Jon Rippe  
PDAQS  
Progress Report 21.10.0

# Issues Addressed

* LiDAR/Camera Alignment
* Configuration Storage
* Development OS change
* GPU Integration
* PointNet Neural Network (NN)

## GPU Integration

Attempts to create a GPU enabled development environment within Windows were fruitless. Visual Studio failed to build OpenCV with CUDA support. I have transitioned to a Linux development environment. As James has been spearheading the video object detection tool, I will test GPU integration on the PointNet NN in the near future.

# Notable Progress

Camera distortion parameters were added to the LiDAR point transformations. There are four customizable distortion parameters , , , and such that

Additionally, a rudimentary tool was developed to assist in aligning LiDAR points to camera frames. The tool allows a user to modify Camera parameters (offset, angle, distortion) and re-render point cloud data on top of camera frames. The tool also allows the user to skip forward a user determined number of frames. Figure 1 shows a screenshot of the tool.

  
*Figure 1: LiDAR/Camera alignment tool*

All project settings and parameters have been switched to json format and are stored in a single **params.json** file. This will act as a single point for saving and loading project configurations and will allow for better modularity in the future.

# New Modules

No new notable infrastructure was added to the main workflow.

# Next Steps

Building the LiDAR/Camera tool uncovered some areas of improvement in the way the cameras retrieve and store frames; more specifically, the behavior of the buffer when switching to the next video file.

Adding time offset adjustment to the LiDAR/Camera tool could be useful (allowing the user to advance either camera frames, LiDAR frames, or both), but will require development of a new way to handle frame information.

LiDAR Point NNs are not as readily available as image processing NNs. There are a lot of projects centered around the KITTI dataset for 3D object recognition, however they’re mostly designed as proof-of-concepts that only work within the scope of the KITTI Suite (<http://www.cvlibs.net/datasets/kitti/>). We’ll need to do some digging to see if cvlib for Python has a NN model suitable for our LiDAR data.