

Report for "Praktikum aus Visual Computing"

Lukas Gersthofer, 1325669

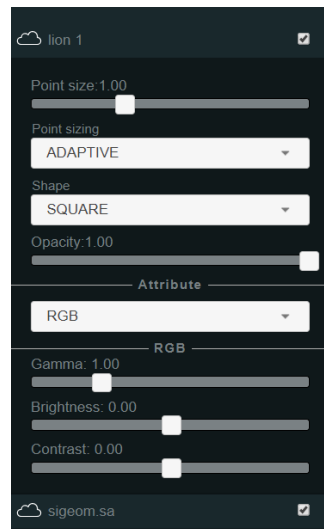
August 29, 2017

Abstract

Potree is an interactive web-based point cloud renderer. During an internship I was responsible for extending the application by a set of new features. That includes a new tool for clipping, an orthographic camera mode and some more minor features which are explained in this report. The following report contains information about the changes made to the Potree application.

1 Pointcloud material

Although the old version of Potree already supported materials to define the appearance of point clouds, a global material was stored and used for every item in the scene. This was extended to support one material per point cloud. In addition the visual layout of the scene graph was redesigned and rewritten. The new overview is shown in the right figure. Every point cloud in the scene is listed and visualized with a small icon, its name and a checkbox to set its visibility. When clicking on the list item, it expands to show all material properties. The former global material is now stored for every point cloud. The definition of the property moved from *viewer.js* to *PointCloudOc-tree.js*.



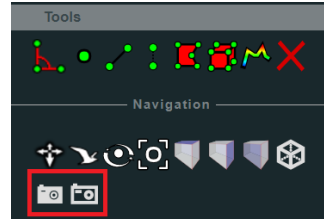
2 Unit of length

Since positions and distances of point clouds are measured in a unit of length created by humans, it makes sense to display the according unit (by default meter) when rendering such data. For example displaying is necessary for length, area and volume measurements. Because not all point clouds are measured in meters, Potree lacked the feature to select a specific unit of length. This feature was added to the file *viewer.js* and is currently available in the scene section of the side bar.



3 Orthographic camera

Until now Potree only supported a perspective camera but for some purposes an orthographic view of the scene might give better insights than the perspective one. Architects and civil engineers often need the scenes not to be distorted, especially when working with plots or buildings. Therefor an additional camera mode in the tools section is introduced to switch between perspective and orthographic camera. The orthographic camera supports viewing the scene, picking and an adaptive point size. However, the adaptive point size turns out to be not quite efficient when viewing large scenes but this might be compensated by tuning the octree node refinement for the orthographic camera which is illustrated in figure 1 (not done in this work). In figure 2 the differences between a scene rendered with perspective and orthographic camera are illustrated.



Like the perspective camera, the orthographic one is defined in the class *Scene*

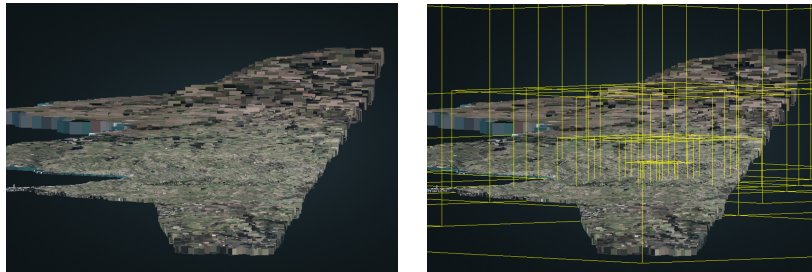


Figure 1: Adaptive point size fails for large point clouds. In the left image the points farther away from the focused points are drawn too big. This is caused by a non-uniform node refinement seen in the right image.

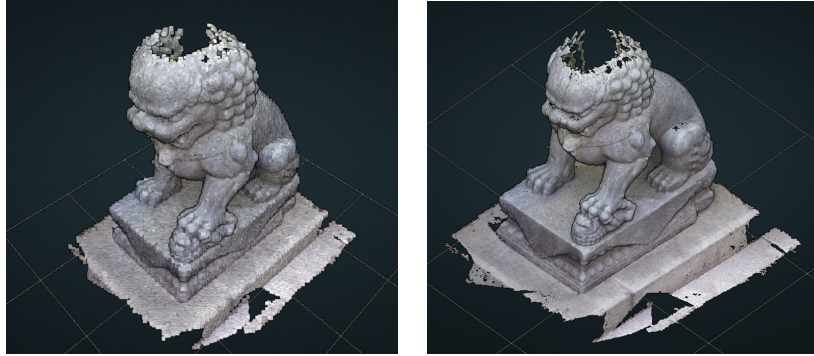


Figure 2: A comparison of perspective and orthographic camera. The perspective view is shown in the left image and the orthographic one in the right.

in the file *viewer.js*. The currently active camera is obtained by calling the scene objects method *getActiveCamera()*. All transformations applied to one camera are also applied to the other one in order to guarantee a smooth transition when switching cameras.

4 Navigation cube

Combined with an orthographic view it is often advantageous to view the scene from special positions like top view, front view or lateral view. Potree already provides functionality to switch between a top, front and left view of the scene. To extend this to all sides and to provide an overview of the current orientation of the scene, a navigation cube was integrated into the application. This cube copies the rotation of the scene and automatically transitions to a specific view when clicked on a face. When clicked, the camera is first translated and rotated to view the selected



face and the scene is then fitted to the cameras frustum. The class *Navigation-Cube* stores geometry information of the cube and handles on-click events. A reference to the displayed cube is stored in the *Viewer* class in *viewer.js*.

5 Clipping

The newly developed clipping tool encompasses a new UI, a selection of clipping modes and 2 new clipping volumes. The clipping section of the side bar now has the option to toggle between different clipping modes, create new clipping volumes and it also lists all the current clipping volumes of the scene. It is shown in the image to the right. Clipping modes define how points inside or outside of a volume are treated. Currently 4 different modes are available:

- Disable: only contours of the clipping volumes are drawn.
- Highlight: All points inside of any volume are highlighted in red.
- Inside: Points inside of any volume are clipped away.
- Outside: Points outside of any volume are clipped away.

This mode is global and determines the behavior for every clipping volume.

The toolbar below provides functionality to create new clipping volumes or to delete all current ones. There are 2 new types of volumes to clip: clipping planes and clipping polygons.

The first one is a plane extended to a box via a size property. This type of volume can be scaled and positioned. For positioning it is possible to translate or rotate the object by a specific offset according to the local or global coordinate system. A local coordinate system displayed on the plane helps to orientate. At creation, a plane is aligned with one of the global coordinate axes (x, y, z). Therefore, 3 different buttons for all 3 axes are provided.

The other type of clipping volume, named clipping polygon, is defined as a polygon extruded along the current viewing direction. The polygon is drawn in screen space by the user and is then extruded according to the current parameters of the camera. Thus, a polygon drawn with a perspective camera will be different from one drawn with an orthographic camera. The maximum number of vertices of a polygon is 8 and there is a maximum of clipping polygons of 10.

Every volume can be deleted by clicking on the right cross of the according volume's property section.

In figure 3 clipping of a point cloud is illustrated.

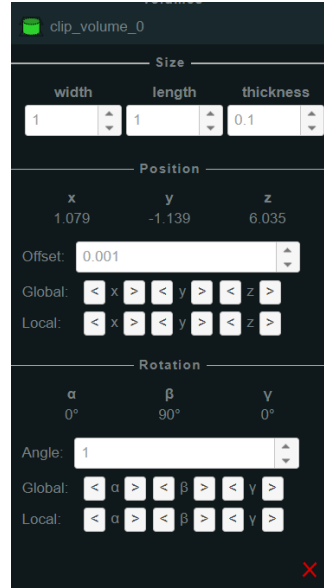
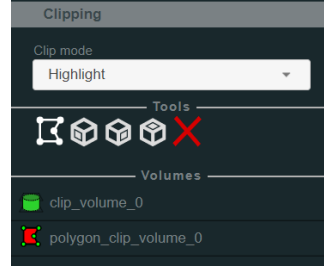




Figure 3: The *lion* point cloud is clipped by a clipping plane, a perspective clipping polygon and an orthogonal polygon.