Point Clouds with Potree

Iezzi Valerio, Zizzo Luigi, Organtini Eleonora

***Abstract*— We present a web application designed to view clouds of points and manage information about them with the possibility to interact in an interactive way with the tools provided by Potree. The user can use a WebGL to get many cloud points. It was carried out using Google APIs, a map that allows us to locate the point clouds. The end user has the opportunity to have an overview of all clouds of detected points.**

I. INTRODUCTION

The popularity and usage of point cloud data is growing exponentially. Governmental institutions and companies are collecting large point clouds of the surfaces of countries and cities. One of the tools used is Potree [1] which is a WebGL based viewer for large point clouds. It is an open source project to allow anyone to view, analyze and publicly share their large data. The implementation is based on standard web technologies and requires no additional plugins to be installed. Potree works on any of the latest generation browser both of desktop pc and mobile side. It doesn't work by loading the whole point cloud , but only up to a certain detail level. The potree file format partitions points into an octree. All octree nodes, intermediate as well as leaves, contain a sparse subsample of points. The spacing defines the minimum distance between points in the root node. With each level, the spacing is halved. Rendering lower level nodes results in a coarse representation of the pointcloud [2]. The more nodes you load and render, the better the quality. Potree also provides the following features:

- Different Materials (RGB, Intensity, Classification,);
- 4 Point Rendering Modes (Squares, Circles, Interpolation, Splats);
- PotreeConverter creates BINARY, LAZ(compressed) or LAS hierarchy;
- Based on three.js so everything three.js can do, too;
- Distance, Area and Height Profile Measurements;
- Fixed, Attenuated and Adaptive Point Sizes;
- Georeferencing.

In addition , we have extended Potree functionality , using Google APIs to create a map where we visualize the places where there are clouds of points.

II. INITIALIZATION PROCEDURE

A. Point cloud conversion

To carry out the project we used the two point clouds in ".pts" format, also we are instructed to relieve the potree program from github to see the clouds of points. We converted the two files through potreeConverter [3] program. To start the conversion, it need to open the terminal in the folder where PotreeConverter.exe is and run the following command: PotreeConverter.exe "path where is located the pts file to convert" -o "path destination of converted files" -p pageName. Where pageName -p must be specified to generate the pagename.html files on the display of the point cloud via a browser (in our case we have renamed it as index.php). At the end of the conversion process for each point cloud it is generated pagename.html files and libs and pointclouds folders. To view the point cloud that we converted, several steps are required: The pagename.html file must be copied into the examples directory of Potree. The libraries contained inside the libs folder created must be copied into the libs folder of Potree program. The folder in pointclouds must be copied to the resources / pointclouds folder Potree and it mainly contains the following files:

- A date folder: This directory contains node and hierarchy files.
- A cloud.js files: A JSON file that contains meta data such as bounding box, spacing, etc.

III. MATERIALS AND METHODS

A. *Home-Page*

In the home page, through maps.php file, it is possible to see the Google map, containing all markers with relative boundingBox associated with clouds of points, and a dynamic list on the right displays all the point clouds present. The markers and their boundigBox (also the list) are loaded and generated by reading data from dati.json file where the information is stored on latitude (lat), longitude (lng), descr (description we will use to pass it as a value) and north, south, east and west for boundingBox.

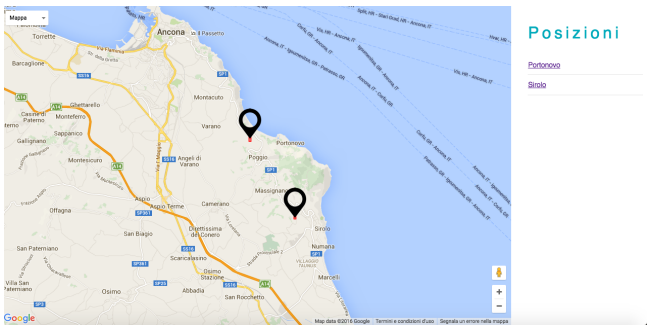

Fig. 1. Home-page: maps.php

For each marker and each element of the list, the action of clicking on them is associated with a link that allows the passage of the values to the index.php file. The values we provide are: descr, lat, and lng. These are taken up in the index.php file and stored in the respective variable $var, $lat, $lng. Within the Potree, in resources/pointclouds folder must be saved to a folder with the same name as the ”descr” value, which contains precisely the cloud.js file.


```
[
{"lat":43.563106,"lng":13.572670,"descr":"Portonovo","north":43.5645932,"south":
    43.5620112,"east":13.5733472,"west":13.5720175},
{"lat":43.524503,"lng":13.603263,"descr":"Sirolo","north":43.5250532,"south":
    43.5240012,"east":13.6039635,"west":13.6026175}
]
```


Fig. 2. dati.json

In this way we are going to generate the index.php page, which allows viewing of the point cloud, which changes dynamically based on parameters that are passed.(Fig. 3)

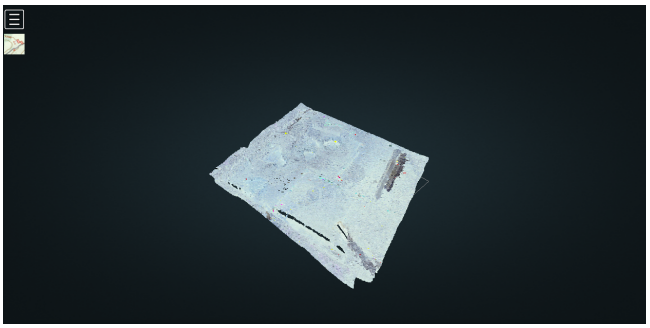

Fig. 3. index.php - Portonovo

B. *Functionality Potree*

Through the button at the top left, you can get a menu with the functionality of Potree. Generally, they are: Appearance, Navigation, Tools, Measurements, Material, Scene, Classification filter and Settings. Below we give a description of the main features:

Fly-Navigation: Rotate camera.

Orbit-Navigation: Rotate camera around pivot and Pan camera & pivot.

Zoom to Point Cloud: Moves camera so that the whole point cloud is within the view area.

Flip Y and Z: Flips y and -z coordinates.

Distance Measurement: Distance between measurement points will be displayed on the edge.

Area Measurement: Distance between measurement points will be displayed on the edge and the total area on the centroid. The area is calculated on the ground plane. The height does not affect the result. Crossings should be avoided.

Volume Measurement: Volume will be displayed on the objects centroid.

Height Profile: Depending on the Clip Mode, points inside the profile will be highlighted or points outside the profile will be clipped.

Clip Volume: Depending on the Clip Mode, points inside the volume will be highlighted or points outside the volume will be clipped.

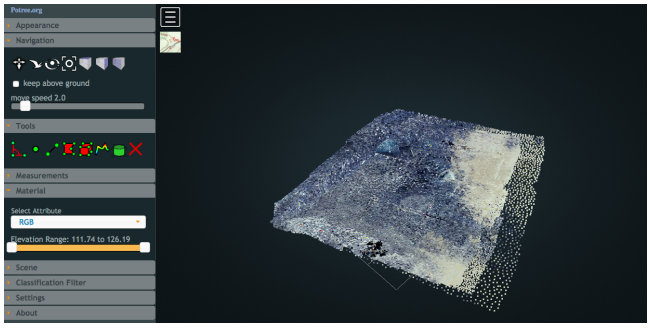

Fig. 4. Functionality Potree

C. Localization

Within the application we have included a button to display the Google map that allows you to see in it the markers and their boundingBox in relative positions of point clouds. Again taking advantage of the shift in values and in particular the $lat and $lng variable is centered the Google map on the point cloud selected and markers and boundingBox on the point cloud being viewed is indicated in red, while the other markers present are marked in black with its boundigBox in blue.

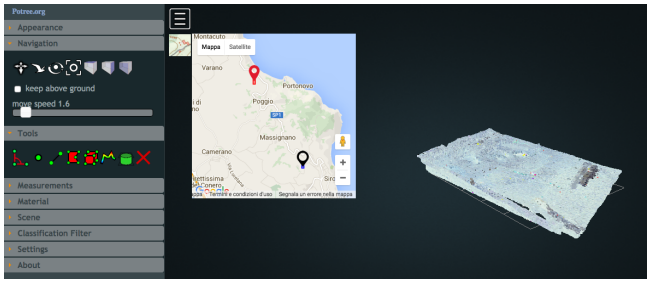

Fig. 5. Dynamic localization of the point cloud

The boundingBox represents the area of the point cloud within the map. To build it we used the coordinates inside the cloud.js file referring to the cloud of points of the boundingBox you want to get. The representation of the area on the map was realized through the Google APIs[4], in particular with a default object called Rectangle of which you can set different parameters. And necessary to transform coordinates on cloud.js in WGS-84, to do this we used www.epsg.io[5] as shown in the Table 1.

These are the transformed coordinates representing the North for ”ux” and the east for ”uy”. To paint the area of the field on the BoundingBox, we

TABLE I

COORDINATE CONVERSION

	UX	UY
Cloud.js	1510978.104	5398307.379
epsg.io	13.5733472	43.5645932

modified the WGS-84 values in the order of three decimal places to represent more accurately the indicated area. The final values of the BoundingBox to the clouds of Portonovo and Sirolo points are shown in Table 2.

TABLE II

COORDINATE BOUNDINGBOX

	North	South	East	West
Portonovo	43.5645932	43.5620112	13.5733472	13.5720175
Sirolo	43.5250532	43.5240012	13.6039635	13.6026175

D. Annotation

We have also added the possibility to view on the cloud of points of information which may be of three types :

- Text
- Picture
- Link

This type of information is contained in annotazioni.json (Fig. 6) file where you need to specify the mode array of arrays:

- Id (which in our case, for example, is Portonovo and must be equal to descr field of dati.json file) that will allow us a dynamic access to records relating to the cloud of points to be displayed.
- In the field of knots:
  - x, y, z: indicate the point where the tag will be displayed;
  - a, b, c: direction that looks at the camera;
  - d, e, f: indicate the display move to the click of the tag;
  - title: is text displayed on the tag;
  - description: where you can enter the type of tag you want (text, image, link).

IV. CONCLUSIONS

The application is presented to the end user as in the figures below (Fig. 7/8) relating to clouds

```
[{"id":"Portonovo","nodi":[
                {"x":0,"y":2,"z":0,"a":0,"b":2,"c":0,"d":7.34,"e":3.24,"f":
                    6.35,"title":"Immagine","description":"<br><center><img
                    src='http://www.rivieradelconero.info/media/widgetkit/spiaggia-
                    portonovo_001-8a6969387759b95f878c3f29bd5b97e5.jpg'
                    width='200px'/></center>"},
                {"x":5,"y":2,"z":0,"a":0,"b":2,"c":0,"d":7.34,"e":3.24,"f":
                    6.35,"title":"Portonovo","description":"Collina"},
                {"x":0,"y":2,"z":5,"a":0,"b":2,"c":0,"d":7.34,"e":3.24,"f":
                    6.35,"title":"Link","description":"Collina di Portonovo (<a
                    href='https://it.wikipedia.org/wiki/Portonovo'
                    target='_blank'>Wikipedia</a>)"}]},
 {"id":"Sirolo","nodi":[
                {"x":0,"y":2,"z":0,"a":0,"b":2,"c":0,"d":7.34,"e":3.24,"f":
                    6.35,"title":"Immagine","description":"<br><center><img src='http://
                    www.turismosirolo.it/images/gallery-home/sirolo10.jpg' width='200px'/
                    ></center>"},
                {"x":5,"y":2,"z":0,"a":0,"b":2,"c":0,"d":7.34,"e":3.24,"f":
                    6.35,"title":"Sirolo","description":"Spiaggia"},
                {"x":0,"y":2,"z":5,"a":0,"b":2,"c":0,"d":7.34,"e":3.24,"f":
                    6.35,"title":"Link","description":"Sirolo  (<a href='https://
                    it.wikipedia.org/wiki/Sirolo' target='_blank'>Wikipedia</a>)"}]}]
```


Fig. 6. annotazioni.json

of points Portonovo and Sirolo provided. It offers the opportunity to add in a simple and dynamic files through the .json further clouds of points and display them, so as to make the most complete application possible. Thanks to Potree program is possible in a few steps displaying large files in a simple and fast way. The project started in education, but in the future you can improve it by adding new features with a more detailed study of Potree. One of the improvements that can be made is to enhance the interactive end-user experience.

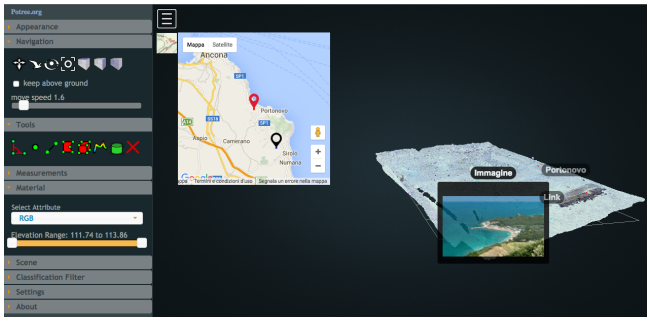

Fig. 7. index.php - Portonovo

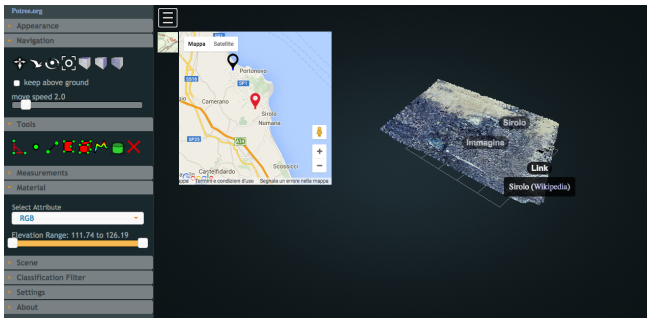

Fig. 8. index.php - Sirolo

REFERENCES

[1] Potree - www.Potree.org

[2] Pointcloud - Documentation: www.pointcloud.org/documentation

[3] PotreeConverter - https://github.com/potree/PotreeConverter/releases

[4] Google Maps - https://developers.google.com/maps/

[5] www.epsg.io - Find a coordinate system and get position on a map.