Web-based Deep Segmentation of Indoor Point Clouds

Z.H. Chen[a], E. Che[b], F.X. Li[a], M.J. Olsen[b], and Y. Turkan[b]

[a]School of Electrical Engineering and Computer Science, Oregon State University, United States
[b]School of Civil and Construction Engineering, Oregon State University, United States
E-mail: chenzeh@oregonstate.edu, chee@oregonstate.edu, Fuxin.Li@oregonstate.edu,
Michael.Olsen@oregonstate.edu, Yelda.Turkan@oregonstate.edu

Abstract –

Deep learning and neural networks have empowered many tasks including semantic segmentation and image classification. Recently, novel neural networks were proposed that could directly process three-dimensional (3D) point clouds. In this paper, we present a software tool incorporating a variety of measuring tools to analyze and validate raw point cloud data while enabling an interactive segmentation of point clouds using deep learning. One of the primary advantages of implementing deep learning for point cloud segmentation is that it enables feature extraction to be learned through neural networks based on a large amount of data. Our software tool allows users to visualize 3D point cloud data sets containing millions of points in standard web browsers and process 3D point clouds using deep segmentation with deep neural networks. The interaction tool can assist with distinguishing structural buildings elements from non-structural objects in a given dataset.

Keywords –

Point cloud; Deep learning; Web-based interaction

1 Introduction

Point clouds are three-dimensional (3D) models that consist of points compared with other 3D model representations, such as meshes, volumetric models, and depth maps. Point clouds are obtained by scanning the real world using an 3D imaging method such as laser scanning (light detection and ranging, lidar) or photogrammetry. Point cloud data can be transformed into other 3D model representations and products by further processing, modeling, and analysis.

Point cloud data can be utilized in a variety of applications, such as construction progress monitoring, building structural analysis, and generation of 3D models. In addition, point clouds are also widely used in real-time perception applications such as the use of lidar perception in autonomous driving vehicles or robotic SLAM (simultaneous localization and mapping) systems. For most of these applications, point clouds are typically treated as raw data, which requires it to be processed and analyzed to extract useful information.

Building Information Modeling (BIM) is a process involving the generation and management of digital representations called "Building information models" (BIMs), which are files which can be extracted, exchanged or networked to support decision-making regarding a building or other built asset. It is also enabling its users to share and compare their models seamlessly throughout the whole lifecycle of their project, from planning to demolition. It offers a variety of benefits including prevention of data loss from one phase to another, as well as improved visualization and coordination between different trades involved in a project. BIM implementation in the construction industry has increased exponentially in the last decade. Scan-to-BIM is a process of 3D laser scanning a physical space or site to create an accurate digital representation through BIM. This representation can be used for design, progress tracking or project evaluation. The data collection in Scan-to-BIM helps eliminate human error that may occur otherwise when using traditional surveying methods and it enables collection of a high volume of data over a short amount of time. Next, the scanning data (i.e. point cloud) need to be shared with the project team; this is done most commonly by importing the data into the project's common data environment (CDE), which enables the team members to analyze, visualize and model the point cloud. This end-to-end approach delivers more detailed and accurate information about a project and enables one to verify the progress of the work in an objective manner.

Different applications usually employ different approaches to extract features from point clouds. Point features often encode certain statistical properties of the points, are designed to be invariant to certain transformations, and can be classified as intrinsic or extrinsic parameters. They can also be categorized as local features and global features. For a specific task, it is nontrivial to find the optimal feature combination. Deep learning can provide end-to-end training which automates the feature extraction process by learning from existing point cloud data, and could significantly reduce data complexity for modeling and data analysis in an automated Scan-to-BIM framework.

In the big data era, WebGL and server framework over web browsers has become increasingly popular. Such architectures are supported over a standard browser, and enables developers, artists, companies, researchers to share their data with a wider audience and access their work on any device in any place without the need of installing additional software.

Furthermore, the widespread use of neural networks demands high-performance computational resources, (e.g., multi-core CPUs and GPUs), especially when users are required to work remotely without powerful computing resources. Web-based client-to-server communication architecture and the application of deep learning on the server side allow users to work remotely with massive 3D data regardless of the device they use. Currently, several web-based services, such as Sketchfab and Potree allow users to upload, share and view content without any knowledge about the underlying 3D modeling and geometry mechanics. But to our knowledge no deep learning service have existed.

This paper introduces a prototype web-based end-to-end point cloud processing toolbox that is capable of loading and viewing point cloud data. Furthermore, it is capable of applying neural network semantic scene segmentation and annotating point clouds. This work is expected to be a basic building block to a complete Scan-to-BIM framework.

2 Related Work

2.1 Web-based Massive Point Cloud rendering

Potree [15], a web-based renderer for large point clouds, allows users to view data sets with billions of points, obtained from sources such as lidar or photogrammetry, in real time in standard web browsers. The focus of Potree is on large point clouds, and its measuring tools allow users to visualize, analyze and validate raw point cloud data without the need for a time-intensive and potentially costly meshing step.

PointCloudViz server [19] and the corresponding web client are commercial services by Mirage Technologies, which complement their free desktop lidar viewer.

The streaming and rendering of billions of points in web browsers, without the need to load large amounts of data in advance, is achieved with a hierarchical structure that stores subsamples of the original data at different resolutions. A low-resolution version is stored in the root node and with each level, the resolution gradually increases. The structure enables Potree to neglect regions outside the view frustum, and to render distant regions at a lower level of detail.

ShareLIDAR [16] is a multi-resolution point cloud renderer with hosting service. Its notable features include illumination through normals, an orthographic top view, a sectioning (height-profile) tool and the adjustment of point sizes. However, it loads data in smaller tiles that do not cover the whole data set, which results in a large empty space within the display while the user waits for the data to stream in.

2.2 Point Cloud Feature Extraction

3D point clouds are typically processed using point cloud features that are customized for specific tasks. Point features often encode certain statistical properties of points that are usually classified as intrinsic or extrinsic parameters of transformations, or they are categorized as local and global features.

2.3 Deep Learning for Processing Large Scale Point Clouds

Deep learning is a recent technique used for point cloud processing. Previous research on 3D Convolutional Neural Networks (CNN) convert 3D point clouds to 2D images or 3D volumetric grids. Qi et al. [18] and Hang et al. [9] developed methods that project 3D point clouds or shapes into several 2D images, and then apply 2D CNN for classification. Although these approaches have achieved great results for shape classification and retrieval tasks, they cannot be extended for high-resolution scene segmentation tasks. [22, 9, and 6] presented approaches that voxelize point clouds into volumetric grids by quantization and then apply 3D CNN. These approaches are constrained by the volumetric resolution of the 3D point cloud; further, the computational cost of 3D convolutions is a significant disadvantage, especially when processing very large point clouds. [11] improves the resolution significantly by using a set of unbalanced octrees where each leaf node stores a pooled feature representation. Kd-networks [4] compute the representations in a feed-forward bottom-up fashion on a Kd-tree with a certain size. In a Kd-network, the input number of points in the point cloud needs to be the same during training and testing, which unfortunately does not hold for many tasks.

In [8], an effective feature learning method called PointNet is introduced that directly applies the raw data without preprocessing the point clouds. Point clouds are simple and unified structures and are easier to learn from compared to meshes or other types of 3D models.

2.3.1 Building Information Modeling

Building Information Modeling (BIM) is both a technology and a process that results in easer collaboration between involved parties by using a virtual

3D model of a structure. From the technology perspective, BIM is an object-oriented database that hosts both geometric and non-geometric information such as material types, costs, and scheduling. BIM has been adopted at a growing rate in the construction industry. It is used for a variety of tasks including design coordination (commonly known as clash coordination/detection), constructability analysis, and 3D/4D analysis, amongst others.

Both geometric and non-geometric data stored in BIM should be well defined when developing the BIM model. BIM data is typically stored online on a data cloud to manage and share information among members of a team working on the same project as well as to avoid the potential complications that may arise from the use of models that are not up-to-date.

One of the objectives of the system proposed in this paper is to ensure that it can assist users with model visualization and data management, which are essential for data collaboration and vital to the success of BIM.

2.3.2 System Design

To overcome the weaknesses inherent in previous point cloud processing and visualization approaches, this study aims to provide an efficient data management and visualization solution with a user-friendly interface.

Figure 1 illustrates the major nodes for achieving the goal with their individual function.

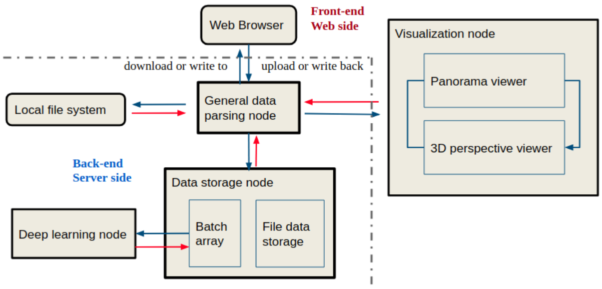


Figure 1. Data flow in the major functionality module of the proposed system

The system can be divided into the front-end and the back-end. The front-end provides the user interaction interface and enables visualization of point clouds using three different approaches. The front-end interaction interface performs several operations, including storing and parsing data into the designated format, which are sent to the back-end. Data parsing is automatically determined by the "General Data Parsing Node". Figure 1 shows that files can be uploaded into the system online or directly by the administrator in the server's file system. After the data processing is complete, the user can download the results or the data that have been processed (such as an intermediate file that is parsed in a different format, a semantic segmentation result, or a 2D panoramic image).

The general data parsing node is one of the core modules of the proposed system. It enables to use different file formats for the input file.

The use of an intermediate file makes sure that for each function module, the input format requirement is not the main hindrance to data flow through the system. This also enables all modules to use the most appropriate file format in order to decrease possible programming difficulties and compatible language problems. The intermediate file communication method ensures the system focuses on the portability and reusability and achieves low coupling, which is convenient for further development and reusing of the code.

After the data been parsed by the "General Data Parsing Node", the formed data will be stored in the data storage node. The data herein has been parsed into two structures. The first structure is in 2D array panorama form, which lists the point cloud information in a 2D array structure following a 2D panorama image sequence that can be traced back to each of the scanners that is used to acquire data. The second structure is in a batch array format, from which we can directly transfer to the neural network in order to train the network for further improvement or simply calculate the semantic segmentation result with the point cloud data from a single scan by going through the forward propagation. We will discuss how data are formed in all these intermediate files in detail in the next section.

The highlight of this system is our "Deep Learning Node", which is an abstract interface that can trigger deep learning related algorithms and trained neural networks obtained in a run. In our extendable system, the neural network model can be changed when desired. Researchers can replace or upgrade the model when they want to have a different model architecture of a network in order to improve the performance, or have a new model that can work on a different task.

Another highlighted feature in this system is the "Visualization Node", which provides a method to visualize the raw data and the analyzed result applies on the raw data views. The panorama viewer and the 3D perspective viewer provide two different ways for users to visualize the point cloud data and task results. The Panorama viewer provides the ability to show the raw data as a panoramic image with original color or by the instance segmentation result processed by the Norvana [2] algorithm. It provides several useful tools that can assist with the organization of the point cloud data and enables users to perform annotation operations on panorama image directly through a canvas. The 3D perspective viewer provides the basic visualization window for viewing the point cloud model in 3D from different perspectives.

2.3.3 Deep Semantic Segmentation of Point Clouds

In this work, the PointNet [8] architecture is utilized in the deep learning function node (Figure 2). The structured batch array, which contains the information per point in an indoor point cloud scene, will be fed to the proposed network. Each point has 9 attributes, including X, Y, Z, R, G, B, and 3D normalized location with respect to the room (from 0 to 1).

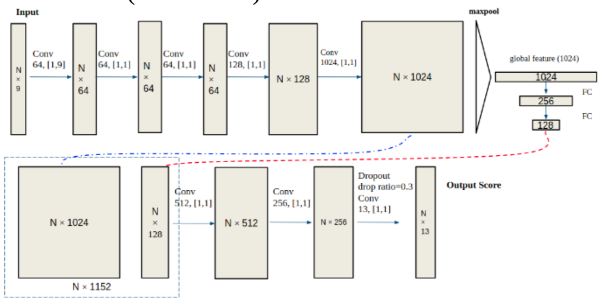


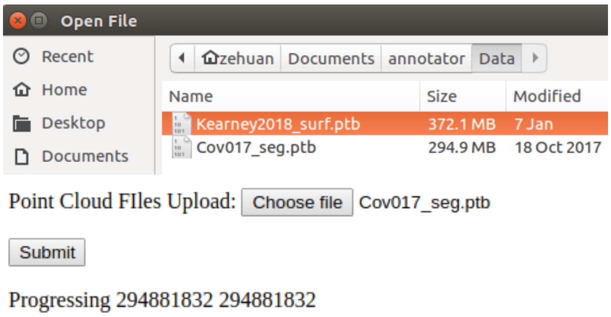
Figure 2. The network architecture based on PointNet baseline structure

The architecture contains 3 consecutive, multiple layer perceptions (MLP). In our implementation, we use an equivalent form of convolutional filters with a size of 1×1 instead.

After the feature learning, resulting in 1024 new point features, we apply a max-pooling layer to down-sample and aggregate these point features into global features. We then make the global features pass through a fully connected (FC) layer to aggregate to 128 features. These 128 features, which are tiled to the same dimension as the number of points, are then concatenated with the previous N×1024 features. After the next several convolutional layers that are similar to previous operations, we obtain the output score of a 13-dimensional array after the softmax layer.

2.3.4 Showcase and Functionality Introduction

The operation with user friendly interaction will lead user to go through the whole pipeline that finish their task step by step.



Name	Size	Modified
Kearney2018_surf.ptb	372.1 MB	7 Jan
Cov017_seg.ptb	294.9 MB	18 Oct 2017

Point Cloud FIles Upload: Cov017_seg.ptb

Progressing 294881832 294881832

Figure 3. File select (top) and file upload (bottom) interfaces

Figure 3 explains the very first step where a user uploads a data file to the server, and then internally the file parsing node on server side will reformat the data.

After data has been reformatted, the panorama image viewer will pop up and show a panoramic image with original color and with three buttons to allow the following operations: image rotation operations, "show instance segmentation," and "show semantic segmentation."

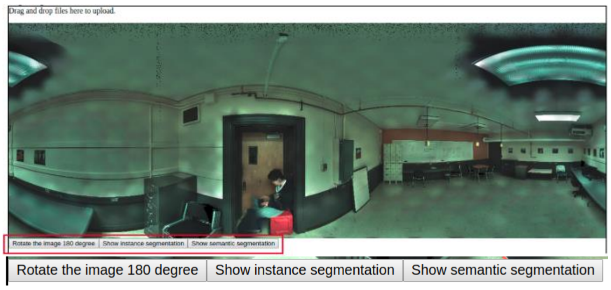
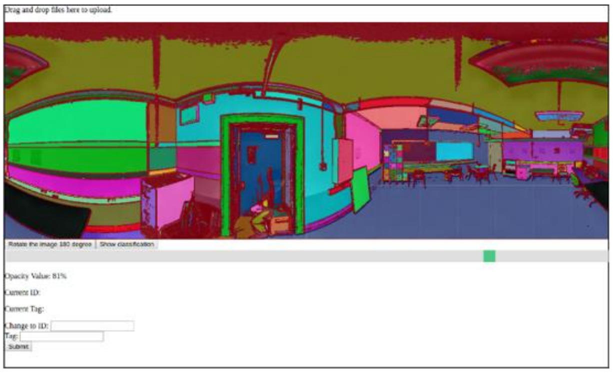

Drag and drop files here to upload.
Rotate the image 180 degree
Show instance segmentation
Show semantic segmentation

Figure 4. Panoramic image view

The result of the instance segmentation will show up and generate the canvas overlay on the original color image. The instance segmentation results were generated according to the algorithm [2] using test data collected from a building (Covell Hall) at Oregon State University main campus. This panoramic image is shown with photographic color in different image fuse ratio (opacity). The interface also includes the panel to show which color block indicates which class ID. In that case, people can easily access the classification result directly from the algorithm and perform manual annotation, or refine the result. The user can further improve the automated results of the algorithm or annotate the dataset with a real ground truth classification result and feed it into deep neural network.


Drag and drop files here to upload.
Rotate the image 180 degree
Show classification
Submit

Opacity Value: 81%

Current ID:

Current Tag:

Change to ID:

Tag:

Figure 5. Panoramic image view with colormap overlay representing the instance segmentation results obtained using Norvana algorithm

2.3.5 Instance Segmentation

The classification and segmentation algorithm are the core operation module in our system. Norvana, introduced in [2] is a highly efficient and novel approach to segment point clouds. Segmentation is a common procedure of post-processing to group the point cloud into a number of clusters to simplify the data for the sequential modelling and analysis needed for most applications. Norvana rapidly segments laser scan data based on edge detection and region growing. The algorithm computes incidence angles and then performs normal variation analysis to separate silhouette edges and intersection edges smooth surfaces. A modified region growing algorithm groups the points lying on the same smooth surface. The proposed method efficiently exploits the gridded scan pattern utilized during acquisition of laser scanning data from most sensors and takes advantage of parallel programming to process approximately 1 million points per second. Moreover, the proposed segmentation does directly not require estimation of the normal at each point, which limits the errors in normal estimation propagating to segmentation. Figure 6 shows an example scan segmented by Norvana.

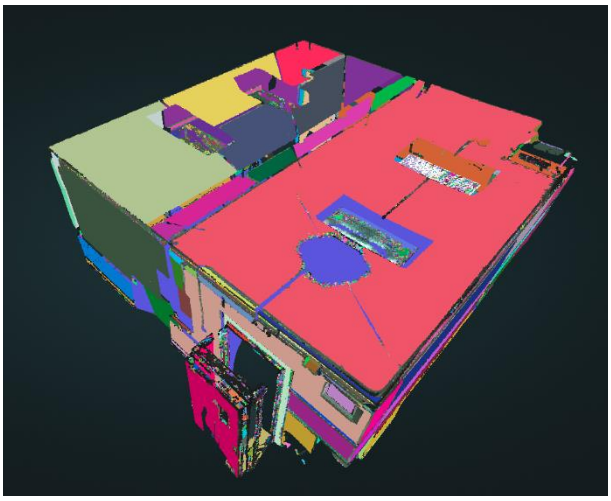


Figure 6. Norvana segmentation results in 3D perspective view via Potree

Along with the appearance of the 2D panorama with instance segmentation overlay and tools, the 3D viewer of the instance segmentation result is also shown (Figure 6). Orbit navigation with mouse and keyboard are supported. This viewer is powered by a Nested octree data structure. Points are rendered in hierarchal order to minimize the use of computational resources and avoid crashing the browser.

Table 1. Indoor object classes

Structural classes		Non-structural classes	
Column	Wall	Door	Window
Ceiling	Floor	Bookcase	Board
Beam		Chair	Sofa
		Table	Clutters

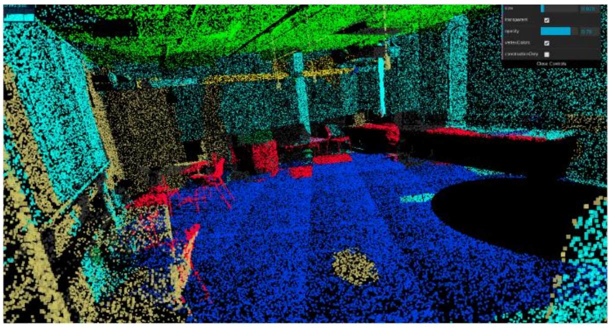


Figure 7. Original semantic segmentation result showing all classes

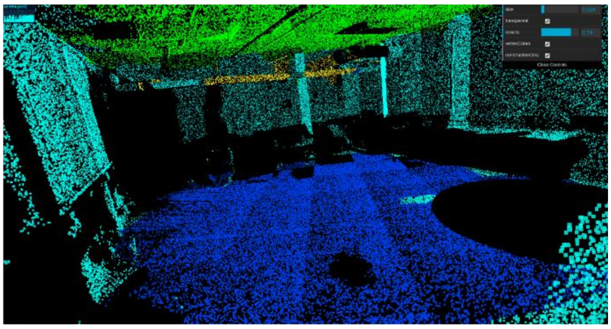


Figure 8. The scene after the semantic filter application to remove the non-construction class objects in the point cloud

3 Experiment

This section details the experiment conducted on the S3DIS (3D Semantic Parsing of Large-Scale Indoor Spaces) dataset that is visualized on the 3D viewer and segmented using deep neural network [1].

The S3DIS dataset (Figure 9) is composed of five large-scale indoor areas from three different buildings, each covering approximately 1900, 450, 1700, 870 and 1100 square meters (total of 6020 square meters). These areas show diverse properties in architectural style and appearance, including office areas, educational and exhibition spaces, conference rooms, personal offices, restrooms, open spaces, lobbies, stairways, and hallways.

One of the areas includes multiple floors, while all the other areas include only one floor. All the point clouds are automatically generated without any manual intervention using the Matterport scanner.

The dataset contains 3D scans of six different areas including 271 rooms. Each point in the scan is annotated manually with one of the semantic labels from the 13 categories (chair, table, floor, wall, etc., and clutter). We can further distinguish these 13 classes as a structural object (e.g., floor, ceiling wall, etc.) or a non-structural object (e.g., table, sofa, door, etc.). With the tag of different classes, we can visualize the two main clusters (structural vs. non-structural, Figure 7) or view individual classes (or combination) via checking each class of interest (Figure 8).

The training data points are split room by room, and the rooms were split into blocks with 1m x 1m area. Then the segmentation algorithm in PointNet was trained to predict point classes of each block. Each point is represented by a 9-dimensional vector of XYZ, RGB and normalized location as to the room (from 0 to 1). For training, 4096 points were randomly selected from each block on-the-fly. After training, all the points in the testing dataset are tested on.

The precision-recall curve of the segmentation is calculated for both construction structure and non-construction structure points (Figure 9). This result is transferred from original 13-class semantic segmentation for each point into binary classification.

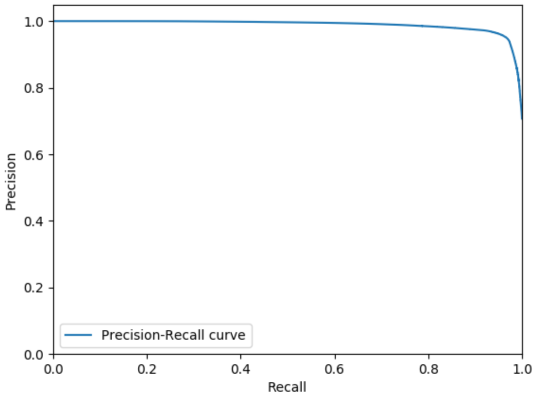
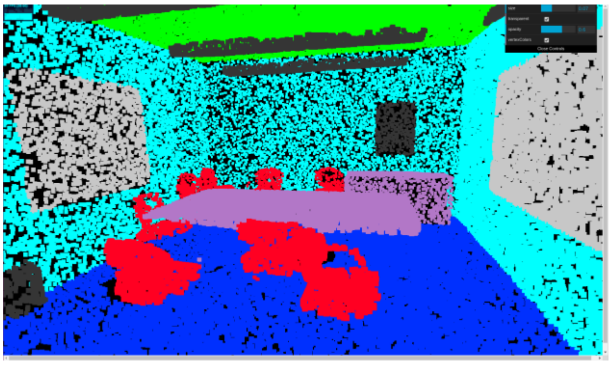


Figure 9. Precision-Recall curve for structural and non-structural objects classification in the S3DIS dataset



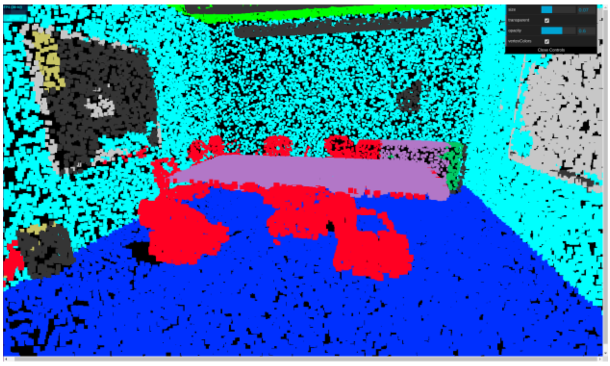


Figure 10. Conference room scene in S3DIS dataset. The ground truth semantic labelling result (top), and the prediction result (bottom) in 3D perspective viewer.

Figures 10-12 present three different point clouds of indoor scenes, selected from the S3DIS dataset. Each figure contains ground truth semantic labeling and the prediction results showing different classes.

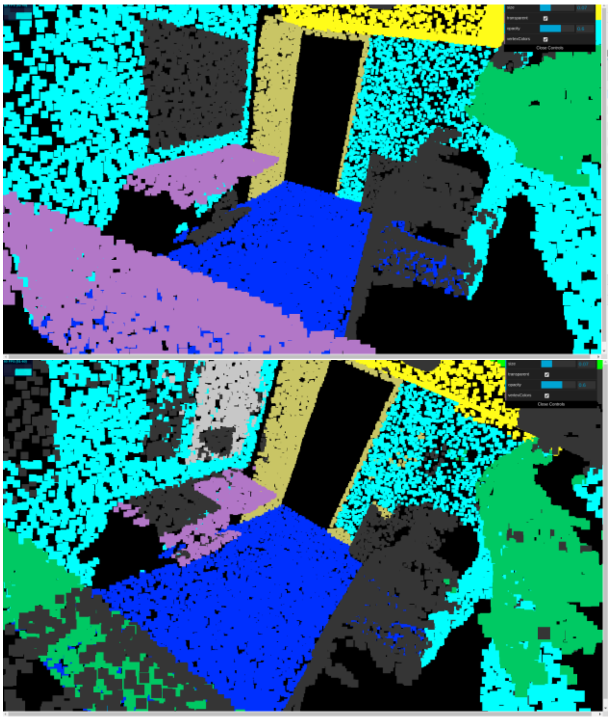


Figure 11. Copy room scene in S3DIS dataset. Top view shows the ground truth, and the bottom view shows the prediction labelling result

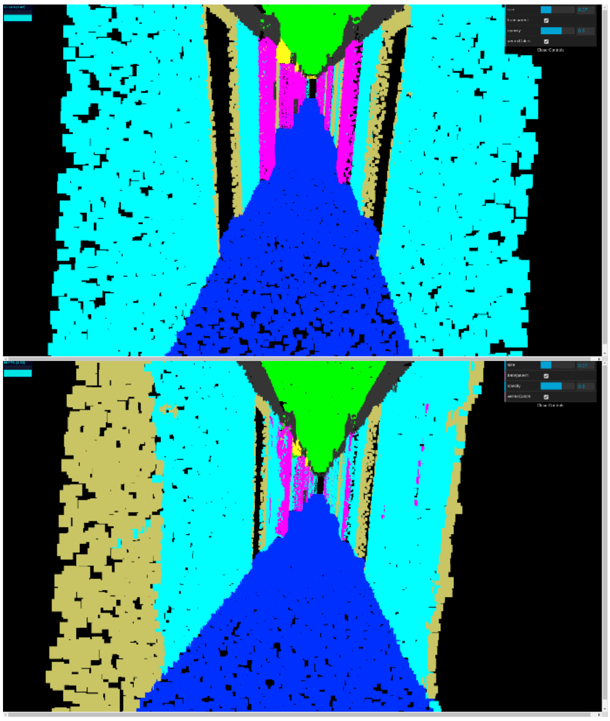


Figure 12. The ground truth (top), prediction result (bottom)

In Figure 10, the board and the clutter were slightly mismatched since the white board and the paint (belong to clutter class) could not be differentiated by the segmentation algorithm. However, tables and chairs were classified correctly with very little mismatch using PointNet.

In Figure 11, in the copy room scene, semantic labeling was performed with high accuracy. However, there were still some mismatches. For example, the table was mismatched with the part of ceiling on the bottom-left corner of the figure.

In Figure 12, a hallway scene is presented. This scene does not contain many corners or small objects. Thus, the segmentation of walls, floor and ceiling were done successfully. The figure also demonstrates that the segmentation prediction result is very close to the ground truth.

4 Conclusion and Future work

This paper presented a system that is capable of rendering 3D point clouds with millions of points real time in standard web browsers. It uses intermediate data storage format and data stream parsing to ensure that the proposed system is compatible with different file formats. In other words, it can process data regardless of the input file format and the neural network architecture. Furthermore, utilization of asynchronous mechanism and recall function during the front-end step dramatically reduces the total data download / visualization time and improve the runtime speed. The general file storage node and multiple view approach enable the system to render raw point cloud data in different representations, including panoramic and 3D viewer. To achieve a better visual quality, methods such as a multi-layer canvas interaction interface was implemented in order to increase the computing efficiency and to enable visualization and input interaction process in parallel.

Deep semantic segmentation results obtained from indoor scene point cloud data obtained from the S3DIS dataset are also presented, where each point in the dataset was labeled as structural and non-structural classes.

In future work, the point cloud rendering efficiency of the system will be improved, and other interaction tools including the synchronous correspondence of two viewers will be added. Furthermore, point cloud selection tools should be added to the 3D perspective viewer. Another issue that needs to be addressed in future work has to do with filling the empty spaces, i.e. “holes”, after filtering out the non-construction objects. More efficient point cloud semantic segmentation algorithms and network architecture could also be applied to the more recent PointConv [22] architecture.

Acknowledgements

Contributions by co-authors Che and Olsen were funded by the National Science Foundation (Award CMMI-1351487).

References

[1] Iro Armeni, Sasha Sax, Amir R Zamir, and Silvio Savarese. Joint 2d-3d-semantic data for indoor scene understanding. arXiv:1702.01105, 2017.

[2] Che, E. and Olsen, M.J. Multi-scan segmentation of terrestrial laser scanning data based on normal variation analysis. *ISPRS Journal of Photogrammetry and Remote Sensing*, 143:233-248, 2018.

[3] URL:https://www.khronos.org/registry/webgl/specs/1.0/ [Online; accessed 22-Oct-2018].

[4] Roman Klokov and Victor Lempitsky. Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. In Computer Vision (ICCV), 2017 IEEE International Conference on, pages 863–872. IEEE, 2017.

[5] matplotlib.org. Choosing Colormaps. https://matplotlib.org/users/colormaps.html#grayscale-conversion, 2015. [Online; accessed 28-Dec 2018].

[6] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In Intelligent Robots and

Systems (IROS), 2015. IEEE/RSJ International Conference on, pages 922–928. IEEE, 2015.

[7] Overview and Comparison of Features. https://github.com/PointCloudLibrary/pcl/wiki/Overview-and-Comparison-of-Features, 2015. [Online; accessed 28-Oct-2018].

[8] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. Proc. Computer Vision and Pattern Recognition (CVPR), IEEE, 1(2):4, 2017.

[9] Charles R Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J Guibas. Volumetric and multi-view cnns for object classification on 3d data. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 5648–5656, 2016.

[10] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In Advances in Neural Information Processing Systems, pages 5099–5108, 2017.

[11] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, volume 3, 2017.

[12] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In Robotics and Automation, 2009. ICRA'09. IEEE International Conference on, pages 3212–3217. Citeseer, 2009.

[13] Radu Bogdan Rusu, Zoltan Csaba Marton, Nico Blodow, and Michael Beetz. Learning informative point classes for the acquisition of object model maps. In Control, Automation, Robotics and Vision, 2008. ICARCV 2008. 10th International Conference on, pages 643–650. IEEE, 2008.

[14] M Schutz. Potreeconverter-uniform partitioning of point cloud data into an octree, 2014.

[15] Markus Sch ̈utz. Potree: Rendering large point clouds in web browsers. Technische Universit ̈at Wien, Wiede ́n, 2016.

[16] URL: https://www.sharelidar.com/ [accessed 22-Oct-2018].

[17] URL: https://sketchfab.com/ [accessed 22-Oct-2018].

[18] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In Proceedings of the IEEE international conference on computer vision, pages 945–953, 2015.

[19] Mirage Technologies. PointCloudViz Server. http://server.pointcloudviz.com/, 2015. [Online; accessed 22-Oct-2018].

[20] URL: https://threejs.org/, 2015. [accessed 22-Oct-2018].

[21] Oriol Vinyals, Samy Bengio, and Manjunath. Kudlur. Order matters: Sequence to sequence for sets. arXiv preprint arXiv:1511.06391, 2015.

[22] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. arXiv preprint arXiv:1811.07246, 2018.

[23] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1912–1920, 2015.