

Point Cloud Visualizer (PCV) Documentation

Project holder: YouLiang, Kan, Max

1. Introduction

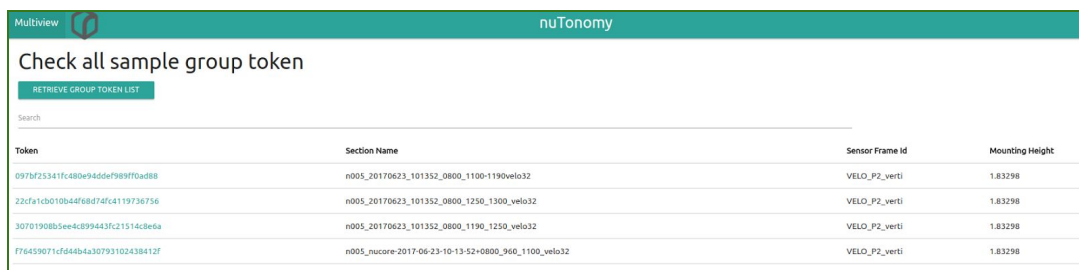
PCV is a web-based point cloud visualizer for the usage of the perception team. The objective of having this tool is to enable the easy access and visualize of the vehicle generated lidar point cloud. Besides, the new database structure and organization will help in providing a much organize database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

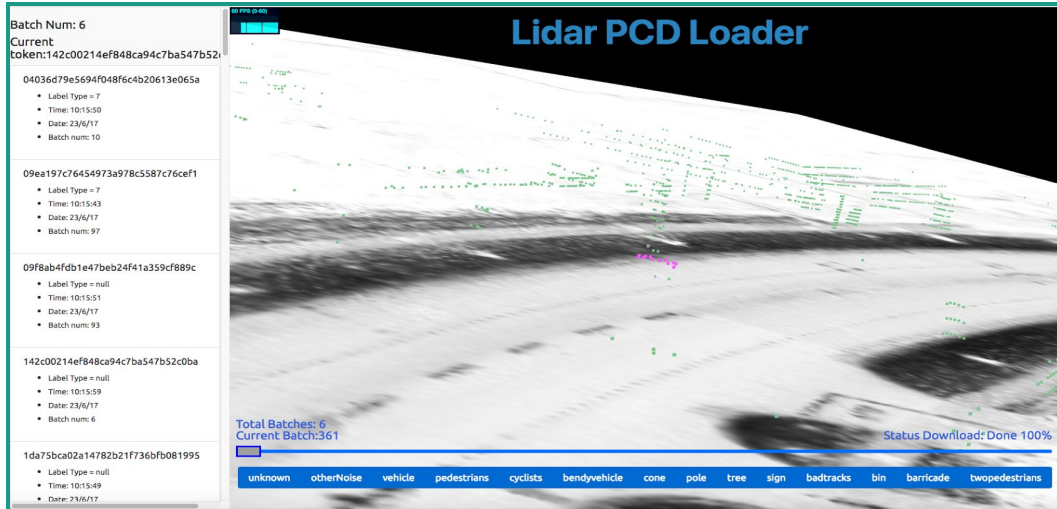


The screenshot shows the 'Multiview' web application interface. At the top, there is a teal header with the 'Multiview' logo and the 'nuTonomy' name. Below the header, there is a section titled 'Check all sample group token' with a 'RETRIEVE GROUP TOKEN LIST' button. A search bar is present below the button. The main content is a table with four columns: 'Token', 'Section Name', 'Sensor Frame Id', and 'Mounting Height'. The table contains four rows of data, each representing a different tracklet.

Token	Section Name	Sensor Frame Id	Mounting Height
097b25341fc480e94dde9b9ff0ad88	n005_20170623_101352_0800_1100-1190velo32	VELO_P2_verti	1.83298
22cfa1cb010b44f68d74fc4119736756	n005_20170623_101352_0800_1250_1300_velo32	VELO_P2_verti	1.83298
30701908b5ee4c899443fc21514c8e6a	n005_20170623_101352_0800_1190_1250_velo32	VELO_P2_verti	1.83298
ff6459071cf644b4a30793102438412f	n005_nucore-2017-06-23-10-13-52+0800_960_1100_velo32	VELO_P2_verti	1.83298

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 dependencies
 - `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

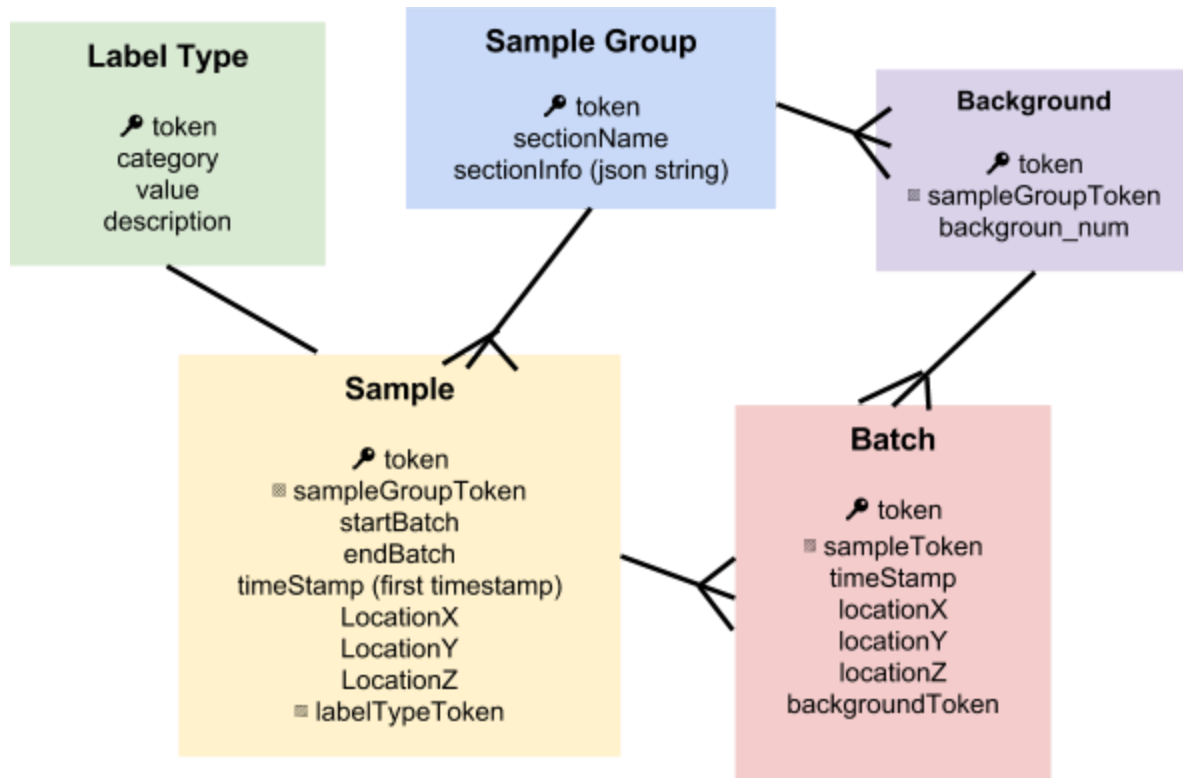
Directory: \$visualizer/view

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
<i>“Track_Token_number”</i>	<p>Folder of “Token” consists of the tarfile of and pcd file of the corresponding sample. Each folder consists of:</p> <ol style="list-style-type: none"> 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. <p><i>*X represents the batch number</i></p>
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:

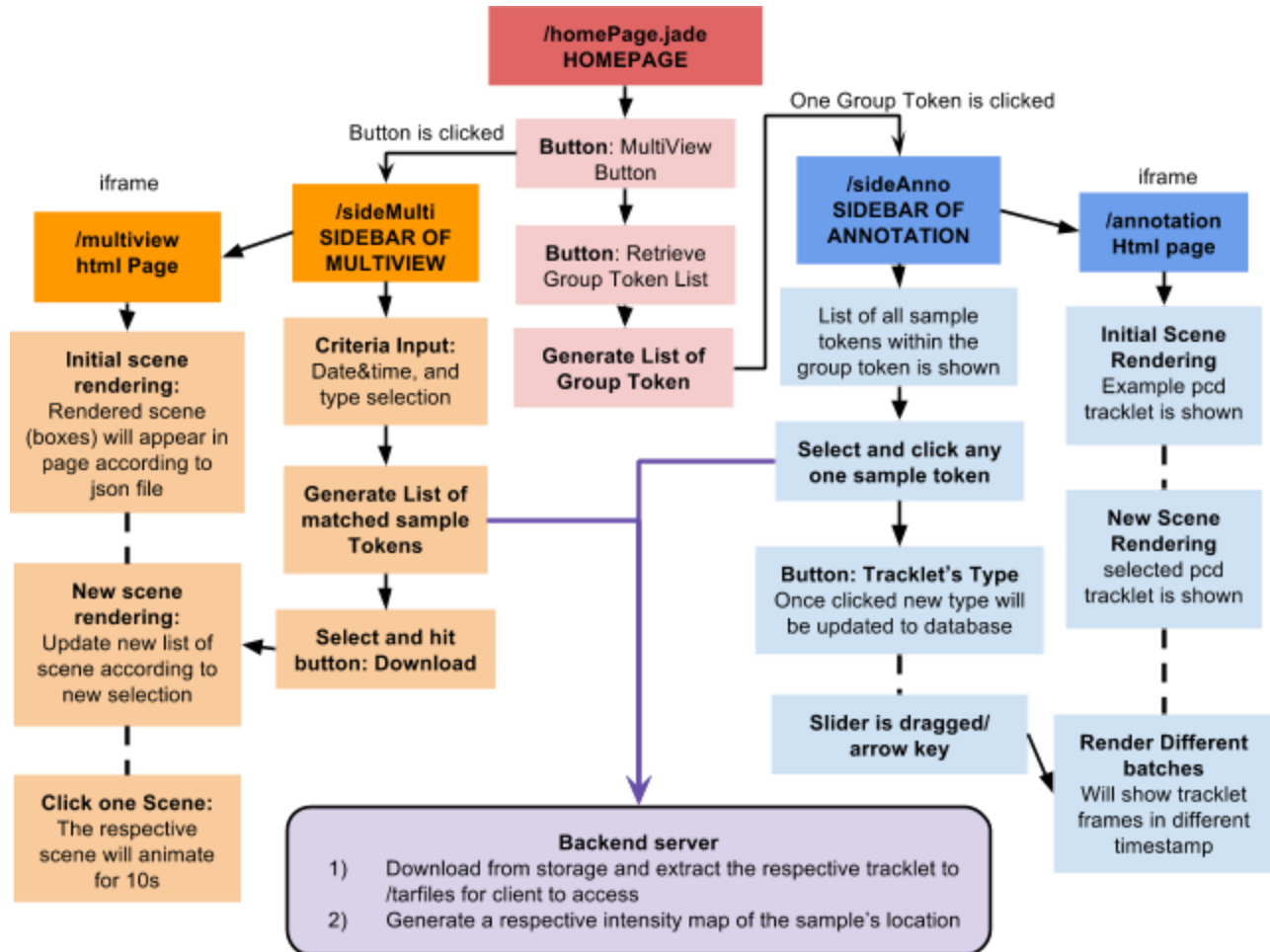
ROUTING	DESCRIPTION
'/'	Home Page
"/annotation/data"	Get all the sample tokens to the sidebar of annotation page
"/multiview"	Multi visualizer page
"/btnanno"	show all label types (eg. pedestrian, cones...) as annotation boxes on client's annotation page
"/samplesToken"	send all samples token info to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

Post:

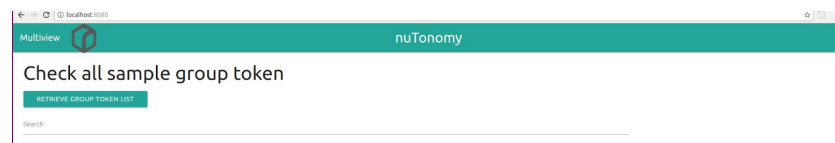
'/annotation'	Annotation main page
'/reTable'	Refresh new group token from the storage and add it inside database
'/reSample'	Refresh Sample tokens from the storage and add it into database
'/data'	show the token on the home page (all sample groups)
'/postMultiJsonToken'	According to selected samples 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

nuTonomy			
Check all sample group token			
RETRIEVE GROUP TOKEN LIST			
Search			
Token	Section Name	Sensor Frame Id	Mounting Height
097b25341fc480e94dde989f90ad88	n005_20170623_101352_0800_1100-1190velo32	VELO_P2_verti	1.83298
22cfa1cb010b44f68d74fc4119736756	n005_20170623_101352_0800_1250_1300_velo32	VELO_P2_verti	1.83298
30701908b5ee4c899433fc21514c8efa	n005_20170623_101352_0800_1190_1250_velo32	VELO_P2_verti	1.83298
efc19da6ee0fcdca0ee47104b0f5ec5	n005_nucore-2017-06-23-10-13-52+0800_910_960_velo32	VELO_P2_verti	1.83298
7f6459071cfd44b4a30793102438412f	n005_nucore-2017-06-23-10-13-52+0800_960_1100_velo32	VELO_P2_verti	1.83298

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

Batch Num: 6
Current token: 142c00214ef848ca94c7ba547b52

04036d79e5694f0d48f6c4d20613e065a

- Label Type = 7
- Time: 10:15:50
- Date: 23/6/17
- Batch num: 10

09ea197c76454973a978c5587c76cef1

- Label Type = 7
- Time: 10:15:43
- Date: 23/6/17
- Batch num: 97

09f8ab4fdb1e47beb24f41a359cf889c

- Label Type = null
- Time: 10:15:51
- Date: 23/6/17
- Batch num: 93

142c00214ef848ca94c7ba547b52c0ba

- Label Type = null
- Time: 10:15:59
- Date: 23/6/17
- Batch num: 6

1da75bca02a14782b2f736bf081995

- Label Type = null
- Time: 10:15:49
- Date: 23/6/17

Lidar PCD Loader

Total Batches: 6
Current Batch: 361

Status Download: Done 100%

unknown otherNoise vehicle pedestrians cyclists bendyvehicle cone pole tree sign badtracks bin barricade twopedestrians

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

DOWNLOAD

Filter

Date: mm/dd/yyyy

Start Time: --:--:--

Stop Time: --:--:--

End Time: --:--:--

☐ unknown
☐ other noise
☒ vehicle
☐ pedestrian
☐ cyclist
☐ bendyvehicle
☐ cone
☐ pole
☐ tree
☐ sign
☐ badtracks
☐ bin
☐ barricade
☐ twopedestrians

CHECK ALL

☐ 01309a49b7004d23a18af3778d8e68a
148818235

☐ 013384797709465975a2f5a622980d
148818276

☐ 3a7370c9419b4f096a98b0f53aea444
148818255

☐ 0a77f6ec2ac81455fa2ac156da4ccdb8a
148818270

☐ 08417969176d459cb2983d9537883cf
148818275

☐ 08946654794039c18081d7457307a
148818256

PointCloud MultiView Visualizer

Scene: 013d9b49b7004d23a18af3778d8e68a

Scene: 0153b87977594cf4975a2fc54c22980d

Scene: 04f3969419b4fb096a98b0f53aea444

Scene: 06716ec2ac81455fa2ac156da4ccdb8a

Scene: 08417969176d459cb2983d9537883cf

Scene: 08946654794039c18081d7457307a

Scene: 095a70a4166c40b88594473b401779b5

Scene: 0e33e991728c4055b2fab1128290b6ef

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 is moving during animation, the moving frame location doesn't reflect the actual coordinate of the object on the map
- 3) Animation 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 way. Thus we would prefer to use WebGL to generate a video or gif format pic, then use Qt to display the corresponding token's point cloud data.

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