Virtual Reality for Accurate and Efficient Classification of Point Clouds

Rebecca Whitten[1], Pradyumann Singhal[1], Dr. Thomas J. Pingel[2], Dr. J. Todd Ogle[3]

[1]Department of Computer Science, College of Engineering, Virginia Tech
[2]Department of Geography, College of Natural Resources and the Environment, Virginia Tech
[3]Applied Research in Immersive Experiences and Simulations Lab, University Libraries, Virginia Tech

Abstract

Both lidar (Light Detection and Ranging) and Structure from Motion (SfM) photogrammetry produce point clouds. Points in a point cloud can include attribute information such as color or intensity but not always semantic information about what that point is. The usefulness of point clouds is limited without high-quality semantic information.

We present work completed over the past year in the development of a virtual reality, immersive point cloud classification system. The system is deployed via a web-browser similar to Potree, and is built on open WebXR (i.e., augmented and virtual reality) components. It features the ability to load user point clouds in the common LAZ format, render them by classification color, change classification, and save the modified point cloud.

Future work will improve the user interface, add the ability to bring in additional geospatial layers to provide context, include support for the new COPC format, and conduct user testing. This represents an important advance for QA/QC in point cloud classification workflows, but more importantly provides a mechanism to generate a large volume of "ground truth" data to facilitate point cloud classification algorithm development.

Background

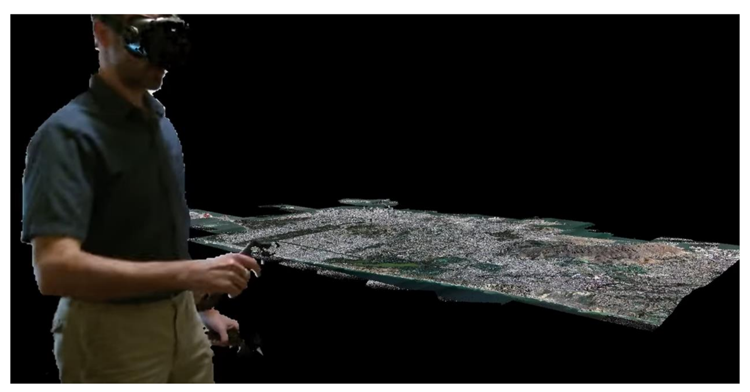
At county- and city-scales, point clouds are produced by specialized airborne platforms, but smaller-scale scans can be sourced from drones, handheld mobile units, and even smartphones. However, while the points in a point cloud can include attribute information, such as the color or intensity of the point, they do not include semantic information about what that point is. Algorithms exist to classify point clouds, but they at best operate at only 90-95% accuracy, and often far worse for point clouds sourced from mobile lidar or SfM.

To solve this problem, we ask: in what way can virtual reality be better incorporated into the visualization, interaction, and especially the classification of point clouds thereby leveraging the strong native spatial sense that virtual environments provide?

There have been several recent attempts to bring point clouds into Virtual Reality including:

- Previous work by Hudson Chase and Thomas Pingel, on which this work builds.
- Work by Nicolas Polys, Peter Sforza, Haitao Wang and others at CGIT and Immersive CAVE / Visionarium here at Virginia Tech.
- Oliver Kreylos, inventor of the Augmented Reality Sandbox, wrote a custom tool for immersive lidar visualization and manipulation.
- Oregon State University's Geomatics Research Lab conducted experiments using point clouds rendered on a 3D TV.
- CloudCompare allows for navigation of point clouds in VR.

Nicolas Polys demonstrating 3D visualization of a point cloud from Polys et al. (2017).



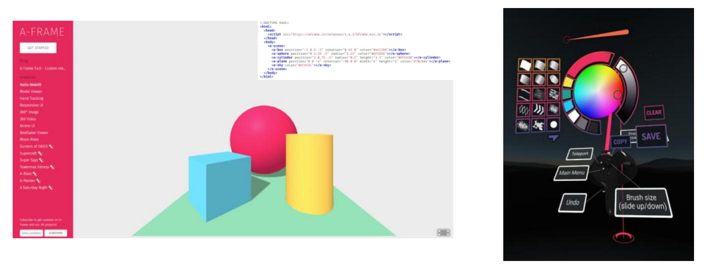
Oliver Kreylos demonstrates a custom-built immersive lidar visualization tool from YouTube demo.

Building Blocks

WebXR and A-Frame

WebXR is an Application Programming Interface (API) for allowing web content to use Virtual Reality (VR) and Augmented Reality (AR) devices. XR stands for "Extended Reality" and encompasses both VR and AR. WebXR provides advantages over standard VR/AR in that applications can be programmed to be compatible with most or all XR devices and can be accessed without having to download or install any software.

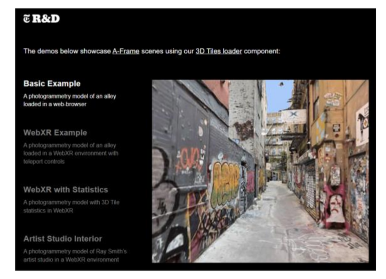
A-Frame is a programming framework for building web 3D, VR, and AR experiences. It is built on top of HTML and JavaScript, which makes it very easy to learn for anyone with web programming experience. The underlying code is built on the three.js library, which is a powerful 3D web library but can be complicated for beginners to use. We built our application using A-Frame along with some three.js for the more complex parts.



A-Frame is a way to author Web VR content in a lightweight, powerful way. The A-Frame demo A-Painter was an inspiration for our user interface.

NYTimes 3D Tiles Component

The New York Times R&D branch (rd.nytimes.org) created an A-Frame loader (nytimes.github.io/aframe-loader-3dtiles-component) for OGC 3D Tiles that allows the use of lidar and photogrammetric models to be explored in VR. Models can be created using RealityCapture or existing models can be converted using Cesium's ion interface. We used the New York Times aframe-loader-3dtiles-component as the reference to begin. They use the loaders.gl library to load the 3d tiles data and we used another part of loaders.gl library to modify the New York Times code to load point clouds from las/laz format.



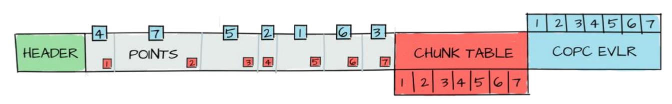
The NY Times aframe-loader-3dtiles-component, designed to allow 3D scenes that are freely explorable in VR to be embedded in news stories.

Potree and COPCs

Potree is a popular open-source WebGL based point cloud renderer created by Markus Schütz. Potree features excellent performance and a Virtual Reality interface for navigation but was not easily extended for manipulation of the attributes of points.

The new Cloud Optimized Point Cloud (COPC, copc.io) is a LAZ 1.4 file that stores point data organized in a clustered octree. The data in COPC is stored as octree in a variably-chunked LAZ data in a single file. This method of storage allows the data to be consumed sequentially by any reader that can handle variably-chunked LAZ 1.4 such as LASzip. It is intended to facilitate the streaming of point cloud data over the web, much like Cloud Optimized GeoTIFFs (COGs). Microsoft's Planetary Computer allows for the effective use of COPCs to more flexibly process lidar data in the cloud.

A point cloud converted to 3dtiles via Cesium ion and rendered using A-Frame.



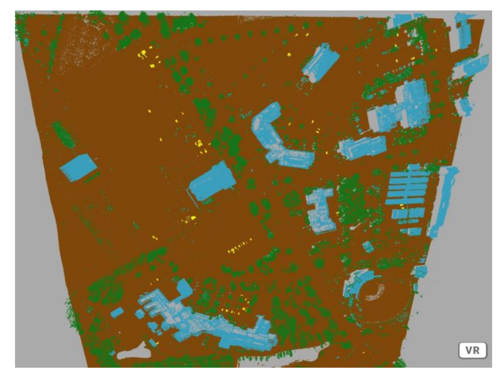
The Cloud Optimized Point Cloud data structure. (source:copc.io)

The Virtual Reality Classifier

Fixing Potree compatibility with A-Frame

We investigated using Potree as the centerpiece of the project, since Potree was designed to render billions of points in the browser efficiently. Mattias Treitler wrote an A-Frame Potree loader (aframe-potree-loader-component), but the component stopped working past A-Frame version 0.9.2. We fixed the package dependency issues to allow the aframe-potree-loader to work with A-Frame 1.3.0. However, this would require the user to convert their point cloud to Potree format prior to use, and we wanted to support LAS/LAZ files directly.

Potree lion rendered in A-Frame 1.3.0 with our component



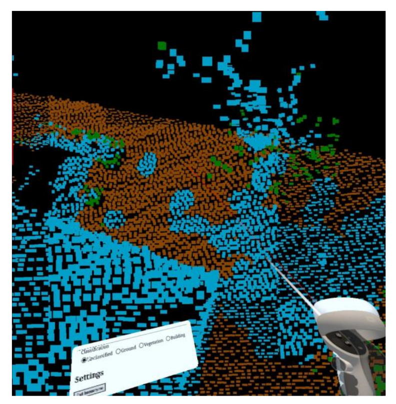
A classified point cloud rendered using our A-Frame laz loader component

Loading Laz files in A-Frame

We were able to successfully modify the NYT aframe-loader-3dtiles-component to use the loaders.gl LasLoader to load a laz/las file. We then used three.js to create the geometry and mesh of the pointcloud and render it to the scene.

Classifying Point Clouds in A-Frame

We use the oculus-touch-controls component in A-Frame to connect with the device controllers to raycast to the point cloud. Using manual auto-refresh every tick, the raycaster reports what points it has intersected with which is then passed into a classify function to change the classification values of those points and then re-render the scene.



VR User Interface

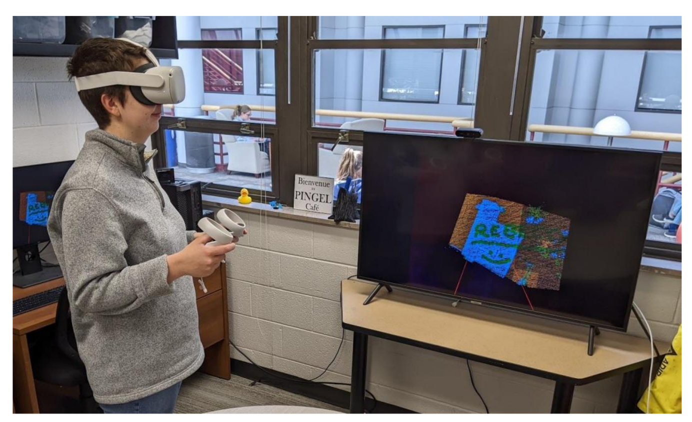
Exporting the point cloud

User Interface

The user interface integrates several Oculus / VR standards, including thumbstick teleportation/blink controls to move. The menus use AdaRoseCannon's htmlmesh system, which allows menu design to be written as an HTML component which is then rendered as a wrist-mounted menu.

Exporting the Point Cloud

After classifying the point cloud, we worked on exporting the point cloud with the newly classified data. We converted the data into a table and exported the data as csv which we can load as point cloud using software like Cloud Compare. We are working to incorporate the copc.js library to allow export of the point cloud in LAZ format.



A live demo of the system

Summary (TL;DR)

- Point clouds with high classification accuracy can be used for a lot of useful purposes
- Algorithms to automatically classify are not of high enough accuracy so manual classification has to be done
- VR can make classifying point clouds easier, more efficient and more fun for humans
- We have created a basic WebVR application using A-Frame that allows users to load a point cloud, classify points and then export it
- Live Demo at: nearearthimaginglab.org/vr_classifier

Code and examples available at

- **github.com/thomaspingel/aframe-potree-loader-component**
- **github.com/thomaspingel/aframe-loader-laz-component**

Next Steps

- **Export straight to LAZ** format for ease of use
- **Improvements to User Interface** – our efforts so far have been to develop a system that can get data in, allow VR to be used to classify, and to get data back out, but future work will refine and improve the user experience. For example, we currently navigate the cloud using teleportation controls, but future versions will do this and also allow the cloud to be treated as an object to be rotated, etc. The length of the raycaster will be configurable to avoid accidentally editing distant point points. We will add the ability to select eligible classes for which classifications can change (e.g., only allow ground points to change to vegetation for a specific classification task).
- **User testing** to compare against traditional methods and to solicit user feedback.
- **We plan to include a drag-and-drop interface** similar to lidarview.com or viewer.copc.io.
- **The ability to add additional geospatial layers**, including imagery and building footprint databases would improve the ability of the user to make decisions about classification.
- **Multi-user interaction** would allow more than one user to work on a point cloud at one time.
- **A paired desktop and VR experience**, to facilitate classifying the cloud in whatever modality is most appropriate.
- **Algorithm integration using PDAL** or other classification engine.
- **Algorithm development** using the high-accuracy point clouds created from this method.
- Potree A-Frame Component will be merged with original repo.

Acknowledgments

We thank the VT Libraries for funding the project via the Library Collaborative Research Grant program. We wish to thank Marc Nassar, Sarah Tucker, and Max Ofsa for their participation in the project. We also thank Hudson Chase for his early work on the project and Chris Kile for his work on point cloud classification which produced data used in this project.

References

Coltekin, A. (2002). An analysis of VRML-based 3D interfaces for online GISs: current limitations and solutions. Finnish journal of the Surveying Sciences, 20(1/2), 80-91.

Kreylos, O., Bawden, G. W., & Kellogg, L. H. (2008). Immersive visualization and analysis of LiDAR data. In Advances in Visual Computing: 4th International Symposium, ISVC 2008, Las Vegas, NV, USA, December 1-3, 2008. Proceedings, Part I 4 (pp. 846-855). Springer Berlin Heidelberg.

Polys, N., Hession, C. W., Sforza, P., Munsell, J., Taylor, A., Wang, H. 2017. Lidar Pipelines for Immersive and Web3D Visualization. SilviLaser 2017.

Schütz, M., Ohrhallinger, S., & Wimmer, M. (2020, October). Fast Out-of-Core Octree Generation for Massive Point Clouds. In Computer Graphics Forum (Vol. 39, No. 7, pp. 155-167).

Wang, H., Chen, X., Polys, N. & Sforza, P. (2017). A Web3D Forest Geo-Visualization and User Interface Evaluation. In Proceedings of the 22nd International Conference on 3D Web Technology (Web3D '17). ACM, New York, NY, USA.