In the future, if any of the systems were made capable to ingest both data types, these data types could be empirically compared. Therefore, the results from this study largely indicate differences in how each exploitation system handles that data structure, but not the data structure itself.

Additionally, each server framework was setup in its most basic distribution. In further iterations of this research, it would be necessary to learn more about each framework and to implement more advanced coloring and caching options. In these tests the default behavior was implemented. Each server can be reprogrammed to display different color panels, and it is likely that browser and server-side caching could be enabled. These configurations would also likely impact performance and in future study would be areas that should be tested.

In order to truly understand performance, a more complex zoom test would also be performed. In this study, three different zoom levels were selected, but it would be interesting to test a wider array of zoom levels at each ‘pixel’ within the dataset. Performing a more robust test would illustrate at which zoom levels and spatial resolutions the performance changes. Performing the loading tests at multiple zoom levels provided valuable feedback in how to the browser handles the tiled datasets at different zoom levels, but these tests could be made more complex and improved to understand even more about the server.

5.5. Insights

The results from this study ultimately reveal the capabilities available in current 3D web GIS technology. From the quantitative and qualitative results, it appears each platform is suitable for a different range of applications. For example, Hexagon Geospatial Luciad RIA would probably be best in an application where researchers show a 3D model, like an archaeological site, since it is most photorealistic out of the exploitation systems. Cesium JS or Esri ArcGIS Enterprise Portal would likely be suited for more complex analytical operations like change detection analysis where high fidelity visualization is not primary concern. The performance of these platforms are very similar and they are likely both suitable for complex operations.

These exploitation systems all show good promise of being able to effectively handle large datasets. They are currently limited to mostly basic single scene visualization but developing them with an integrated web and cloud environment may provide the ability to develop more complex and analytic tools for 3D GIS data. The cloud provides an accessible environment that is interoperable across devices and using tile-based data structures has shown promise in large scale visualization of 3D data. The results from this research help demonstrate the current capabilities and limitations of 3D web GIS, and hopefully further research will continue to enhance the capabilities of 3D web GIS. This research ultimately provides a snapshot in time of current technology, and as technology continues to develop rapidly the capabilities and performance of these systems will undoubtfully change.

Much more complex analysis and tool development is required in order to further analyze these systems. This research provided a joint qualitative and quantitative investigations into 3D web GIS and it provides a valuable baseline of data in which more in depth analytical testing and tool development could be based off in the future.

References

Agrawal, Sonam, and Rajan D. Gupta. 2014. "Development and Comparison of Open Source Based Web GIS Frameworks on WAMP and Apache Tomcat Web Servers." *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XL-4: 1-5.

Anderson, S.P., Qinghua, G., and Parrish, E.G., 2012, "Snow-on and snow-off Lidar point cloud data and digital elevation models for study of topography, snow, ecosystems and environmental change at Boulder Creek Critical Zone Observatory," Colorado: Boulder Creek CZO, INSTAAR, University of Colorado at Boulder, digital media, distributed by OpenTopography. https://doi.org/10.5069/G93R0QR0

Auer, Michael, and Alexander Zipf. 2018. "3D WebGIS: From Visualization to Analysis. An Efficient Browser-Based 3D Line-of-Sight Analysis." *ISPRS International Journal of Geo-Information* 7(7): 279.

Bevis, M. and Hudnut, K. 2005. "B4 Lidar Project: Airborne Laser Swath Mapping (ALSM) survey of the San Andreas Fault (SAF) system of central and southern California, including the Banning segment of the SAF and the San Jacinto fault system." National Center for Airborne Laser Mapping (NCALM), U.S. Geological Survey, the Ohio State University, and the Southern California Integrated GPS Project, distributed by OpenTopography. https://doi.org/10.5066/F7TQ5ZQ6.

Boerner, Richard, Ludwig Hoegner, and Uwe Stilla. 2017. "Voxel Based Segmentation of Large Airborne Topobathymetric LiDAR Data." *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 42(1): 107-114.

Bröring, Arne, David Vial, and Thorsten Reitz. 2014. "Processing Real-Time Sensor Data Streams for 3D Web Visualization." In *Proceedings of the 5th ACM SIGSPATIAL International Workshop on Geostreaming*, 72–80. ACM.

Canuto, Marcello A., Francisco Estrada-Belli, Thomas G. Garrison, Stephen D. Houston, Mary Jane Acuña, Milan Kováč, Damien Marken, et al. 2018. "Ancient Lowland Maya Complexity as Revealed by Airborne Laser Scanning of Northern Guatemala." *Science* 361(6409): 1355–+.

Cesium. 2019. "CesiumJS." Last modified November 6, 2019. https://cesium.com/cesiumjs/

Cesium. 2015. "Introducing 3D Tiles." August 10, 2015. https://cesium.com/blog/2015/08/10/introducing-3d-tiles/.

Cesium. n.d. "Getting Started." Accessed February 28, 2020. https://cesium.com/docs/tutorials/getting-started/.

Chen, Yiqun, Erfan Shooraj, Abbas Rajabifard, and Soheil Sabri. 2018. "From IFC to 3D Tiles:

An Integrated Open-Source Solution for Visualising BIMs on Cesium." *ISPRS International Journal of Geo-Information* 7(10).

Coghe, Nele. 2018. "Science Meets Art in Luciad V2018." *Hexagon Geospatial Blog*. https://blog.hexagongeospatial.com/science-meets-art-in-luciad-v2018/.

Collins, Victoria. "The Decline of the Native App and the Rise of the Web App." *Forbes*, April 5, 2019. Accessed February 28, 2020. https://www.forbes.com/sites/victoriacollins/2019/04/05/why-you-dont-need-to-make-an-app-a-guide-for-startups-who-want-to-make-an-app/#1a3f9abb6e63

Colomina, Ismael, and Pere Molina. 2014. "Unmanned aerial systems for photogrammetry and remote sensing: A review." *ISPRS Journal of Photogrammetry and Remote Sensing* 92:79–97.

Cura, Rémi, Julien Perret, and Nicolas Paparoditis. 2017. "A Scalable and Multi-Purpose Point Cloud Server (PCS) for Easier and Faster Point Cloud Data Management and Processing." *ISPRS Journal of Photogrammetry and Remote Sensing* 127: 39-56.

Daraeihajitooei, Mohammadhossein. 2018. "Tightly-Coupled Lidar and Camera for Autonomous Vehicles." PhD diss., University of California Santa Cruz.

Esri. n.d. "ArcGIS API for JavaScript 3.29." Accessed November 16, 2019. https://developers.arcgis.com/javascript/3/jsapi/.

Esri, n.d. "Lidar point classification." Accessed November 17, 2019. https://desktop.arcgis.com/en/arcmap/10.3/manage-data/las-dataset/lidar-point-classification.htm.

Esri. n.d. "Types of Lidar." Accessed November 16, 2019. https://desktop.arcgis.com/en/arcmap/latest/manage-data/las-dataset/types-of-lidar.htm.

Esri. n.d. "Cloud Formation Templates to Deploy ArcGIS Enterprise on Amazon Web Services." Accessed March 29, 2020. http://arcgisstore105.s3.amazonaws.com/6491/docs/index.html

Farkas, Gabor. 2017. "Applicability of open-source web mapping libraries for building massive Web GIS clients." *Journal of Geographic Systems,* Springer 19(3):273-295.

Forte, Maurizio. 2014. "3D Archaeology New Perspectives and Challenges-The Example of Çatalhöyük." *Journal of Eastern Mediterranean Archaeology & Heritage Studies* 2(1): 1.

Foy, Kylie. 2018. "Lidar scans over the Carolinas accelerate hurricane recovery." *Massachusetts Institute of Technology Lincoln Laboratory*, December 13, 2018. https://www.ll.mit.edu/news/lidar-scans-over-carolinas-accelerate-hurricane-recovery.

Gaillard, Jérémy, Alexandre Vienne, Rémi Baume, Frédéric Pedrinis, Adrien Peytavie, and Gilles Gesquière. 2015. “Urban Data Visualization in a Web Browser.” In *Proceedings of the 20th International Conference on 3d Web Technology* 81–88. ACM.

Gaillard, J., A. Peytavie, and G. Gesquière. 2016. “Data Structure for Progressive Visualisation and Edition of Vectorial Geospatial Data.” *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* IV-2/W1(2): 201–209.

Geological Survey of Israel. 2006. “Israel's Mediterranean Sharon sea cliff,” distributed by OpenTopography. https://doi.org/10.5069/G99C6VBM

GtMetrix. “GtMetrix.” Accessed February 6, 2020. https://www.opengeospatial.org/pressroom/pressreleases/2946.

Guan, Haiyan, Jonathan Li, Liang Zhong, Yu Yongtao, and Michael Chapman. 2013. “Process Virtualization of Large-Scale Lidar Data in a Cloud Computing Environment.” *Computers and Geosciences* 60: 109-116.

Hexagon Geospatial. n.d. “Luciad Portfolio.” Accessed November 16, 2019. https://www.hexagongeospatial.com/products/luciad-portfolio.

Huang, Qunying, Chaowei Yang, Kai Liu, Jizhe Xia, Chen Xu, Jing Li, Zhipeng Gui, Min Sun, and Zhenglong Li. 2013. “Evaluating Open-Source Cloud Computing Solutions for Geosciences.” *Computers and Geosciences* 59.

Jackson, Keith, Lavanya Ramakrishan, Krishna Muriki, Shane Canon, Shreyas Cholia, John Shalf, Harvey J. Wasserman, and Nicholas J. Wright. 2010. “Performance Analysis of High Performance Computing Applications on the Amazon Web Services Cloud.” In *2010 IEEE Second International Conference on Cloud Computing Technology and Science* 159-168. IEEE.

Jayaraj, Purnima, and Anandakumar M. Ramiya. 2018. “3D CityGML Building Modeling From Lidar Point Cloud Data.” *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLII-5(5). 175-180.

Jiang, Wei, Vignesh T. Ravi, and Gagan Agrawal. 2009. “Comparing Map-Reduce and FREERIDE for Data-Intensive Applications.” In *2009 IEEE International Conference on Cluster Computing and Workshops*, 1-10. IEEE.

Józsa, Oszkár, Attila Börcs, and Csaba Benedek. 2013. “Towards 4D Virtual City Reconstruction from Lidar Point Cloud Sequences.” *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* II-3/W1(3): 15-20.

Krämer, Michel, and Ralf Gutbell. 2015. “A Case Study on 3D Geospatial Applications in the Web Using State-of-the-Art WebGL Frameworks.” In *Proceedings of the 20th International Conference on 3d Web Technology*, 189–197. ACM.

Kreylos, Oliver, Gerald W. Bawden, and Louise H. Kellogg. 2008. “Immersive Visualization and Analysis of LiDAR Data.” *Advances in Visual Computing Lecture Notes in Computer Science* 846-855.

Leonard, Paul B., Robert F. Baldwin, Jessica A. Homyack, and Thomas Bently Ben Wigley. 2012. “Remote Detection of Small Wetlands in the Atlantic Coastal Plain of North America: Local Relief Models, Ground Validation, and High-Throughput Computing.” *Forest Ecology and Management* 284: 107–115.

Leu, Jenq-Shiou, Yun-Sun Yee, and Wa-Lin Chen. 2013. “Comparison of Map-Reduce and SQL on Large-Scale Data Processing.” *Journal of the Chinese Institute of Engineers* 36(1): 27–34.

Li, Wenwen, Sheng Wu, Miaomiao Song, and Xiran Zhou. 2016. “A Scalable Cyberinfrastructure Solution to Support Big Data Management and Multivariate Visualization of Time-Series Sensor Observation Data.” *Earth Science Informatics* 9(4): 449–464.

Liew, S.C., X. Huang, E. S. Lin, C. Shi, A. T. K. Yee, and A. Tandon. 2018. “Integration Of Tree Database Derived From Satellite Imagery And Lidar Point Cloud Data.” *The International Archives of the Photogrammetry* XLII-4-W10:105-111.

Liu, Kun, Jan Boehm, and Christian Alis. 2016. “Change Detection Of Mobile Lidar Data Using Cloud Computing.” *The International Archives of the Photogrammetry* XLI-B3: 309–313.

Marx, Andrew, Kevin Mercy, Yu-His Chou, and Richard Windisch. 2019. “A Lightweight, Robust Exploitation System for Temporal Stacks of UAS Data: Use Case for Forward-Deployed Military or Emergency Responders.” *Drones* 3(1): 29.

Meng, Xuelian, Nate Currit, and Kaiguang Zhao. 2010. “Ground Filtering Algorithms for Airborne LiDAR Data: A Review of Critical Issues.” *Remote Sensing* 2(3): 833–860.

Murshed, Syed, Ayah Al-Hyari, Jochen Wendel, and Louise Ansart. 2018. “Design and Implementation of a 4D Web Application for Analytical Visualization of Smart City Applications.” *ISPRS International Journal of Geo-Information* 7(7).

National Ecological Observatory Network. 2016. “Data Products NEON_D17.” Provisional data downloaded from http://data.neonscience.org on 20 Oct 2018. Battelle, Boulder, CO, USA, distributed by OpenTopography. https://doi.org/10.5069/G9V98609

National Research Council. 2014. *Laser Radar: Progress and Opportunities in Active Electro-Optical Sensing*. Washington, DC: The National Academies Press.

Nebiker, Stephan, Susanne Bleisch, and Martin Christen. 2010. “Rich Point Clouds in Virtual

Globes – A New Paradigm in City Modeling?." *Computers, Environment and Urban Systems* 34(6): 508-17.

OCM Partners. 2015. "2013-2014 U.S. Geological Survey CMGP LiDAR: Post Sandy (New York City)." https://inport.nmfs.noaa.gov/inport/item/49891.

Open Geospatial Consortium. 2019. "OGC Adopts 3D Tiles as Community Standard." February 5, 2019. https://www.opengeospatial.org/pressroom/pressreleases/2946.

Open Geospatial Consortium. n.d. "Indexed Scene Layers (i3s)." Accessed November 17, 2019. https://www.opengeospatial.org/standards/i3s.

Richter, Rico, and Jürgen Döllner. 2014. "Concepts and Techniques for Integration, Analysis and Visualization of Massive 3D Point Clouds." *Computers, Environment and Urban Systems* 45: 114-24.

Richter, Rico, and Döllner, Jürgen. 2010. "Out-of-Core Real-Time Visualization of Massive 3D Point Clouds." In *Proceedings of the 7th International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa*, 121–128. ACM.

Stavros, E.N., T. Zachary, V. Kane, S. Veraverbeke, R. McGaughey, J. Lutz, C. Ramirez, & D.S. Schimel. 2016. "Unprecedented remote sensing data over the King and Rim Megafires in the Sierra Nevada Mountains of California. National Park Service, Fire and Aviation Management Branch, Fuels and Ecology Program," distributed by OpenTopography. https://doi.org/10.5069/G9V69GJ4

Stoter, Jantien, Bruno Vallet, Thomas Lithen, Maria Pla, Piotr Wozniak, Tobias Kellenberger, Andre Streilein, Risto Ilves, and Hugo Ledoux. 2016. "State-Of-The-Art Of 3d National Mapping In 2016." *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* XLI-B4: 653–660.

Wang, Chao, and Yong K. Cho. 2015. "Smart Scanning and near Real-time 3D Surface Modeling of Dynamic Construction Equipment from a Point Cloud." *Automation in Construction* 49: 239-49.

Whitehead, K., and C. H. Hugenholtz. 2014. "Remote Sensing of the Environment with Small Unmanned Aircraft Systems (Uass), Part 1: A Review of Progress and Challenges 1." *Journal of Unmanned Vehicle Systems* 2 (3): 69–85. doi:10.1139/juvs-2014-0006.

Yue, Peng, Hongxiu Zhou, Jianya Gong, and Lei Hu. 2013. "Geoprocessing in Cloud Computing Platforms - a Comparative Analysis." *International Journal of Digital Earth* 6(4): 404–425.

Zhang, Jianqin, Jianhua Gong, Hui Lin, Gang Wang, Jianling Huang, Jun Zhu, Bingli Xu, and

Jack Teng. 2007. “Design and Development of Distributed Virtual Geographic Environment System Based on Web Services.” *Information Sciences* 177(19): 3968–3980.