owlRobotics 3D LiDAR product finder

Depending on your specific application usage, we can assign your application to one or more of the following LiDAR scenarios:

- 3D Odometry
- Localization
- Mapping/3D scanning/ 3D measurement
- Object recognition
- Object tracking
- Obstacle detection

Product finder

This table summarizes the recommended owlRobotics LiDAR types and their typical usages. The LiDAR scenarios, recommended owlRobotics LiDARs, and recommended ROS software packages are explained in detail further below.

	long range point clouds in **fast** motion	**long range**, **dense** point clouds in **fast** motion	**dense** point clouds in **slow** motion
Recommended owlRobotics LiDAR	LS LIDAR C8/ C16	LS LIDAR C32	Livox MID 360
Max. recommended open space distance	100 m	100 m	30 m
Max. recommended sensor speed	30 m/s	30 m/s	0.5 m/s
Expectable localization accuracy	1-5 cm	1-5 cm	1-5 cm
Expectable mapping accuracy	10-20 cm	5-10 cm	1-5 cm
Min. detectable obstacle size	0.3 m (near) - 1.0 m (far)	0.3 m (near) - 0.3 m (far)	0.1 m (near) - 0.3 m (far)
Recommended usage	• Odometry • Localization • Obstacle	• Odometry • Localization • Mapping	• Odometry • Localization

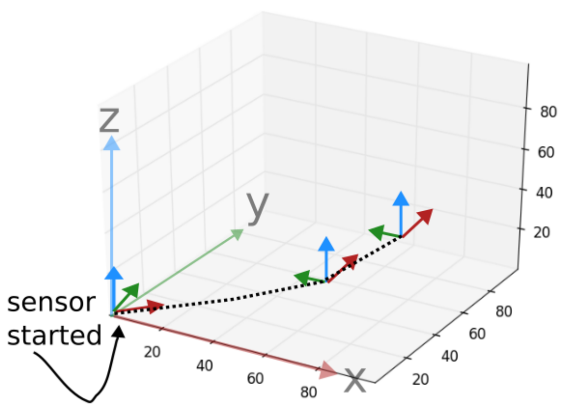
	detection ● Obstacle tracking ● Distance measurement	● Obstacle detection ● Obstacle tracking ● Distance measurement	● Mapping ● Obstacle detection ● Obstacle tracking ● Distance measure ment

3D Odometry

LiDAR realtime odometry is the use of a LiDAR sensor to estimate the **change in position** over time. Using a 3D (360 degree) LiDAR for odometry typically results in very precise position change estimations. However, as no map is generated in this application, the position is always relative to the time you powered-on the LiDAR sensor (sensor initialization time is zero point). A 3D LiDAR will give you the following information relative to the starting time of the LiDAR:

- position in 3D space in meter (x, y, z) relative to LiDAR power-on time
- orientation in 3D angles (roll, pitch, yaw)

When recording this information over time, you will get a 3D trajectory (dotted line):



The following owlRobotics LiDAR sensors are recommended for 3D odometry. Because the LiDAR needs to find enough points, there is a limit on the maximum speed and the size of the maximum open area you can use it. The expected position accuracy depends on many factors (speed, size of open area etc.).

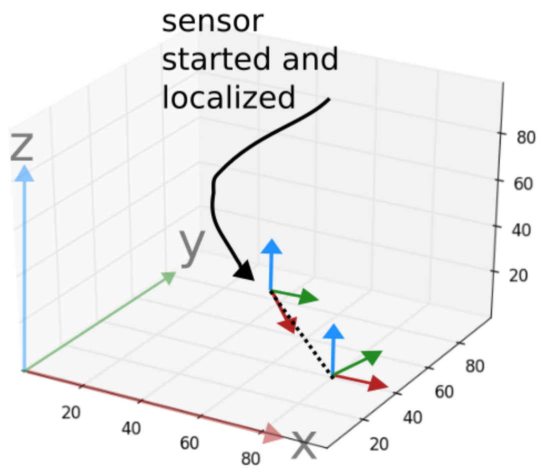
	max. recommended sensor speed	max. recommended open area	expectable position accuracy
Livox MID 360	1 m/s	20m	1-5 cm
LS LIDAR C8/C16	30 m/s	100m	1-5 cm

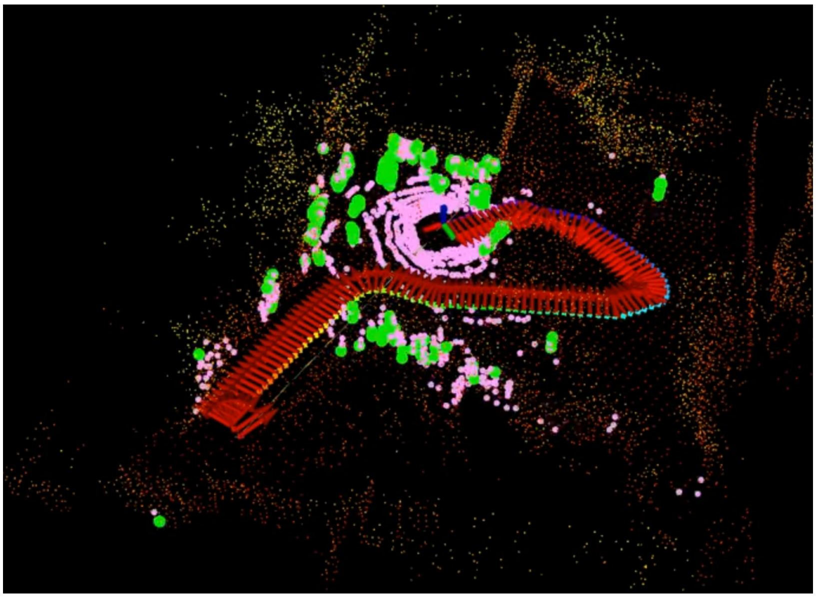
The following ROS software packages are recommended for 3D odometry:

Livox MID 360	https://github.com/SylarAnh/fast_lio_mid360	
LS LIDAR C8/C16	https://github.com/RobustFieldAutonomyLab/LeGO-LOAM	

Localization

A LiDAR sensor can be used to realtime-localize the sensor in the 3D world which means to compute the **position of the LiDAR in the world**. For robust localization, a 360 degree LiDAR is recommended. For outdoor localization, a 3D LiDAR (360 degree) is a must-have. While moving the LiDAR, the software will generate a map of the world and localize the LiDAR in this world (also called 'Simultaneous Localization And Mapping / SLAM algorithm'). For localization, the LiDAR does not need to have many rays as no precise maps are required for localization. When the LiDAR sensor is restarted at another location in the world (also called '**LiDAR kidnapping**'), the software can find the new position of the sensor in the map. This feature is called '**relocalization**'. The generated maps used to localize the sensor are not metric-correct and because of that you will get a misalignment when e.g. moving in a circle-like motion on the ground. The software will detect this misalignment and correct it. This feature is called '**loop-closure**'.





Example: Localization with LS LIDAR C16

The following owlRobotics LiDAR sensors are recommended for 3D localization. Because the LiDAR needs to find enough points, there is a limit on the maximum speed and the size of the maximum open area you can use it. The expected localization accuracy depends on many factors (speed, size of open area etc.).

	max. recommended sensor speed	max. recommended open area
Livox MID 360	1 m/s	20m
LS LIDAR C8/C16	20 m/s	100m

The following ROS software packages are recommended for 3D localization:

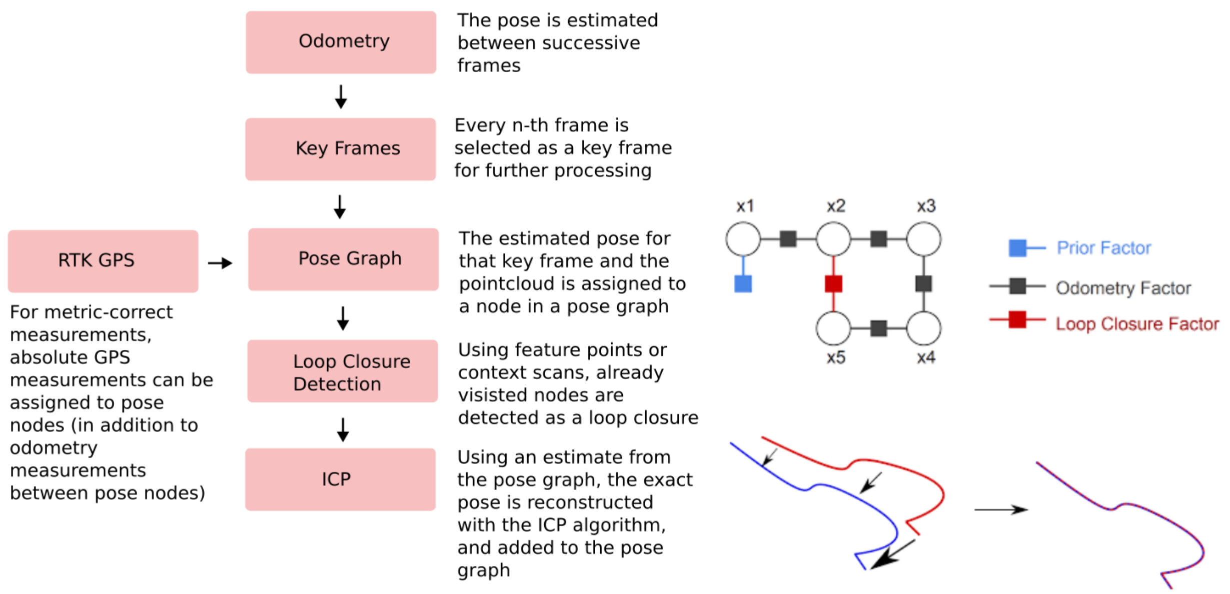
		Relocalization possible?	Expectable localization accuracy	Loop closure possible?
Livox MID 360	https://github.com/SylarAnh/fast_lio_mid360	no	1-5 cm	yes
LS LIDAR C8/C16/C32	https://github.com/Nishantgoyal918/LeGO-LOAM-BOR	yes	1-5 cm	yes

SLAM

How does a SLAM (‘Simultaneous Localization And Mapping’) pipeline work? Here you can see how a typical SLAM pipeline works:

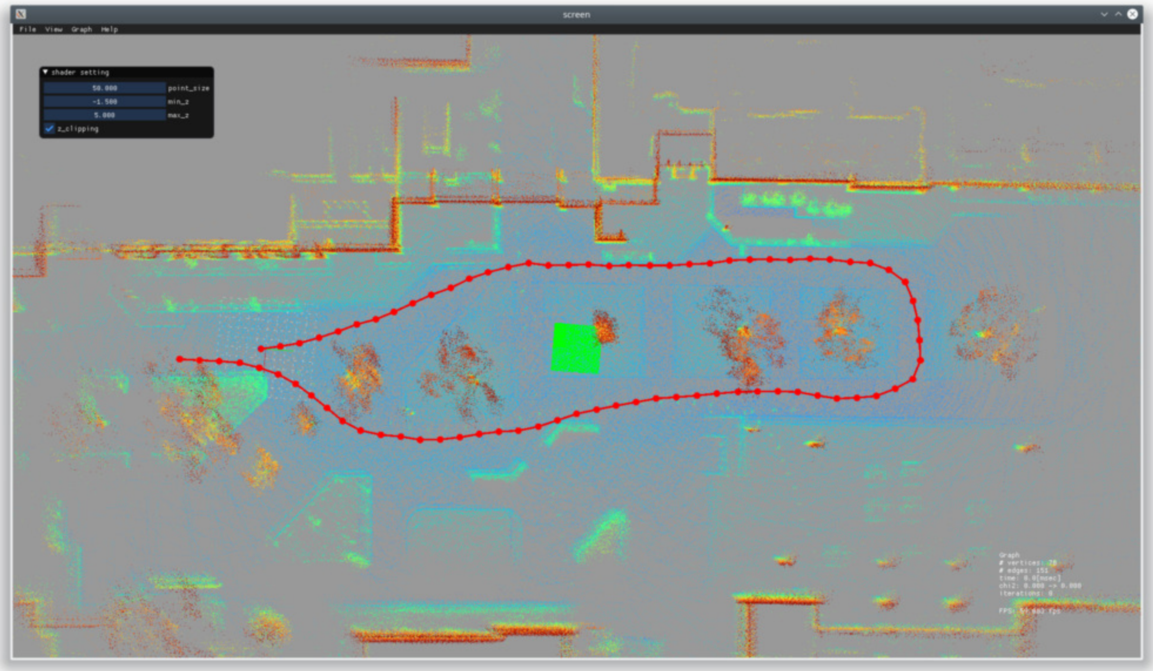
LiDAR **odometry** is used to estimate the pose change between successive frames. Every n-th frame is selected as a **key frame** for further processing. The estimated pose for that key frame and the corresponding point cloud is assigned to a node in a **pose graph**. The nodes in the pose graph are connected by ‘**constraints**’. For example an odometry constraint would be that a certain pose node is a certain length (+x,+y,+z) away from the previous pose node. A RTK **GPS constraint** would be that a certain pose node is at a certain absolute coordinate (x,y,z). By detecting feature points in the point clouds (or using context scans), already visited pose nodes are detected as a **loop closure**, which means that two pose nodes are locally close together. If such a loop closure is detected, a **loop closure constraint** is added between those pose nodes. Also, the exact transformation between those nearly located pose nodes can be computed e.g. using the ICP algorithm. Periodically, the **pose graph** is optimized (the constraints are distributed evenly and hard constraints like absolute positions are considered) and the corresponding key point clouds are transformed correspondingly.

Typical SLAM pipeline:

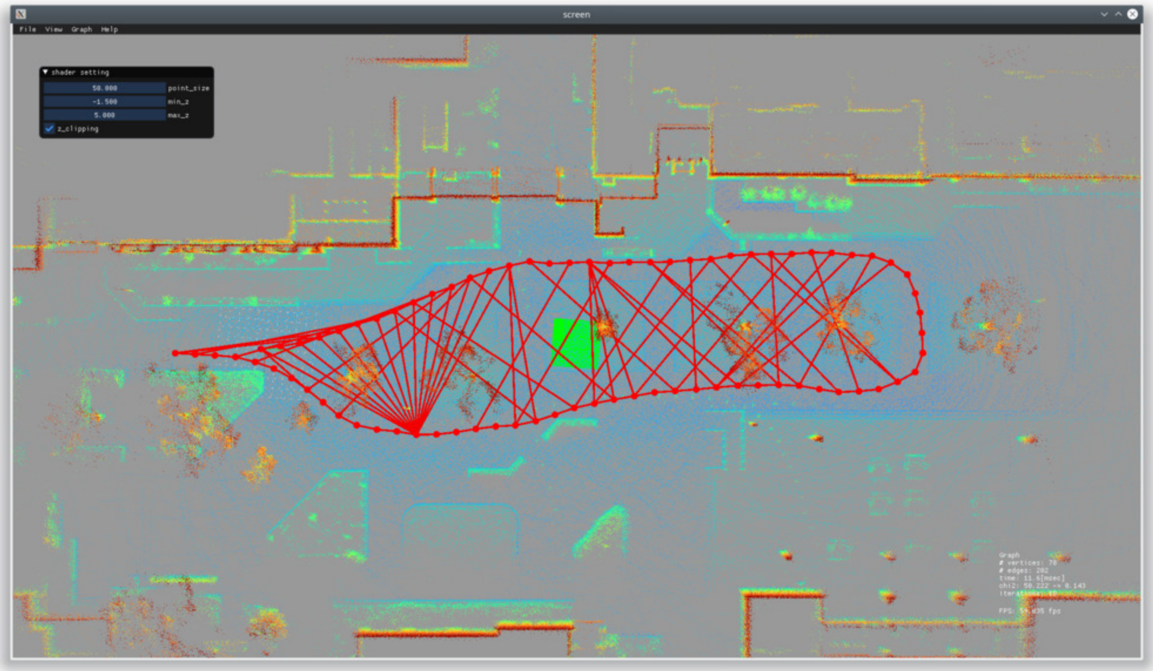


Examples:

Before loop-closing:

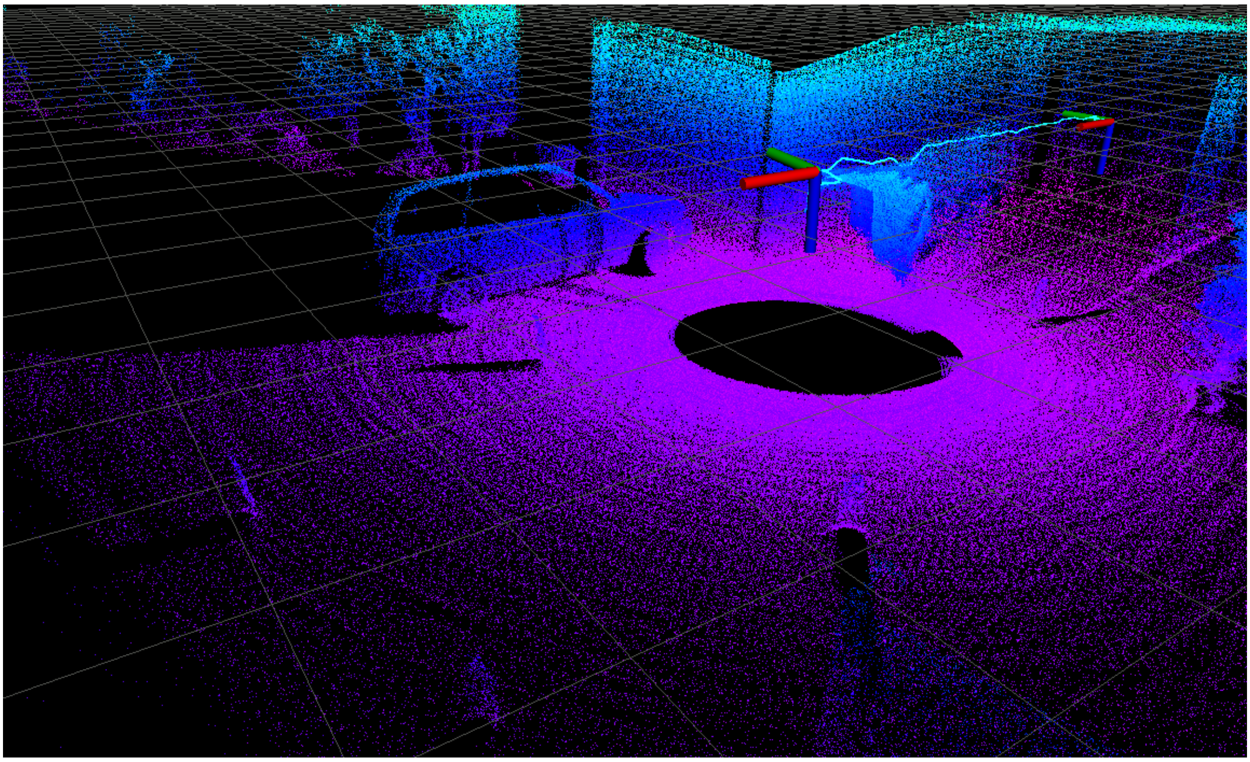


After loop-closing (double walls are **corrected**):



3D Mapping / 3D Object scanning / 3D Measurement

In mapping and scanning applications, the world is precisely measured to generate a **3D map of the world** (or a 3D object). For precise mapping applications, the LiDAR should be able to 'shoot' at all points in the world. Often, for mapping applications, realtime is not required, and one can use 3D software (e.g. CloudCompare, MeshLab etc.) to improve the mapping results. A 3D LiDAR is required for outdoor applications and a 360 degree LiDAR helps to reduce the scanning time.



Example: Mapping with Livox MID-360

The following owlRobotics LiDAR sensors are recommended for 3D mapping. Because the LiDAR needs to find enough points, there is a limit on the maximum speed and the size of the maximum open area you can use it. The expected mapping accuracy depends on many factors (speed, size of open area etc.).

	max. recommended sensor speed	max. recommended open area	expectable mapping accuracy
Livox MID 360	0.5 m/s	20m	1-5 cm
LS LIDAR C32	15 m/s	100m	5-10 cm

The following ROS software packages are recommended for real-time 3D mapping:

		Relocalization possible?	Loop closure possible?	
Livox MID 360	https://github.com/SylarAnh/fast_lio_mid360	no	yes	

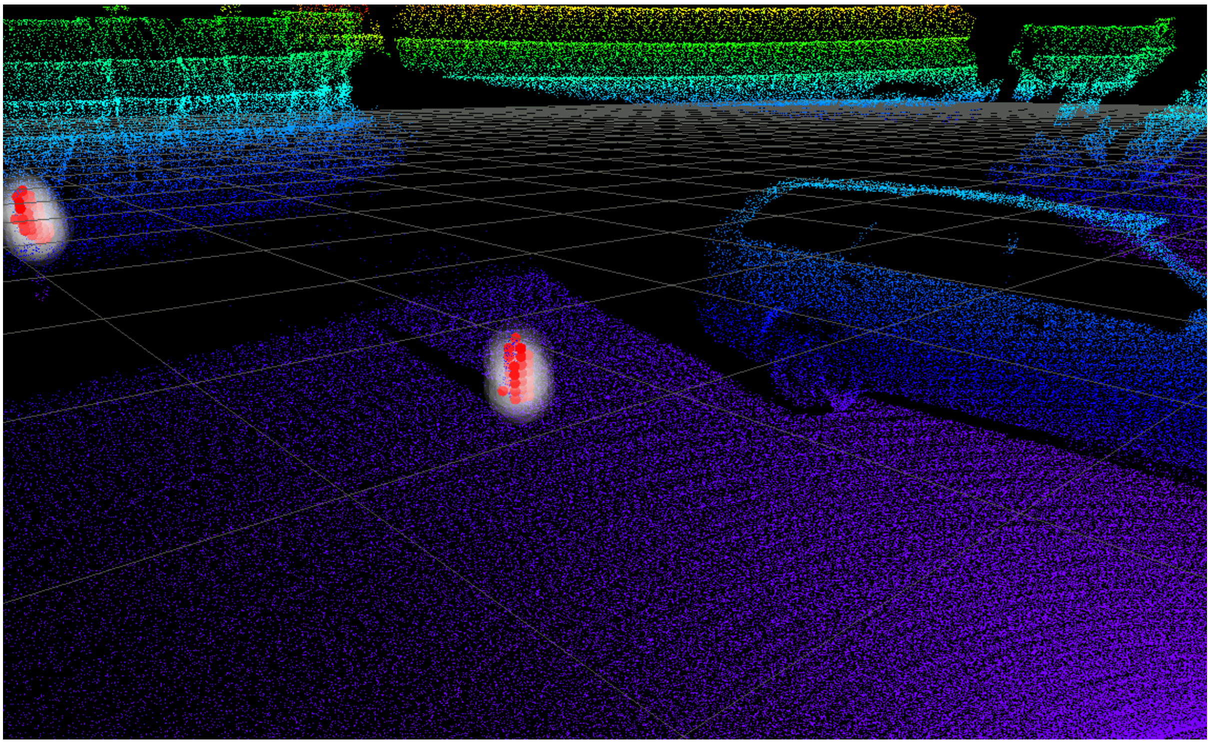
LS LIDAR C32	https://github.com/RobustFieldAutonomyLab/LeGO-LOAM	yes	yes	

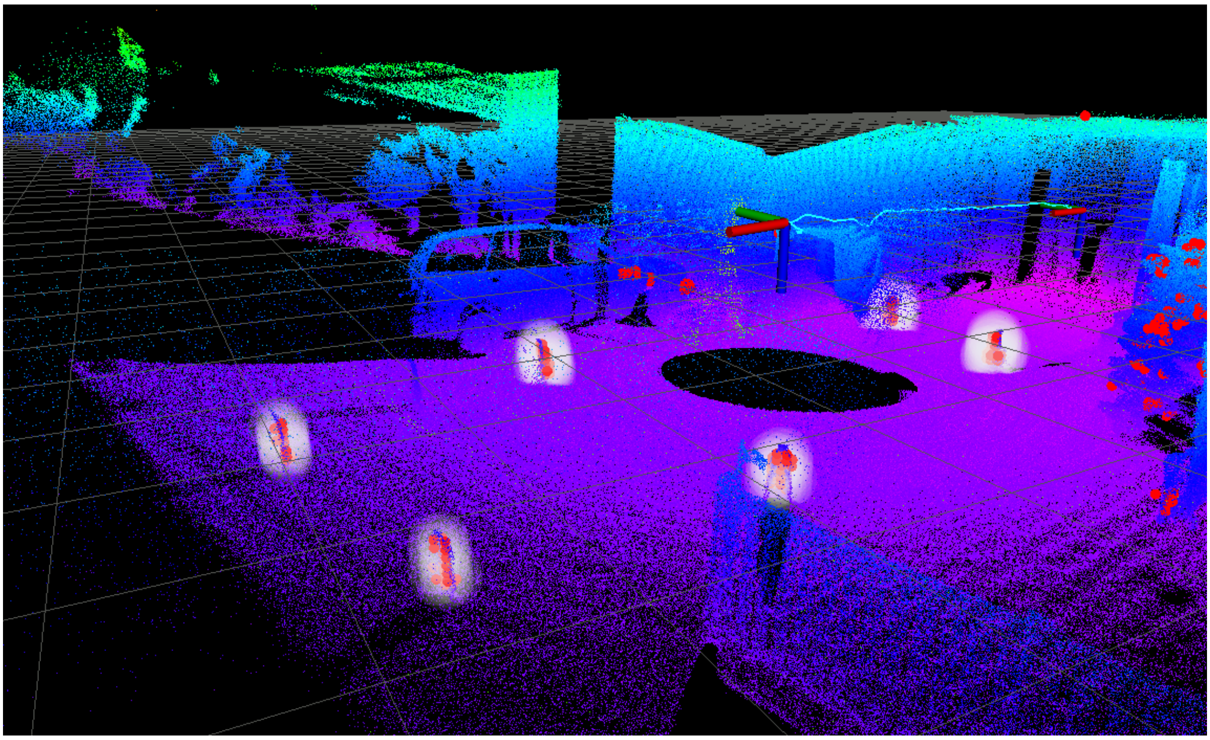
Object recognition/tracking and obstacle detection

A LiDAR sensor can be used to **recognize and track objects** or to detect obstacles. Example objects are people, cars, traffic signs etc. Because a LiDAR sensor cannot detect colors but instead can detect reflectivity strength, you can detect black and white patterns with it.

Depending on your application, you can identify objects by:

- size (width, depth, height)
- distance to ground (object height from ground)
- reflectivity (from back to white color)
- surrounding objects and points
- feature points (points with certain properties like belonging to a surface or edge points)
- 3D object templates (comparing point clusters with a point cloud template based on feature points, euclidean distance etc.)





Example: Object detection with Livox MID-360

Depending on motion speed of the sensor, the size of the object etc. we recommend the following owlRobotics LiDAR sensors for object recognition/tracking and obstacle detection:

	Max.	Min. object size expectable

	recommended sensor speed	
Livox MID 360	0.2 m/s	0.2 m (near), 0.5m (far)
Livox MID 360	0.5 m/s	0.5 m (near), 1.0m (far)
LS LIDAR C8/16	15 m/s	0.5 m (near), 1.0m (far)
LS LIDAR C32	15 m/s	0.2 m (near), 0.5m (far)

The following software ROS packages are recommended for 3D obstacle detection, object detection and tracking:

PCL (Point Cloud Library)	http://wiki.ros.org/pcl/Tutorials
PCL (ROS)	http://wiki.ros.org/pcl_ros/Tutorials

LiDAR viewer software (Windows / Mac / Linux)

LS LIDAR:
Livox: https://www.livoxtech.com/de/mid-360/downloads

ROS driver software (Linux)

Robotic operating system (ROS) drivers:
LS LIDAR: https://github.com/Lslidar/Lslidar_ROS1_driver
Livox: https://github.com/Livox-SDK/livox_ros_driver2

Your project is our spirit

Tell us what you need for your project - owlRobotics can help with Open Source software as well has developed proprietary software that can be used and adopted on request - examples:

- Object detection (traffic cones etc.) and LiDAR odometry (Livox MID 360)
  https://www.youtube.com/watch?v=r4irRZcBTy8
  https://www.youtube.com/watch?v=Vv_SAqXiNP4

- Moving object (persons etc.) tracking (LSLIDAR C16)
  https://www.youtube.com/watch?v=N8ziA8BsCNk
  https://www.youtube.com/watch?v=J9Z_Dh7w2zw

- Localization (LSLIDAR C16)
  https://www.youtube.com/watch?v=BIVYWe2SjLY
  https://www.youtube.com/watch?v=XMt8soaGbpQ
  https://www.youtube.com/watch?v=CcamIQFkaog
  https://www.youtube.com/watch?v=Dh_WT0Na0mA
  https://www.youtube.com/watch?v=v2qZtLXayS8

- Obstacle and ground detection (LSLIDAR C16)
  https://www.youtube.com/watch?v=S-Qs9OPLtkc