



# GPUService

## GPU Acceleration of Robotic Services: Real Time 3D Point Cloud Processing

Leonardo Milhomem Franco Christino, Dr. Fernando Santos Osório

- <http://www.lrm.icmc.usp.br/>
- <http://www.lrm.icmc.usp.br/carina/>
- <http://www.lrm.icmc.usp.br/en/?page=projetos>
- <http://www.velodynelidar.com/lidar/lidar.aspx>

### Motivation

- Three-dimensional sensors data processing applied to mobile robotics (e.g. Velodyne LIDAR)
- Usage of ROS operating systems to divide complex projects in smaller modules communicating by net packets (in this project modules are called services)
- Services: 3D point clouds pre-processing, segmentation, planar zones detection (ground, roads), and detection of elements of interest.
- Main Application: Services for a self-driving car, forcing to approach a system for real-time processing
- Equipment: Low-Powered Nvidia GPU (620 GT)

### Velodyne (HDL-32) Sensor

Omnidirectional multilayer LIDAR sensor

- 360 degrees and 32 layers
- Output: XYZIR pointcloud (XYZ coord.; I: intensity; R: ring)
- +/- 2cm Precision
- Up to 70,000 points per frame at 10Hz 100m Range



Problem: Unorganized Sparse Data Array

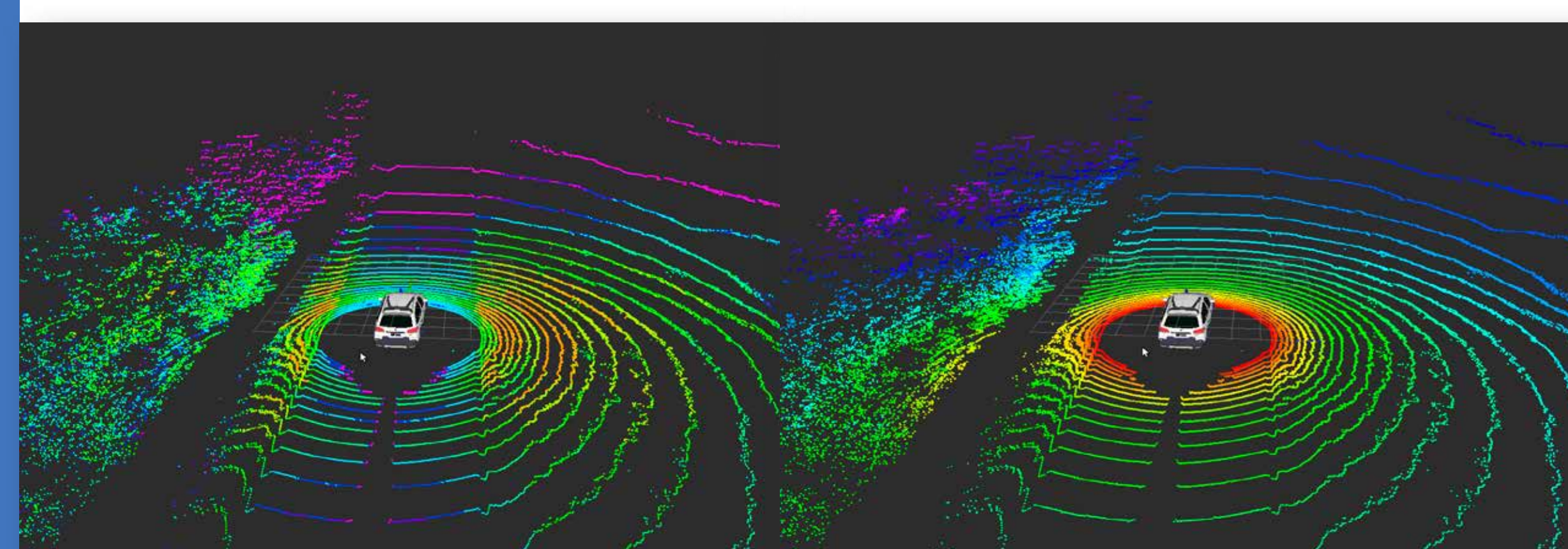
Solution: Parallel Array organizer and filler

- Fill a 32x1800 matrix (size decided by number of blocks and the sensor data sets)
- This helps hash access of other parallel services

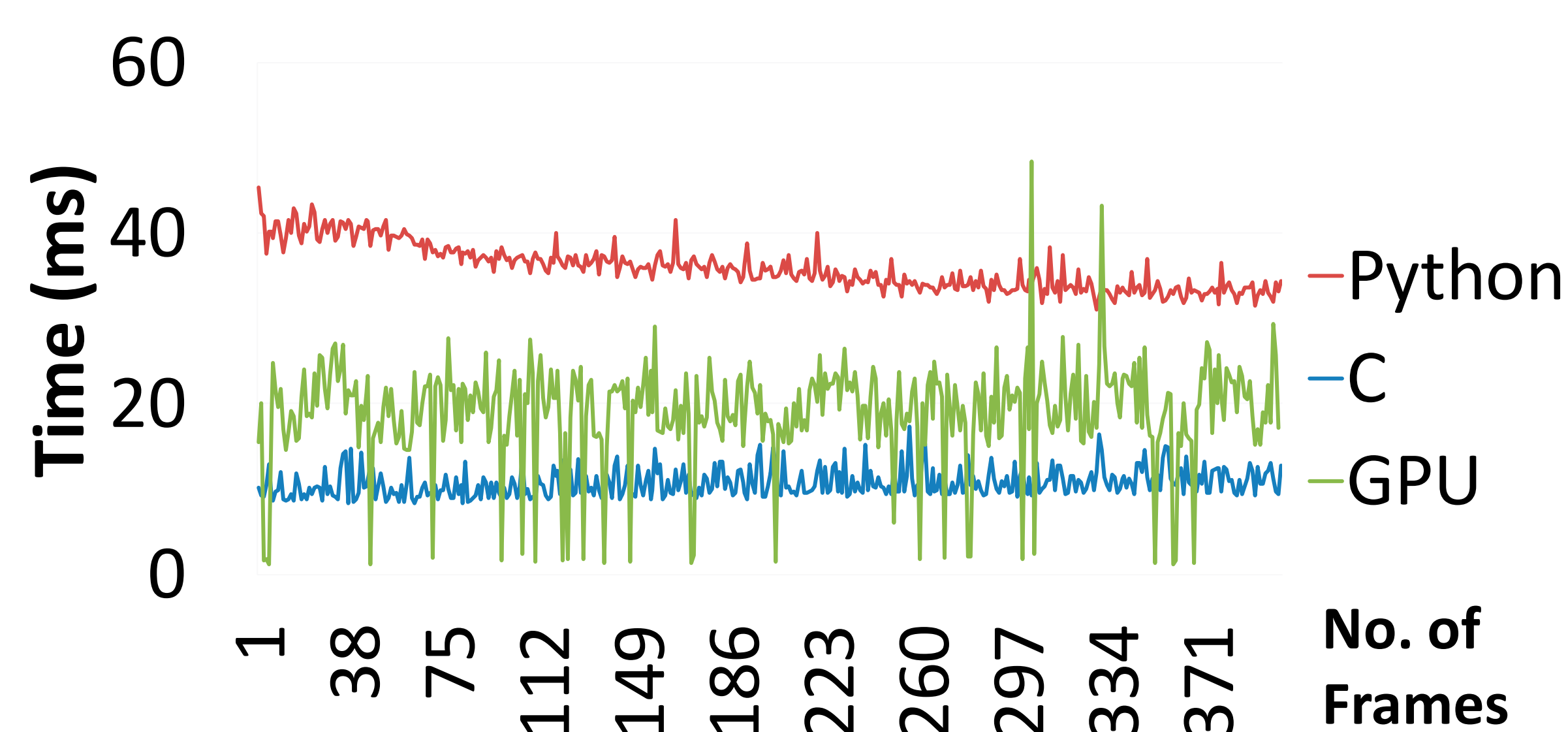
### Preprocessing

Rearrangement of Sensor data

- Input: Sparse unorganized array
- Output: ROS PointCloud2 data type Dense Matrix Structure (32x1800)



#### Rearrangement Times

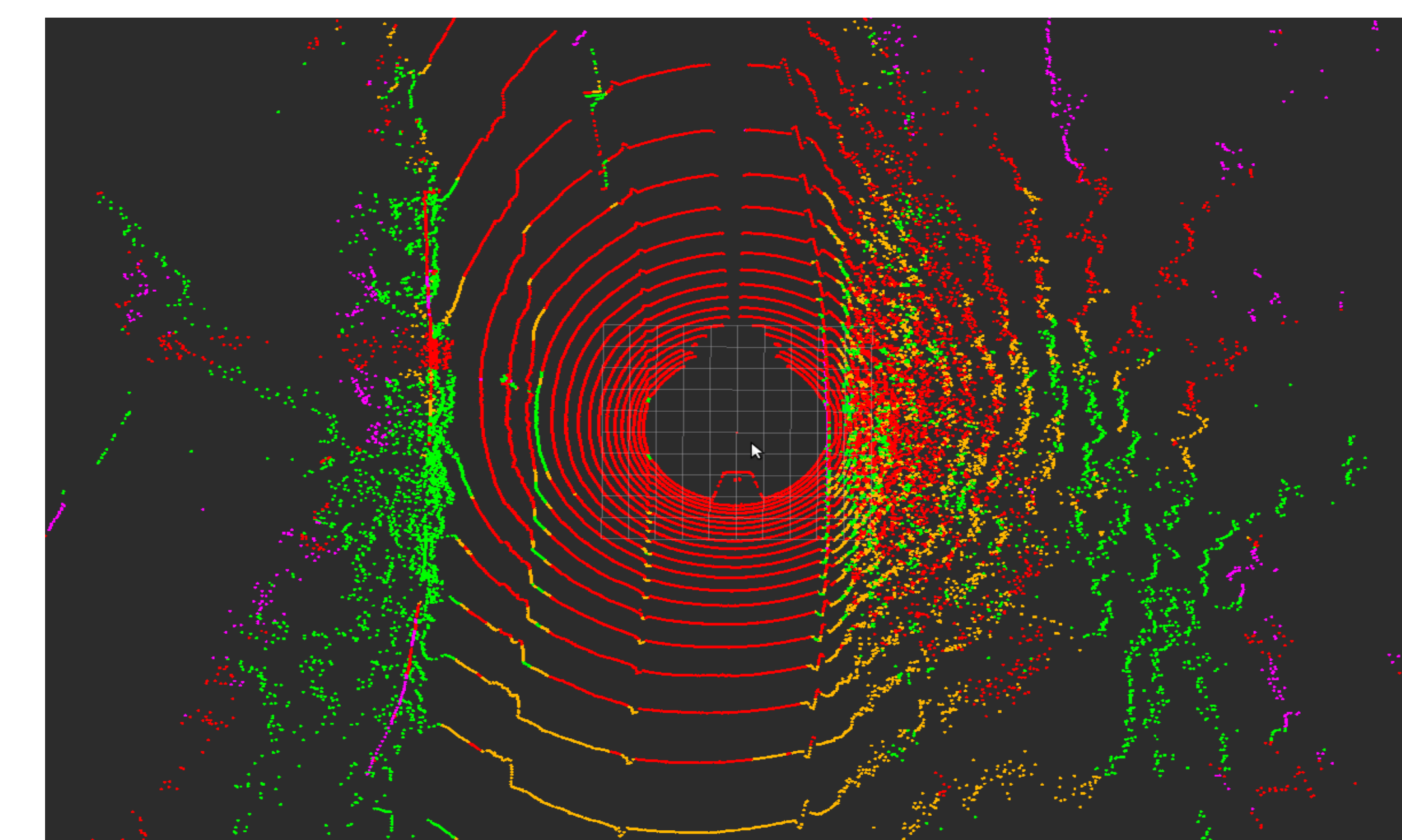


Packet Arrival Id	1	1	1	2	2	2	3	3	3	...	7	7	7	...
Real Id (y,x)	1,1	1,2	1,3	1,4	1,6	1,7	1,6	1,7	1,8	...	1,32	2,1	2,2	...

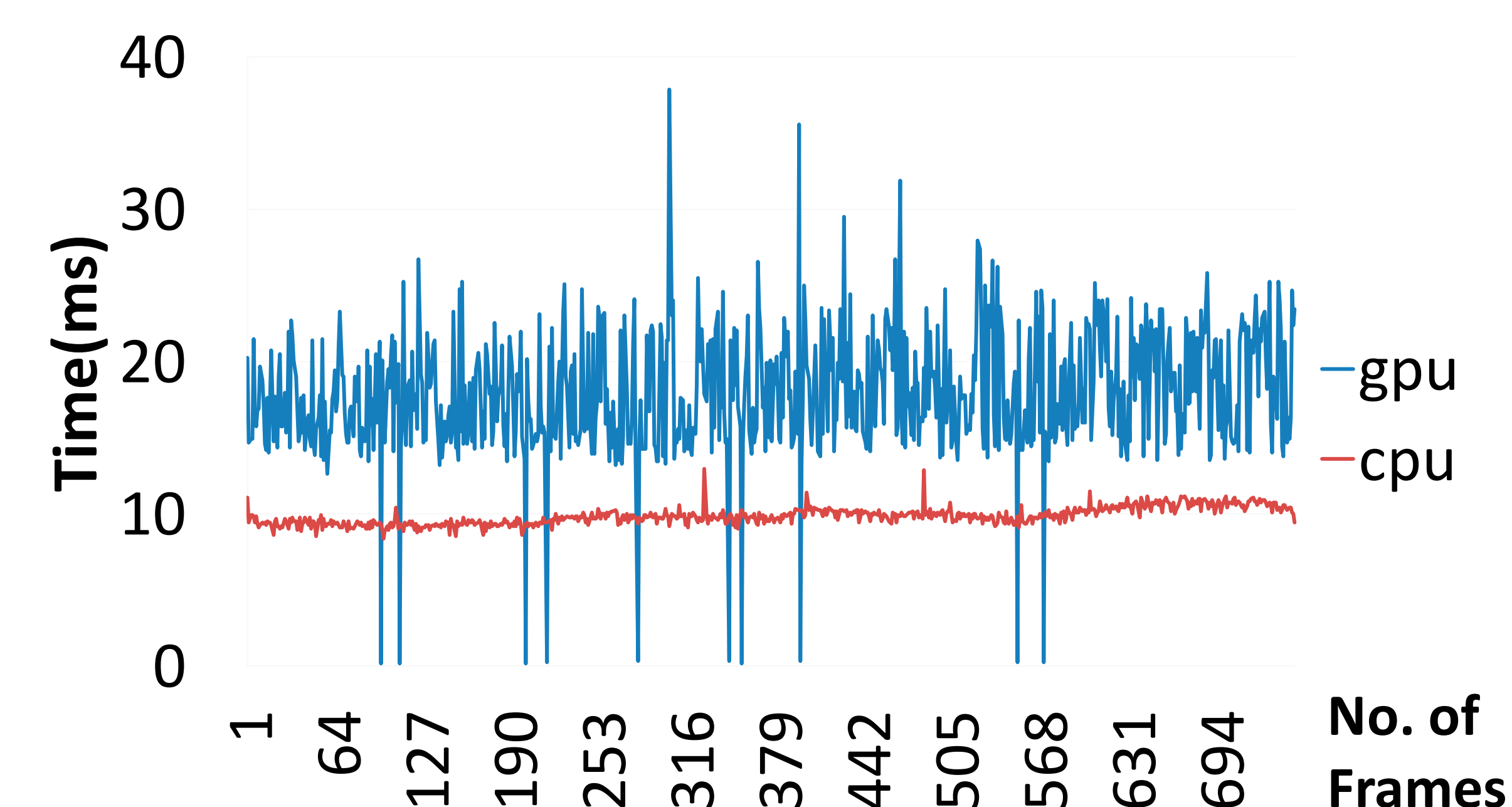
### Planar Zones Segmentation Processing

Many approaches tested. Best:

- Hip between radials/height
- Median Filter 1x5 for noise reduction
- Threshold classification



#### Planar Zones Segmentation Times



### Conclusion & Future

Different techniques tested

- More techniques to be added in future services

Throughput not compatible for one kernel per service

- By far, the memory copy between device and host is the biggest time consuming instruction
- Use a single kernel for many services in sequence

Biggest I/O blocking and data heavy services in Carina Project (Autonomous Driving Car)

- Now done in GPU, leaving CPU for other algorithms
- More GPU types to be used, such as embedded Tegra (Jetson)

