|  |  |
| --- | --- |
|  | Visualizing Point Clouds in Virtual Reality |
|  |  |
|  | Tyler B Gates  Senior Design Project  2019-2020 |

Table of Contents

* [Abstract 2](#_Toc40391962)
* [Introduction 3](#_Toc40391963)
* [Background 4](#_Toc40391964)
* [Methods 6](#_Toc40391965)
* [Results 11](#_Toc40391966)
* [Discussion 13](#_Toc40391967)
* [References 14](#_Toc40391968)

# Abstract

This goal of this project is to analyze different methods of virtualizing and visualizing historic monuments and locations for the purpose of site preservation and exploration. We were also looking to see if this task was possible, if the process was automatable and informational enough to teach in the classroom. Multiple programs, renderers and virtual reality methods were investigated and chosen based on performance, ease of use and total processing time; these include RealityCapture, CloudCompare, Potree and Cyclone. Our point clouds were collected from previous projects hosted by the Geographical and Sustainability Sciences College due to the COVID-19 epidemics, along with photogrammetric models taken during. These point clouds were then processed through different software in order to upload them into our chosen visualizing engine, Unity 3D. Others were further processed, and converted into mesh objects in order to give an accurate, solid object that one can interact with. The most effective method of visualization found seemed to be produced with images and photogrammetry, being supplemented with lidar scans to add detail. These data were then put through Cyclone to align and complete the point clouds, brought into CloudCompare to clean the clouds of artifacts, and then into RealityCapture to create accurately colored and proportioned meshes. These were uploaded into a WebGL-formatted application in order for interested parties to walk around and interact with the objects.

# Introduction

This study’s focus was on creating replicable workflows for visualizing point clouds in virtual reality settings, both fully-immersed and in-browser. There are quite a few methods of achieving these goals which have been implemented in the past, but not many that can be recreated easily nor viable for a classroom-based setting. These applications are receiving increased interest from academic institutions and geographic corporations alike due to their effectiveness in explaining data (*Schwarz et. al, 2019*), so being able to teach these ideas in an academic setting will become more important in the near future. Another goal of this study is to find programs which will create and display high-quality data visualizations with a focus on efficiency. Not everyone will have access to high-powered graphics machines which can handle the high intensity of large data structures, so providing simplified workflows capable of delivering quality results is necessary. The last goal of this study was to find out if the entire process, from data collection to visualization, could be automated in some way.

The methods that will be discussed in this study include preprocessing of point cloud and photogrammetric data, model and mesh creation from images, visualizing the resulting models in a rendering engine, and implementing these visuals into virtual reality settings. Multiple software versions and methods are discussed for each step in the workflow processes, and effectiveness of each one is determined for a specific objective. A final workflow for two different methods of visualization will be determined through these comparisons, and finalized based on quality of model construction and computational effectiveness.

This study will be useful for anyone trying to find an efficient way of displaying their point cloud or photogrammetric data, who may not have access to high-powered machines that are normally used for these tasks. It will also be useful for professors and students alike who depend on visuals to communicate their data effectively. Museums and other historical preservation organizations could also use these workflows to assist in the preservation of artifacts or areas of interest that may not be around forever (*Haddad, 2011*).

# Background

Some of the earliest studies done in the field of virtual reality via laser scanning was published in the Open Digital Cultural Heritage Systems Conference in 2008. A study by S. Sotoodeh et. al focused on early-stage algorithms for triangulating laser scan points to create mesh objects, including applications with underwater objects and site reconstruction. Another early adaptation of virtual / alternate reality applications was produced in the vhdPlus framework, which was an early virtual simulation program often used in museums and online platforms (*Arnold, 2008*). While this study did not involve the use of laser scanning, it was one of the first mass-scale applications of virtual reality to provide a means to explore a large geographic location. This journal in its entirety provided the basis for much of the virtual reality research focused on historical preservation to this day.

Some of the first 3D models created through a combination of laser scanning and photogrammetry were studied in early 2008 as well. Again with a focus on architecture and archaeology, this methodological approach involved using a laser scanning model, photogrammetric model, a theoretical 3D model created in CAD software, and a final model derived from a merging of all three previous methods (*Drap et. al, 2008*). Later that year, these algorithms and methods were improved upon by Changjae Kim of the Schulich School of Engineering, who applied these methods to constructing 3D visualizations of urban environments.

Similar studies to the one being presented in this paper have been explored in the recent past. A study done by researchers concerned with the preservation of the UNESCO Melaka World Heritage Site proposed that a mixed AR/VR application may be an effective way to preserve the site, while enhancing visitor experiences (*Kamarulzaman, 2014*). This particular application of a VR environment is to help preserve this fragile site, and add visual information to those who visit the region to seek knowledge on the subject.

A methodological study produced by HafenCity University in Hamburg discussed the concept of virtual museums for the purpose of historical preservation and timeline presentation (*Kersten et. al,* *2017*). This team used a mixture of terrestrial laser scanners and cameras to recreate the building of interest in virtual reality, which was processed through Unreal engine on an HTC Vive. The building was also reconstructed in a CAD software to show the progression in construction over the past 450 years, adding more historical value to the presentation. This workflow is the most similar to this study that we have found in recent publications, which also had comparable results.

The history of these studies stems only few years into the past, as it is a relatively new school of thought in visualization. This study intends to further these findings from the past, and aid those who are also looking to progress the field of geographic and spatial visualization. We will be using similar methods to the study carried out by the team from HafenCity University, with the interest of replicability and efficiency in mind.

# Methods

Data Collection

The laser scanners used in this study include the Leica P30, and the Leica BLK model, which are both terrestrial lidar scanners. The Leica P30 is a larger scanner which mainly used in surveying, construction and is able to retrieve data at a scan rate of 1,000,000pps, and has a range of 1.2mm at up to 270m away. It uses a 1550nm and 658nm laser for this collection, and has a full-RGB camera in order to supply accurate color values to the point cloud (*Leica, 2017*). This scanner was used to capture the Herbert Hoover House, a national monument located in West Branch, Iowa. There were scans taken in total of both the inside and outside of the building, which includes a bedroom, living room and kitchen. This scanner was also used to scan the Beer Caves of Iowa City, constructed in the 1850s and located underground in the northern side of the city.

In contrast to the P30, the Leica BLK is an extremely lightweight, portable scanner. This machine is able to collect data at 360,000pps, with a resolution of 7mm at a maximum range of 60m. It uses a 830nm laser to collect data (*Lecia, 2017*). This was used to scan the Mammal Hall Museum, located in the University of Iowa’s MacBride Hall. In the case of both scanners, certain marks need to be met in order to register, or align, the point clouds that are collected. According to the California DOT (*2018*), there should be a minimum of 5-20% overlap between each scan taken. This was taken into account before taking every can, ensuring that complete coverage of each subject was achieved.

The camera model used for the photogrammetric processes was the Canon EOS Rebel XTi, produced in 2006. This model has a 10.1mpx, single plate, high-sensitivity sensor. The images were captured in large-format .JPG (3888x2592) files, using the Auto-adjustment capture setting (*Canon, 2006*). The building of which the pictures were taken is the Chicago, Rock Island and Pacific Passenger Station in Iowa City, Iowa. This is an historic train station built in 1898, and has been repurposed into rentable office space. Around 230 images were taken of the station in various positions and angles around the building in order to produce a complete and accurate model. In order for the photogrammetric process to work smoothly, there needs to be around 80% overlap in the x-plane, and around 65% in the y-plane (*King, 2017*). Consistent lighting is a preferred attribute (Keyser, et. al, 2018), so photos of the building needed to be taken at the same time of day. If lighting is inconsistent, and has too many shadows, the registration of images may not be as accurate as possible.

The roof of a house

Description automatically generatedA large brick building with grass in front of a house

Description automatically generated

Figure 1: Two images which were used in the modeling of the historic Passenger Station.

The machine used throughout this study was a mid-level gaming computer. The Windows 10 operating system was used as it is one of the most commonly-found operating systems. An eight-core AMD FX-8320 was installed, along with an NVIDIA GeForce GTX 1060 6GB graphics driver. 16GB of DIMM memory is also being used – the preferred minimum amount of memory to be used in virtual reality development. This is all driven by a Gigabyte GA-78LMT-USB3 motherboard, allowing for the quick data transfer timing needed for virtual reality applications. The Oculus Rift headset is the hardware used for testing the applications that are created in this study.

Point Cloud Processing

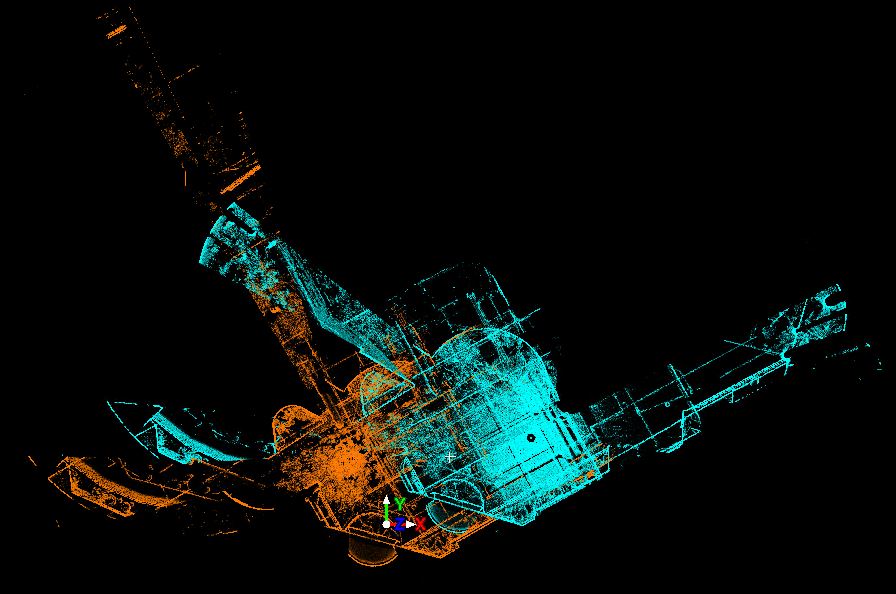
Cyclone, a point-cloud registration software created by Leica Geosystems, is an extremely powerful program allowing for the manipulation and editing of laser scans (*Leica Geosystems*). This is one of the most popular subscription-based software on the market. It provides mass amounts of processing power along with plenty of features which can be applied to nearly any type of point-could data. We used this program to align the point clouds once they had been collected. All scans were uploaded into databases based on the subject of the scans. Visual registration (*Figure 2)* was used for all workspaces, where a maximum RMSE of 0.08 achieved. This process can also be done using target registration or a new feature called REGITER 360, but the visual alignment method works very well and takes far less time overall to accomplish. After all the scans were aligned with one another, the complete cloud was ready to be edited. Points determined to be artifacts were removed in the appropriate ModelSpaces within the Cyclone databases, and were re-evaluated afterwards. These were saved in both .e57 and .ptx formats; the two specific file types are widely accepted by point cloud and mesh editing software (*Gray, 2020*).

Figure 2: Visual registration in Cyclone

CloudCompare is an open-source 3D point cloud processing program (*Mastrogiovanni*). Its main purpose was to be able to compare different point clouds together, and create meshes from these scans. It uses an octree structure which was designed for this particular purpose, and can efficiently render clouds with up to 120 million points. Alignment of point clouds in this application is possible, but proved to be extremely time-consuming, computationally exhausting, and fairly inaccurate compared to the results from Cyclone. However, after alignment in a separate software, editing of the clouds was fairly effective. Visualizations were in true RGB color, which aided in the identification of points and regions which did not belong in the scan or could otherwise be considered artifacts. Due to these findings, Cyclone was decidedly the best method of scan alignment. However, this software had proven to be the more effective method of editing the aligned clouds, and were combined as such in the final workflow.

Photogrammetric Processing

A screenshot of a computer

Description automatically generatedA picture containing table

Description automatically generatedRealityCapture, produced by Capturing Reality, is a photogrammetry software which creates high-detail, realistic 3D models. This application uses overlapping images to create measurements between objects, creating a geometric model of the object (*Wheeler, 2016*). Using efficient and powerful photogrammetric algorithms, this program was able to create a mesh out of the images taken of the train station. The process of this program included first aligning the images, which created an initial dense point cloud. This dense cloud is then triangulated, forming a detailed mesh of the object. RC created a rough mesh afterwards, adding color and texture to the model based off of photo calculations and measurements. If lidar scans were to be taken of the object, they could be used to augment the photos in order to create a more complete and detailed reconstruction. After the process had finished, the object was exported as an .obj file in normal detail settings, as to lower the computational cost of rendering it later on.

Figure 3: A dense cloud (top) and low-quality mesh (bottom) created in RealityCapture

MeshLab was used post-photogrammetric processing in order to clean and edit the objects created by RealityCapture. Basic smoothing was done on most parts which were sticking out, and some faces were added to parts of multiple mesh objects due to the presence of missing data which the photographs did not encapsulate. Due to the sheer size and detail of some objects, MeshLab tended to crash somewhat often. However, not many major changes were needed and could be completed in time before the program ceased to work.

Visualization

Potree is an open-source WebGL application which allows for the rendering of large point clouds. This program was created by Markus Schuetz at the Institute of Computer Graphics and Algorithms in TU Wein. This renderer uses an algorithm known as the modifiable nested octree, which allows for the subsampling of point clouds at different levels of detail (*Scheutz, 2016*). This application was very useful for visualizing point clouds in a browser setting, as it can render them on basically any system with a small amount of memory used. However useful, this was not the goal of the project. The only part of this software which was used is the point cloud file converter, which converted nearly any type of point cloud in t a Potree file type. This is the filetype which was used in the Unity plugin, and is a necessary step in the point cloud visualization workflow overall.

The last component to both photogrammetric and point cloud visualization workflows is the Unity 3D engine. This is an all-encompassing graphics engine, allowing for the creation of 2D and 3D video games, simulations, and virtual / alternate reality applications (*Unity, 2020*). Both WebGL and 3D projects were created for this study, both of which use different scripts to run their builds. To visualize point cloud data, The Potree-converted clouds were uploaded into the asset folder. Multiple point cloud plugins were tested on these files – one made by Simon Fraiss, and another by a GitHub user going by the name of Keijiro. The former renderer is able to render millions of points using an octree algorithm, allowing for efficient processing of an entire scene at one time without much lag (*Fraiss, 2017*). This plugin also allowed for the changing of point shape, which helped fill in gaps between points, helping make the cloud appear to be a smooth object rather than many separate dots. This plugin was ideal for virtual reality applications due to its effectiveness and efficiency in rendering clouds as it does. The latter plugin, made by Kejiro, is very detail-oriented, however only effective for smaller point clouds. When attempting to render the entirety of the Hoover House scans, the plugin crashed the Unity engine. Although is was effective when a small example point cloud was uploaded, which contained less than 500,000 points. This was determined to not be useful for our particular application.

The last plugins used which was essential to our study were the Unity Standard Assets, and the Oculus Integration Assets. Both of these packages were found through the Unity Asset Store, and serve different purposes. The Oculus package allowed for the application to be walked through by a user wearing any Oculus-branded virtual-reality headsets, and made visualization in this manner very simple for us in the development process. The other package, the Standard Assets, played a key role in the WebGL mesh visualizations. This package included a prefab which created a movable character, including a first-person camera. This made it possible to create a first-person gaming experience to walk through the train station and beer cave meshes.3

# Results

Two final workflows were successfully created which met the original goals of this study. The first process was focused on the creation of a terrestrial point cloud visualization in virtual reality. The first step in this process was to collect scans with an appropriate scanner for the task at hand. These scans were then uploaded into a unique database in the Leica Cyclone software environment. Using the visual alignment registration method, the scans were knit together to form a cohesive point cloud which encapsulated the entire area of interest. This point cloud was exported as an .e57 filetype in order to be used in a wide variety of applications. The aligned cloud was then taken into the CloudCompare environment, where it was cleaned, refined, and re-exported using the same filetype. The Potree file converter tool was then implemented through the command line terminal to transform the .e57 file into a file which would be usable by a Unity 3D engine plugin created by Simon Fraiss. The Unity engine was opened, and saved to a local drive on our machine. The files created by the Potree converter were added to this project’s asset folder so it could be accessed through other tasks. Next, the point cloud visualization tool was downloaded via GitHub and added into the Unity project, which allowed us to efficiently load and view the entire point cloud at one time. The “cone” point-shape method was chosen to represent the cloud due to its continuity between points and the detail at which the points could be viewed. We then downloaded the Oculus Virtual Reality software package via the Unity Asset Store to allow for the exploration of the point cloud in virtual reality. This workflow enabled a realistic recreation of all models which we had ran through it, ranging from 3-million to over 200-million points being rendered. While there was minor frame rate drop in models with higher point counts, the overall execution of this workflow ended up being a success.

The second workflow which was created was focused on the exploration of a realistic object through a web-based build, similar to a first-person video game. First, we took images of our area of interest. We uploaded all of these images in to the RealityCapture software, which helped us align the images, calculate a 3D model derived from them, colorize the triangles created by the reconstruction, and overlay a texture on top of the model. This entire reconstruction was edited from within RC, exported from the program, and then brought into Unity 3D engine. A WebGL build was designated before an environment was built to allow for viewing on an online browser platform. A first-person player controller was downloaded from the Unity Standard Assets package which let the user walk around the scene. This entire scene was then built and uploaded to a server, enabling anyone to walk through on their computer. The WebGL process, without major mesh editing, seemed to be successful overall, with minor framerate drop when trying to walk through a large scene with high detail.

The results of the WebGL workflow can be found at the following addresses:  
[www.tg.skiwi.xyz/beerCaveV1](http://www.tg.skiwi.xyz/beerCaveV1) | [www.tg.skiwi.xyz/oldCapWebGL](http://www.tg.skiwi.xyz/oldCapWebGL) | [www.tg.skiwi.xyz/trainWebGL](http://www.tg.skiwi.xyz/trainWebGL)

# Discussion

This study has shown that it is possible to create an efficient, high-quality method of point cloud data in a virtual reality environment. These workflows are asily replicable, and can be understood by nearly anyone who would like to learn about it. While these is a slight learning curve for some of the programs, most of what needs to be done is simple in practice and does not need to be changed much in order to work with a variety of applications and data. This study will also help those without access to high-powered programs and computers, and those who may not have money to pay for expensive programs such as those produced by AutoDesk and Adobe.

This study could be of assistance in the field of data visualization and historical preservation. There are many companies, according to forums related to lidar and photogrammetric data, which are trying to find cost-effective methods for displaying these data in an effective manner. For sites that need a way to be preserved and displayed in a short amount of time – fragile buildings, disaster areas or delicate historical items – these methods can provide a team with the means to achieve their goals. It would also give them a great way to explore these areas in-depth after sites are closed, as well as let them study the fine details of items which may be too fragile to handle. There are countless possible uses of these workflows in academic and professional settings related to preservation and geographic exploration.

There are areas of improvement which could be made to these workflows in the future. While these particular programs may not be easily automated, there are multiple open-source programs which may allow for a more streamlined workflow. Software such as Meshroom, a photogrammetry processing tool, have very customizable workflow options for users to choose from. However, these programs tend to have a great learning curve due to the complexity and multitude of options that are offered within each step of the processing flow. In terms of automation, these programs would be ideal; in this particular application, they would not work well due to the overall intricacy of the processes. There is much more research which can be done into this area of visualization, and more effective options will most likely become available in the near future.

# References

Arnold, D, et al. “Tools for Populating Cultural Heritage Environments with Interactive Virtual Humans.” *EPOCH Conference on Open Digital Cultural Heritage Systems (2008)*, 2008, pp. 25–31.

Aziz, Kamarulzaman Ab., and Tan Gek Siang. “Virtual Reality and Augmented Reality Combination as a Holistic Application for Heritage Preservation in the UNESCO World Heritage Site of Melaka.” *International Journal of Social Science and Humanity*, vol. 4, no. 5, 2014, pp. 333–338., doi:10.7763/ijssh.2014.v4.374.

Fraiss, Simon M. “TU Wein Institute of Visual Computing and Human-Centered Technology.” *TU Wein Institute of Visual Computing and Human-Centered Technology*, TU Wein, Sept. 2017, www.cg.tuwien.ac.at/research/publications/2017/FRAISS-2017-PCU/.

Fraiss, Simon M. “GitHub.” *GitHub*, TU Wein, 4 Apr. 2020, https://github.com/SFraissTU/BA\_PointCloud.

Gray, David. “What Are the Most Popular Types of Point Cloud Processing Software?” *Vercator*, 2020, info.vercator.com/blog/what-are-the-most-popular-types-of-point-cloud-processing-software.

Haddad, Naif Adel. “From Ground Surveying to 3D Laser Scanner: A Review of Techniques Used for Spatial Documentation of Historic Sites.” *Journal of King Saud University - Engineering Sciences*, vol. 23, no. 2, 2011, pp. 109–118., doi:10.1016/j.jksues.2011.03.001.

Kersten, T. P., et al. “Development Of A Virtual Museum Including A 4D PresentationOf Building History In Virtual Reality.” *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2/W3, 2017, pp. 361–367., doi:10.5194/isprs-archives-xlii-2-w3-361-2017.

Keyser, Henry, et al. “The 4 Key Components to Photogrammetry Capture: Location, Stillness, Camera Settings and Camera Movement.” *Knight Lab Studio*, studio.knightlab.com/results/photojournalism3D/the-4-key-components-to-photogrammetry-capture/.

Kim, Changjae. “Object-based Integration of Photogrammetric and LiDAR Data for Accurate Reconstruction and Visualization of Building Models.” (2011).

King, Nigel. “QuestUAV Sensor Gimbals Improve Flight Efficiency By More Than 15 Percent: QuestUAV News.” *QuestUAV Ltd*, 26 Apr. 2017, www.questuav.com/media/news/questuav-sensor-gimbals-improve-flight-efficiency-15-percent/.

“Leica BLK360 Imaging Laser Scanner.” *Leica Geosystems*, leica-geosystems.com/en-us/products/laser-scanners/scanners/blk360.

“Leica Geosystems.” *Leica Geosystems*, Leica Geosystems, 2020, https://leica-geosystems.com/en-us/products/laser-scanners/software/leica-cyclone.

“LeicaScanStation P30.” *Leica ScanStation P30 / P40 Data Sheet - Leica Geosystems - PDF Catalogs | Technical Documentation | Brochure*, Leica Geosystems, pdf.directindustry.com/pdf/leica-geosystems/leica-scanstation-p30-p40-data-sheet/14324-610554.html.

“Lidar and Point Cloud Processing.” *Lidar and Point Cloud Processing - MATLAB & Simulink*, 2020, www.mathworks.com/help/vision/lidar-and-point-cloud-processing.html.

Mastrogiovanni, Luca, and Emiliano Casalicchio. “CloudCompare.” *CloudCompare*, https://www.danielgm.net/cc/.

“GitHub.” *GitHub*, 2019, https://github.com/keijiro/Pcx.

Schuetz, Markus. “Potree: Rendering Large Point Clouds in Web Browsers.” *Vienna University of Technology*, vol. 1, 19 Sept. 2016.

Schwarz, Sebastian, et al. “Emerging MPEG Standards for Point Cloud Compression.” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, 2019, pp. 133–148., doi:10.1109/jetcas.2018.2885981.

Technologies, Unity. “Solutions.” *Unity*, unity.com/solutions.

Wheeler, Andrew. “Photogrammetry Explained: The State of Reality Capture.” *Engineering.com*, 2016, www.engineering.com/DesignSoftware/DesignSoftwareArticles/ArticleID/13313/Photogrammetry-Explained-The-State-of-Reality-Capture.aspx.