Point Cloud Visualizer (PCV) Documentation

Project holder: YouLiang, Kan, Max

1. Introduction

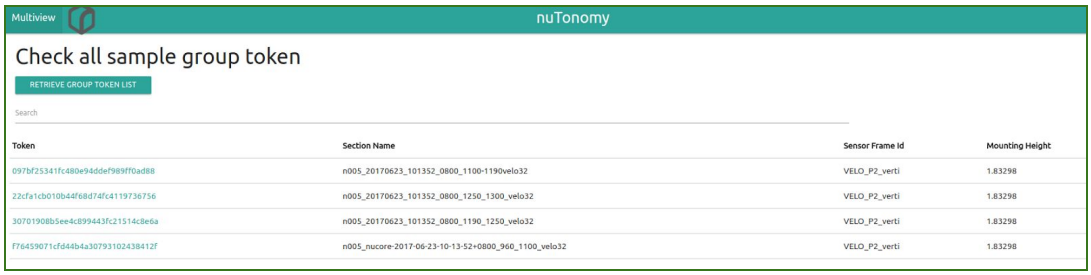
PCV is a web-based point cloud visualizer for the usage of the perception team. The objective of having this tool is to enable the easy access and visualize of the vehicle generated lidar point cloud. Besides, the new database structure and organization will help in providing a much organize database storing structure and update mechanism of all existing tracklets.

2. Project Structure

This project structure consists of three main components, which is Home, Annotation, and Multiview. The function of each will be further elaborated here as below:

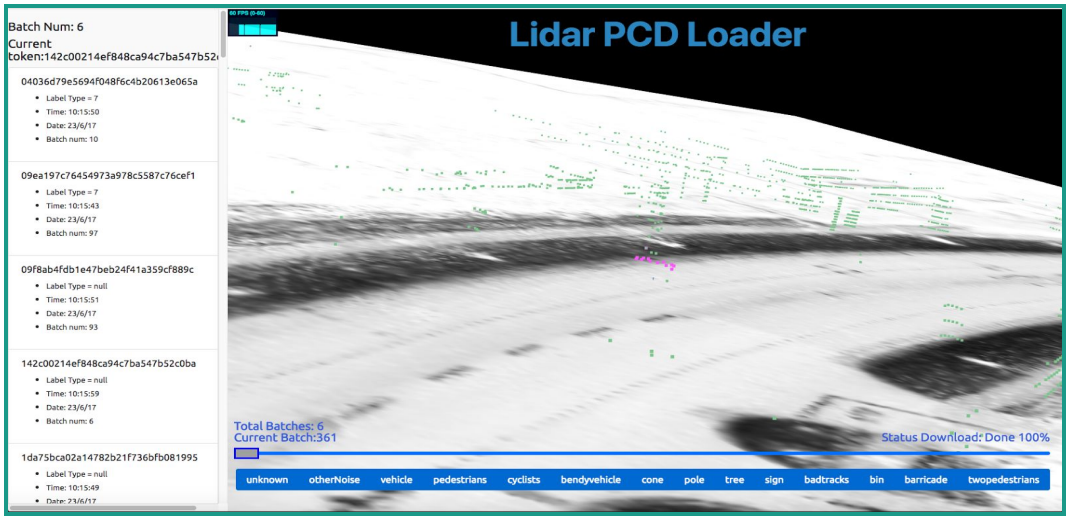
1) Home

Display a list of all generated pc tracklet from the log. User can click the top right side of the "multiview" to access multiview page, or click one of the listed "group_token" to access the annotation page.



2) Annotation

Annotation tool for data labeller to label tracklets' type, also for overall visualisation of individual tracklet.



3) *Multiview*

Multi Scenes pc visualizer of selected tracklets according to user's input criteria

3. Setup the environment

➢ Path of project: *sandbox/visualizer*

Installation

➢ Install NodeJs

- Nodejs or
- sudo apt-get install -y nodejs

➢ Install npm package depencies
- npm install

➢ Install mysql
- sudo apt-get install mysql-server
- Access by: ./mysql -u root -p

Running of nodejs

cd $visualizer/

➢ Update database:
- node callend.js (to add things inside database)
- node addSample.js
- node addLabelType.js

➢ Run PCV

Run: *node server.js*

On this stage, server will be running successfully. To access the PCV tool, user is required to enter localhost:8080 to access the page on a browser. Good Luck!

4. Folder and Files Structure

Directory: $visualizer/

Files	Description
callend.js	Request all the group tokens from the server storage and add it into the mysql database

addSample.js	Request all the Samples from the server storage and add it to the mysql database
data.sql	Create Tables and rows
server.js	Run server.js to start the server on port 8080

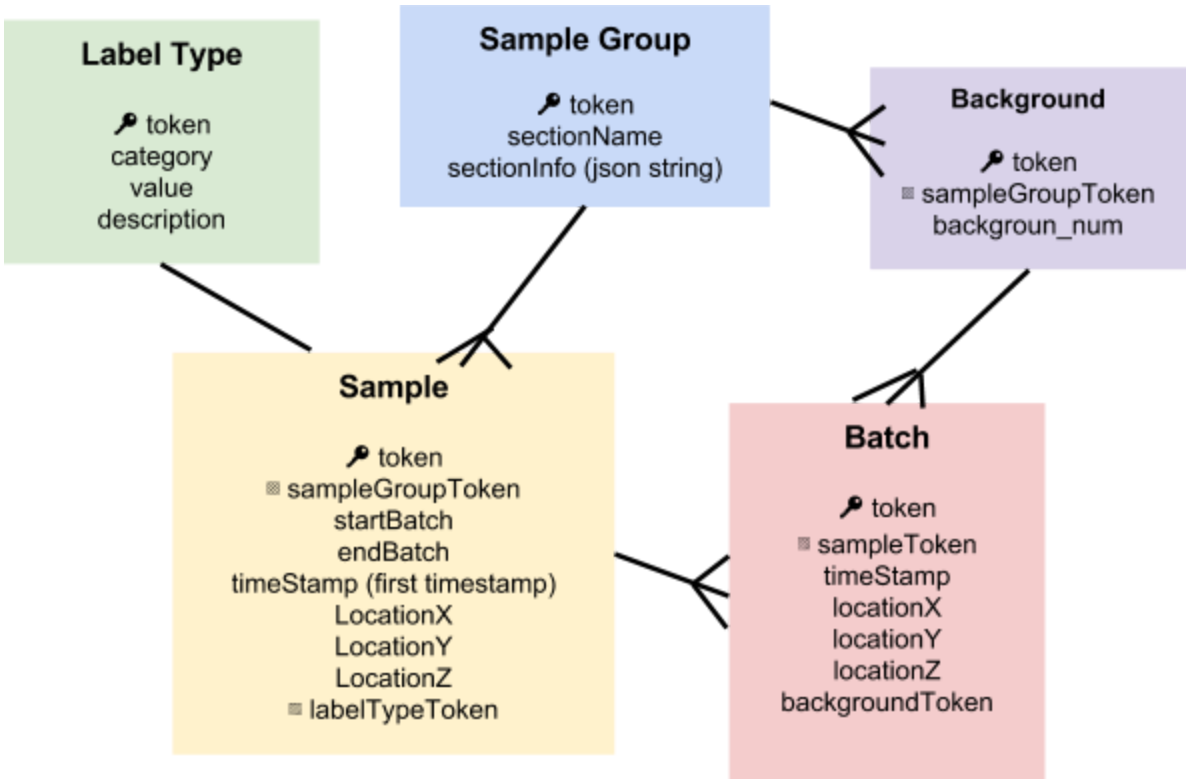
Directory: $visualizer/view

Views Files	Description
homePage.jade	Homepage that display all the tables of the group token
sideAnno.jade	Sidebar of the /annotation page
annotation.html	It is the rendering on the /annotation page which is displayed using <iframe>
sideMulti.jade	Side bar of the /multiview page it includes time filter and type filter and the result after filter.
multiView.html	It renders the multiple boxes of the rendering in the /multiview page
/resource folder	It stores all the images that will be use on the website

Directory: $visualizer/tarfiles

File/folder	Description
"*Track_Token_number*"	Folder of "Token" consists of the tarfile of and pcd file of the corresponding sample. Each folder consists of: 1) background_X.pcd: The background pcd of the whole scene during that timeframe. 2) Track_info: info of the specific tracklet 3) track/batchX_st.pcd: Pcd of the sample(object of interest) in that specific timeframe. **X represents the batch number*
track.json	Needed for the reference of /Multiview. The page will render the scene stated in the json generated from server.js after user select his selection.

5. DataBase Model



6. Nodejs Structure

End points in **Server.js:**

GET:

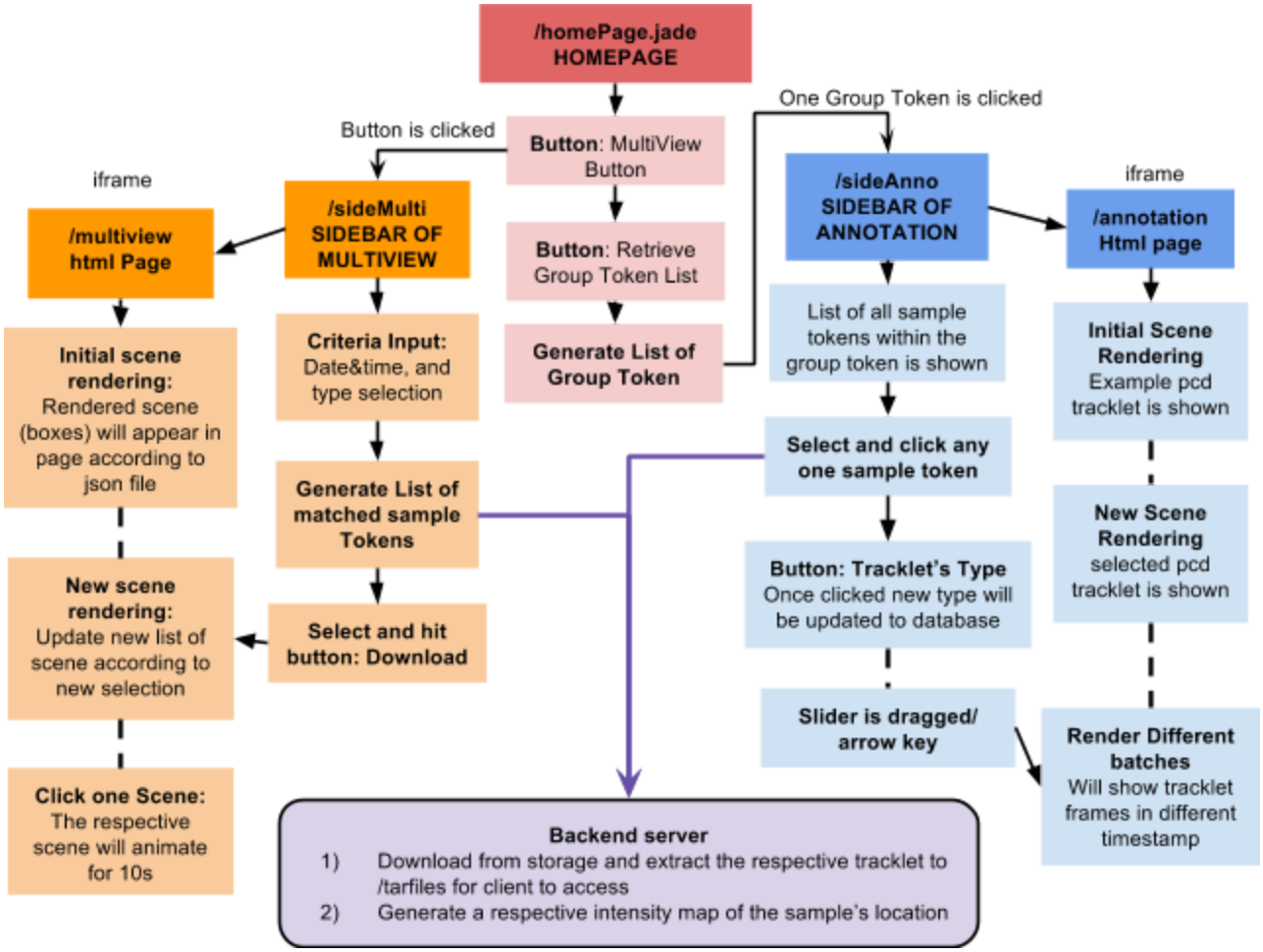
ROUTING	DESCRIPTION
'/'	Home Page
'/annotation/data'	Get all the sample tokens to the sidebar of annotation page
'/multiview'	Multi visualizer page
'/btnanno'	show all label types (eg. pedestrian, cones...) as annotation boxes on client's annotation page
'/samplesToken'	send all samples token info to the annotation page sidebar

Post:

'/annotation'	Annotation main page
'/reTable'	Refresh new group token from the storage and add it inside database
'/reSample'	Refresh Sample tokens from the storage and add it into database
'/data'	show the token on the home page (all sample groups)
'/postMultiJsonToken'	According to selected samples in sidebar, generate json file of all samples info for multiview
'/filterMulti'	Return all label type value of samples to multiView for the criteria filtering
'/samples'	get all samples info of the correspond sample group token once it's clicked from home
'/labeling'	send type status of the clicked token on client side to show 'type' on annotation boxes below

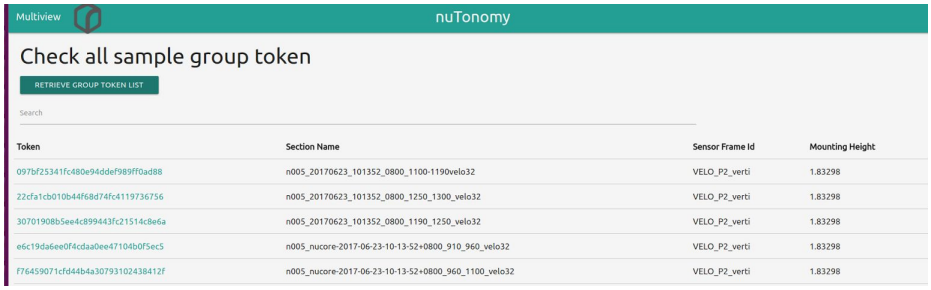
7. Process Flow

Block Diagram

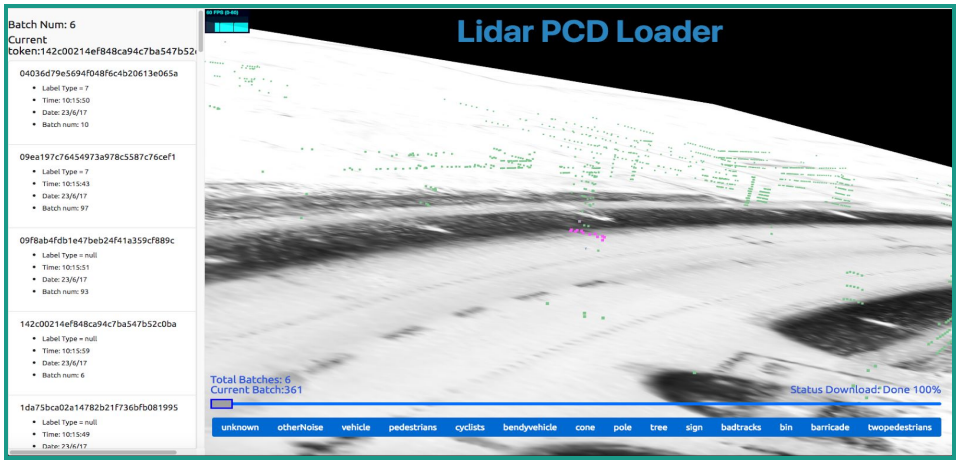


User Manual

1. **Click on the retrieve button to display all the tokens**



2. Click on one of the token to bring to annotation page



3. Click one of the Samples on the side bar and it will download and render on the page

- To annotate click the button that under the sliders to select the type of the object

4. To go to multiView page by clicking the multiview tab at the Home Page

5. Select one of the type and select time frame to filter all of the sample

6. Tick checkall or select one by one and then click on Download button to download all the tracklets from server and render in multiple boxes

8. Limitation

Multi View:

1) Limited to 30 scenes per page, when number increases performance will drop significantly.
2) Intensity map for each scene is static, means if object if moving during animation, the moving frame location doesn't reflect the actual coordinate of the object on the map
3) Animation is rendering is not smooth as silk

After considering the limitation of the multiview tool, an alternate solution to display all point cloud is needed to ensure performance is not compromised along the day. Thus we would prefer to use webgl to generate a video or gif format pic, then use qt to display to corresponding token's point cloud data.

---------- **end of document** ----------