# 3D Object Annotation Tool for Point Cloud Data

**Problem Statement**

There is a lack in the market of an open source tool capable of annotating 3D point clouds. Point cloud annotations serve as ground truth for many applications, including self-driving technology. This project serves to fill this gap.

**Market Survey**

Point cloud annotation tools are generally either proprietary, or if open-source, are intended for applications other than the self-driving realm, such as geomodelling.

A variety of for-profit applications exist for performing annotations; many of these services also offer in-house annotation ability of provided data. One such popular option is Playment [1], based in India.

Another interesting solution is provided out of the Stanford’s Computation Vision and Geometry Lab. Their MATLAB-based tool involves orienting a Google Warehouse model to match the object orientation in the camera image, and then clicking through a series of point correspondences between the model and the object in the image.

**Technology Stack**

Our application uses the below technologies:

* Lidar Data ( [Velodyne HDL-64E](http://wiki.ros.org/velodyne/Tutorials/Getting%20Started%20with%20the%20HDL-64E) ) {Compatible with any point cloud data also(.ply, .pcd)}
* Python 2.7.x (ROS), Python3.5.x (Application), C++ (Processing)
* ROS Kinetic
* OS Support (Windows, Ubuntu)
* Python Libraries (Open3D, PySimpleGUI, Numpy, OS, TensorFlow, MatPlotLib)

**Methodology**

The easiest and generally quickest way to run 3D annotation is to provide the user with options and to let them choose which ones they wish to keep and the ones they choose to discard.

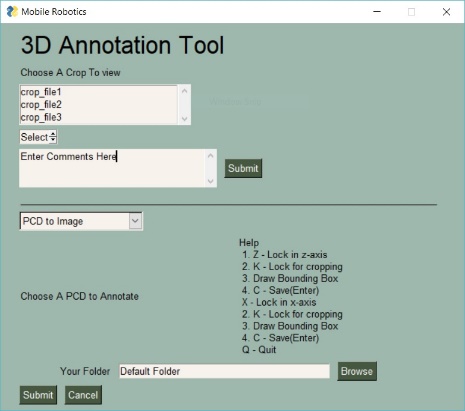
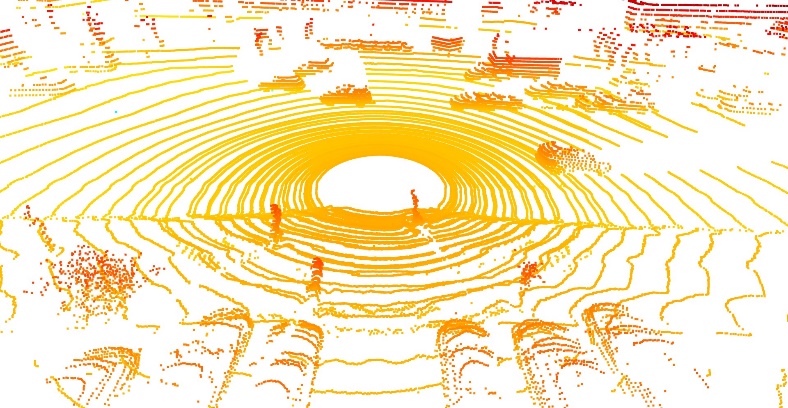
Providing a dual option (Point-cloud to Image and vice versa) in helps in better visualization from the annotator’s perspective allowing them to work with whichever frame they are comfortable with or is more intuitive.

**Steps**

**Progress Made**

Our System is capable of annotating and translating the 3D points from both Pointcloud to image and vice versa and stores these in a JSON file for pcd data and text file for the respective imaging data.

GUI allows for easy access to the data and file handling allowing rookie users to easily utilize the resources.

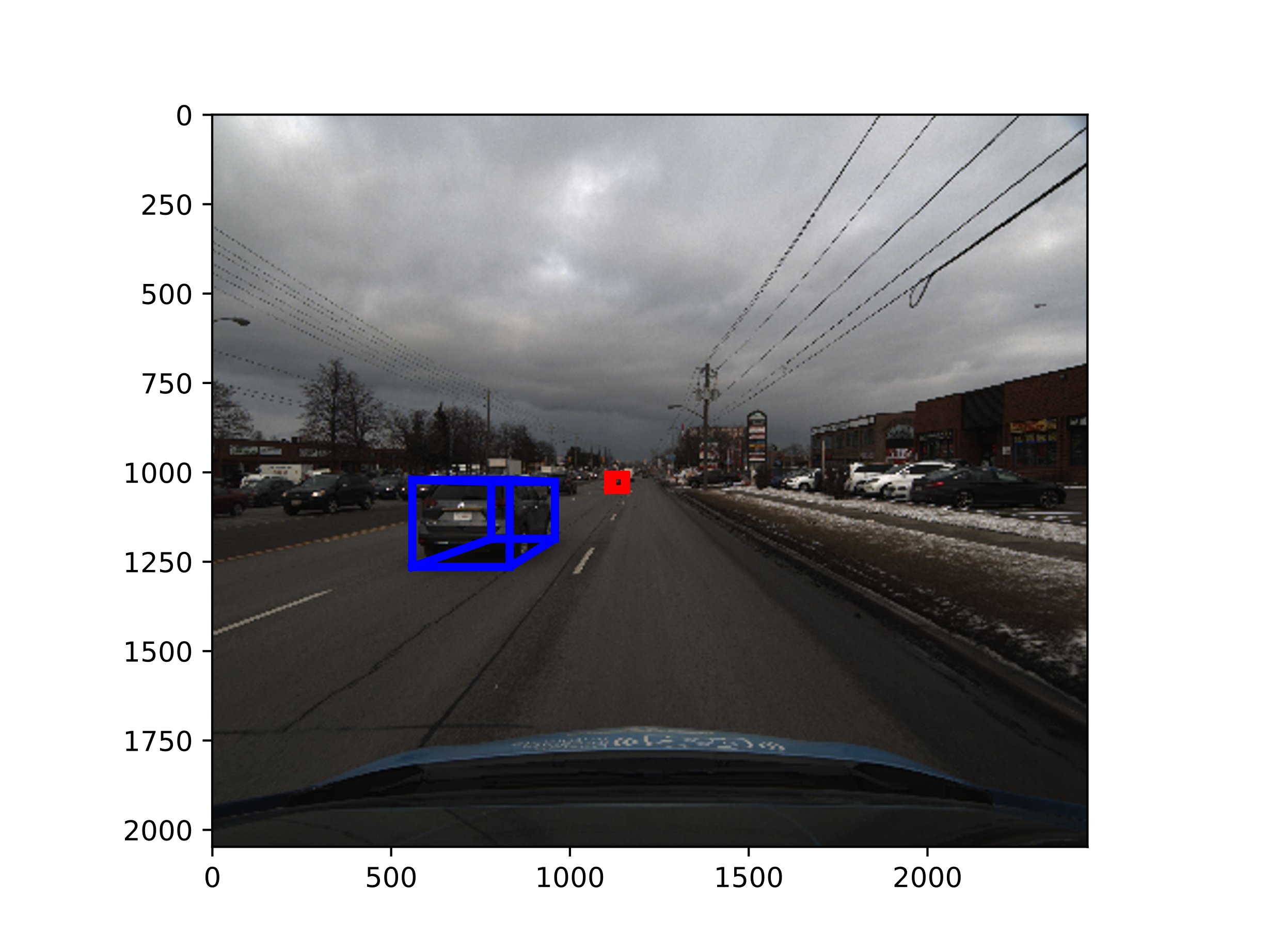
**Conclusion**

The automatic cropping option almost halves the time of annotating the frames.

The tool is OpenSource and under constant development available at

Primarily for vehicle detection but can be used for any 3D data set pairs or can be modified to only annotate in pointcloud or image data in case missing on of the pair.

**References**

**[1]**

**[2]**