CHALMERS
UNIVERSITY OF TECHNOLOGY

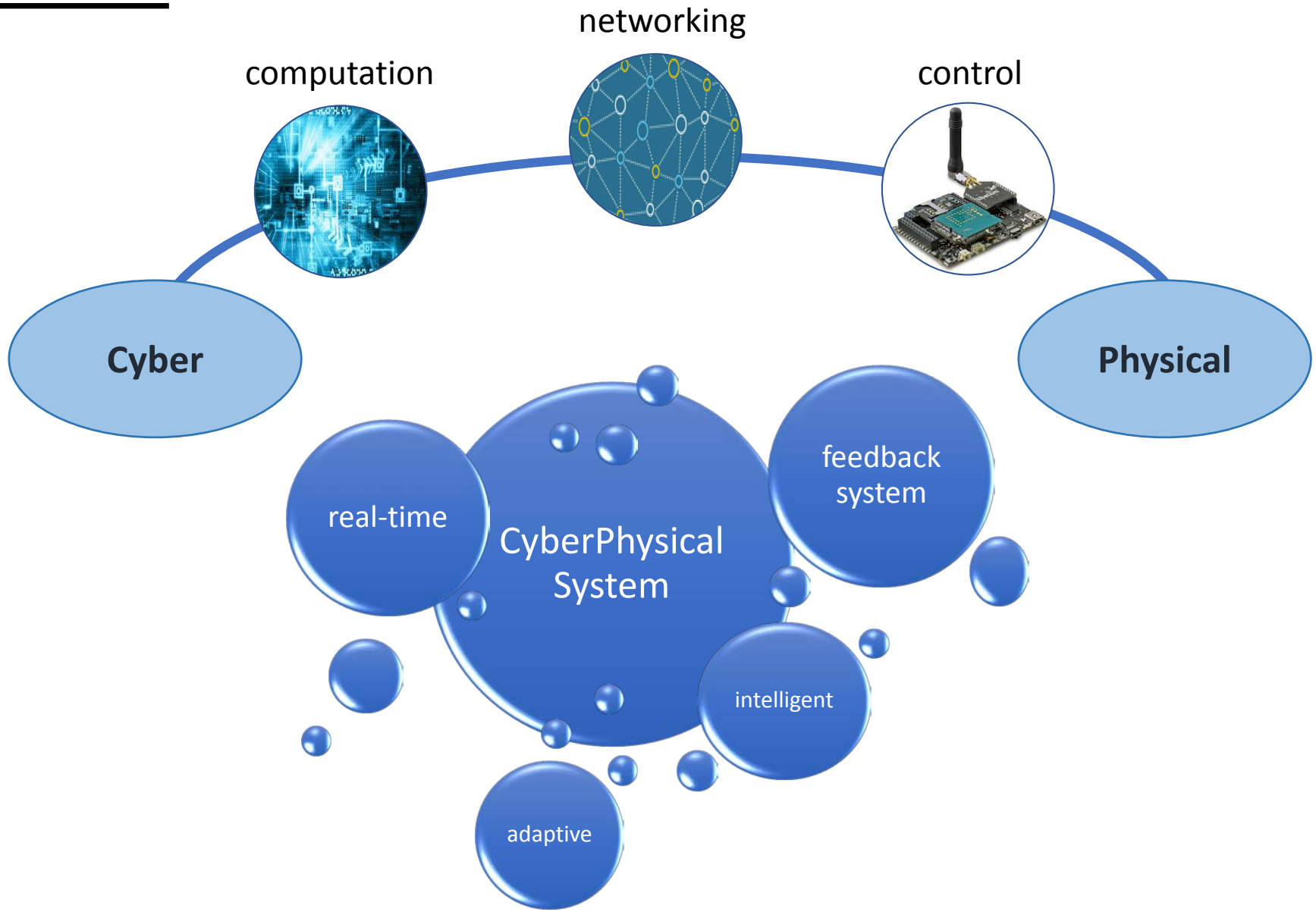
Distributed Computing and Systems
Chalmers university of technology

LiDAR Data Processing For Automated Environments

Hannaneh Najdataei

.: October 2017 :.

Introduction



CPS Applications

Health care

Energy

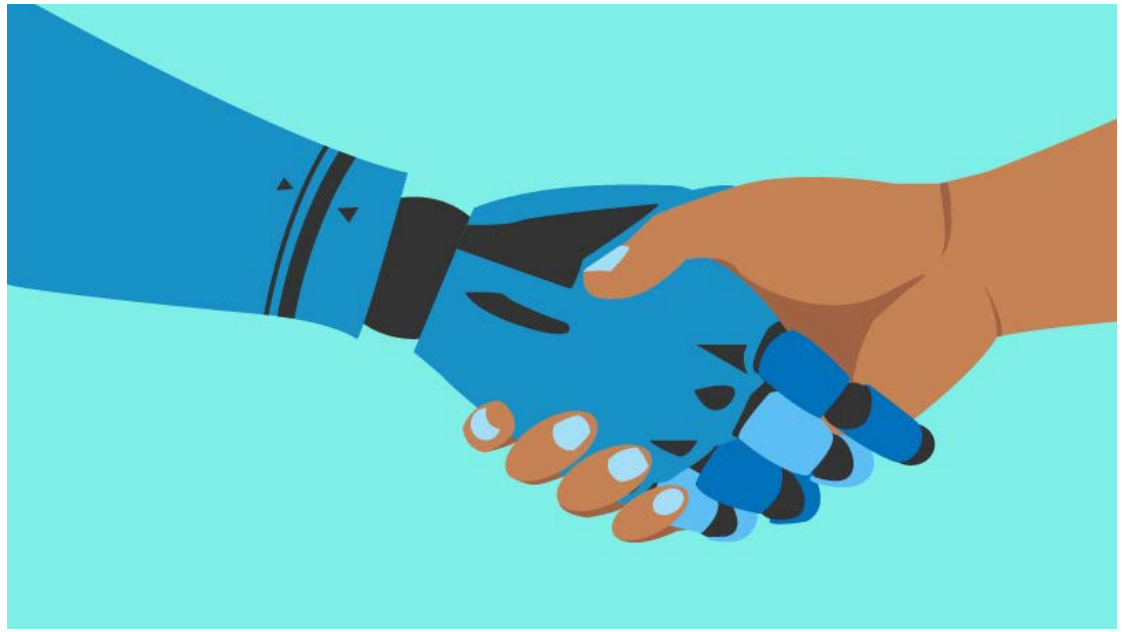
Infrastructure

Manufacturing

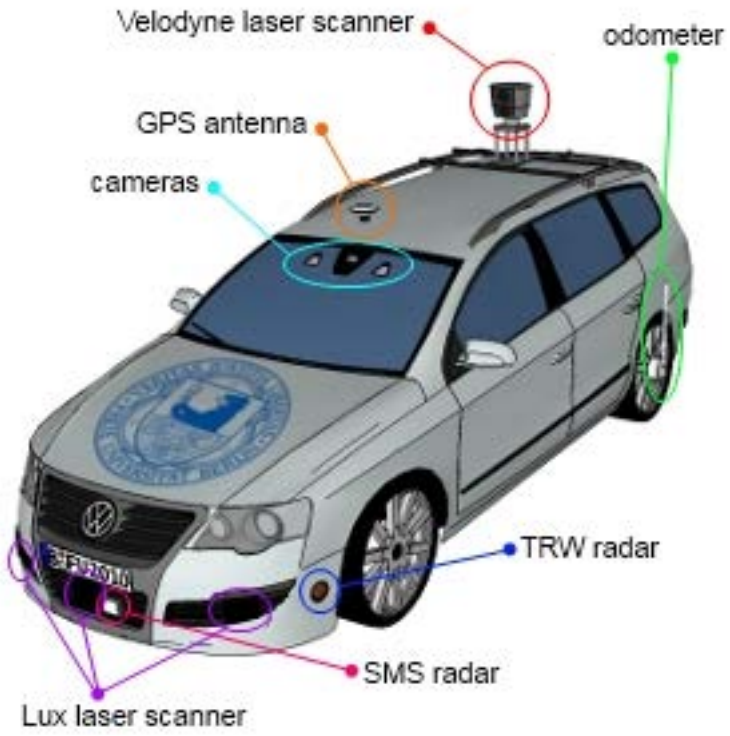
Robotics

Transportation

Automated Environment



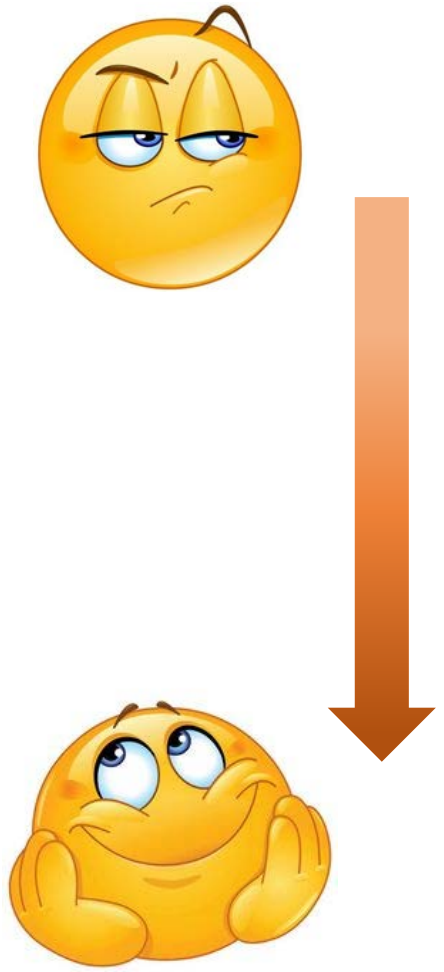
✓ Makes life easier! and safer



Pic: robotland

Outline

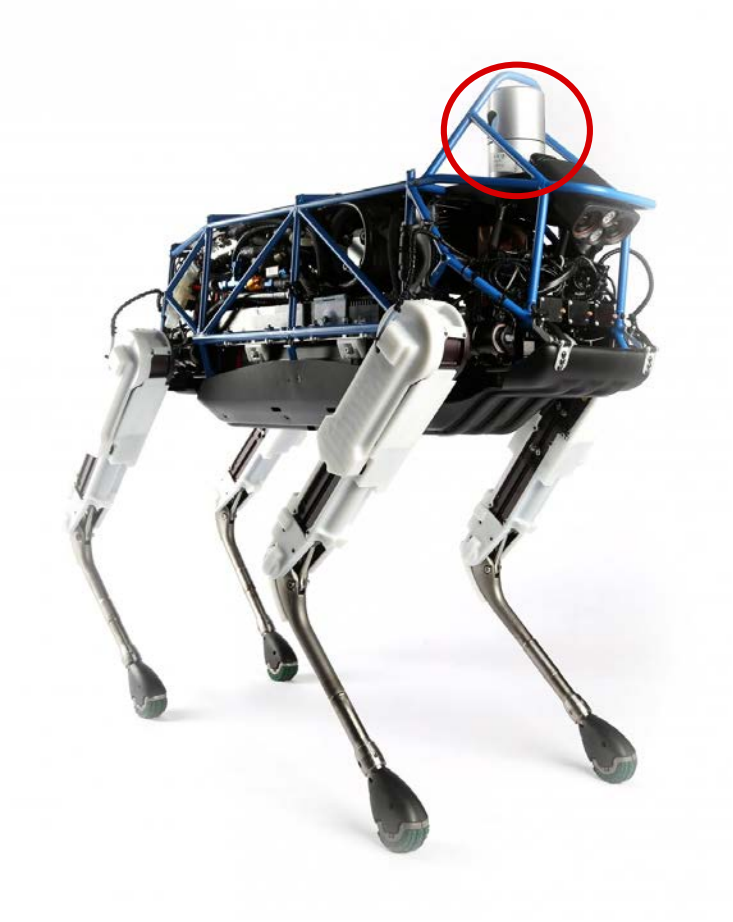
- LiDAR and Point Cloud
- Point Cloud Library
- Webots Simulator
- Use cases



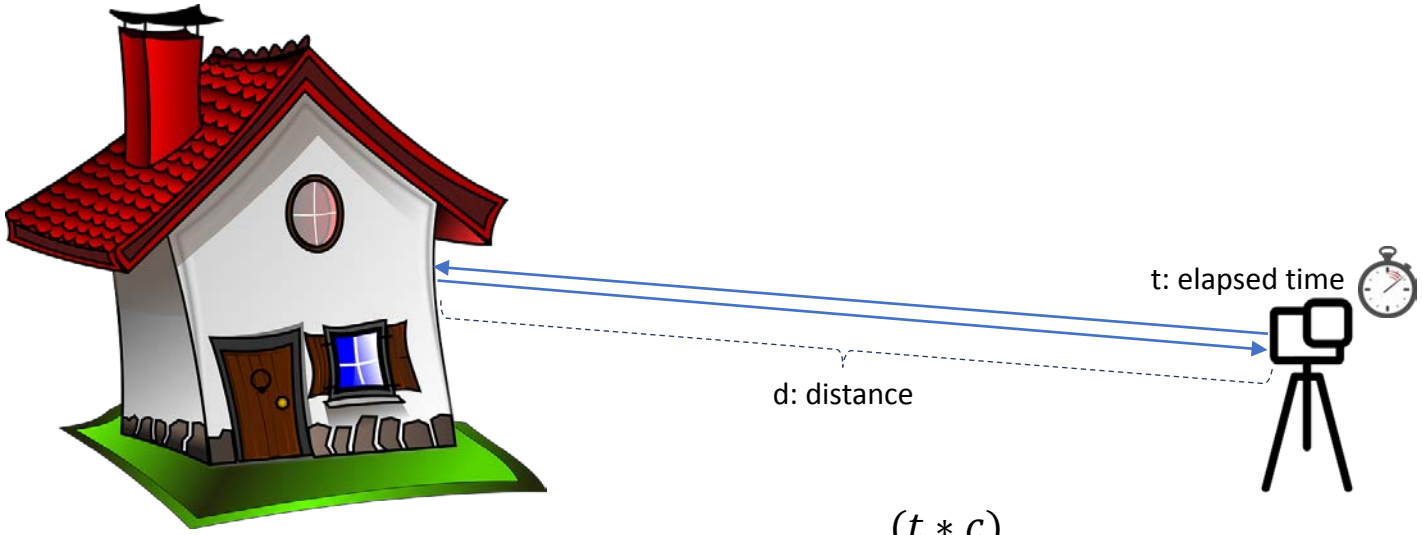
LiDAR and Point Cloud


Introducing Spot

LiDAR

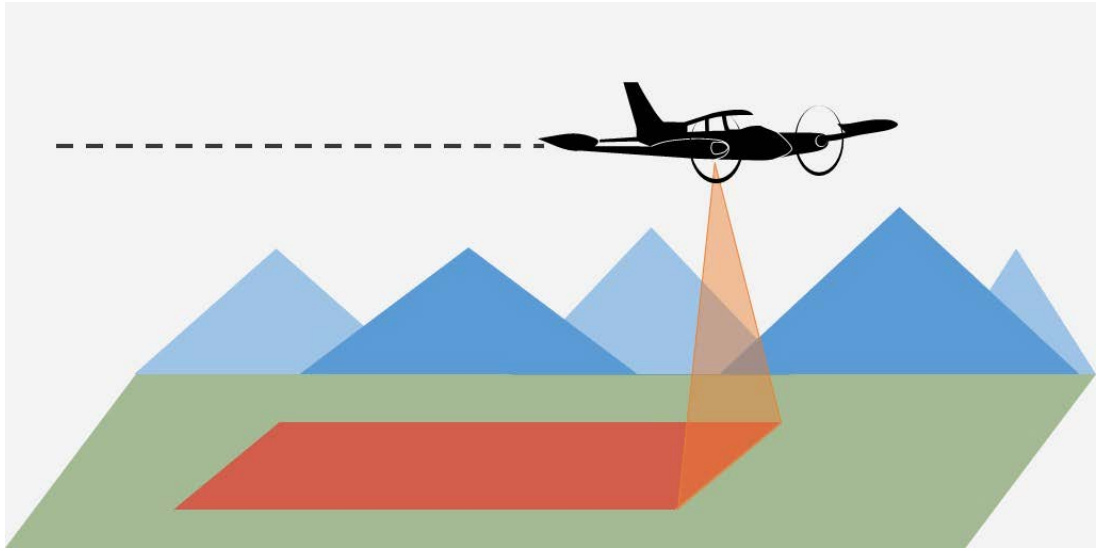


- Portmanteau word of *light* and *radar*
- Stands for Light Detection And Ranging

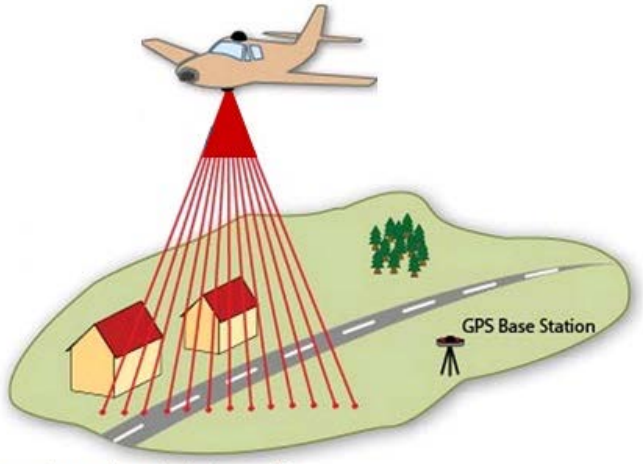


$$\mathrm{d} = \frac{(t * c)}{2}$$

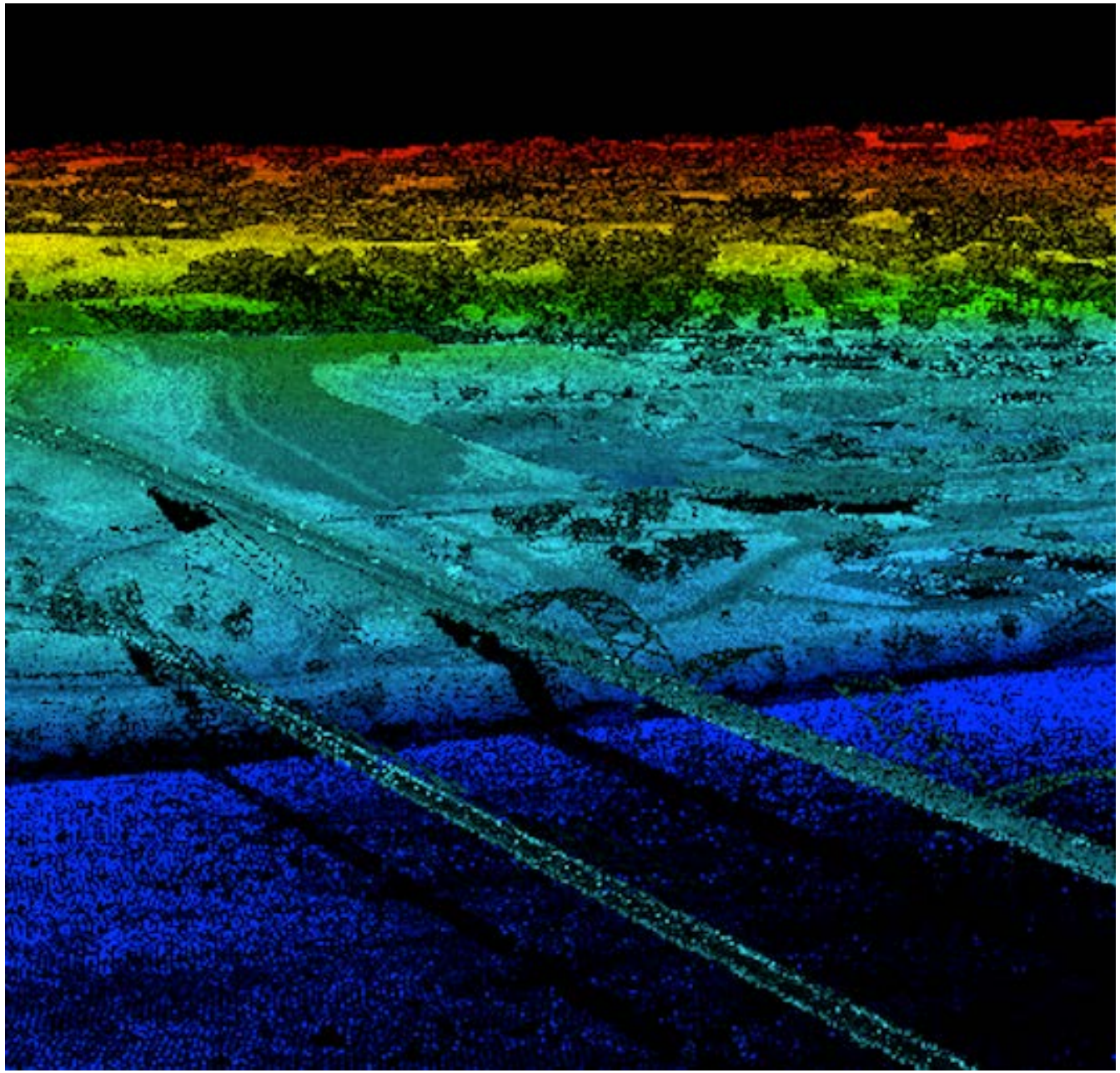
Airborne LiDAR



Airborne LiDAR – 3D mapping



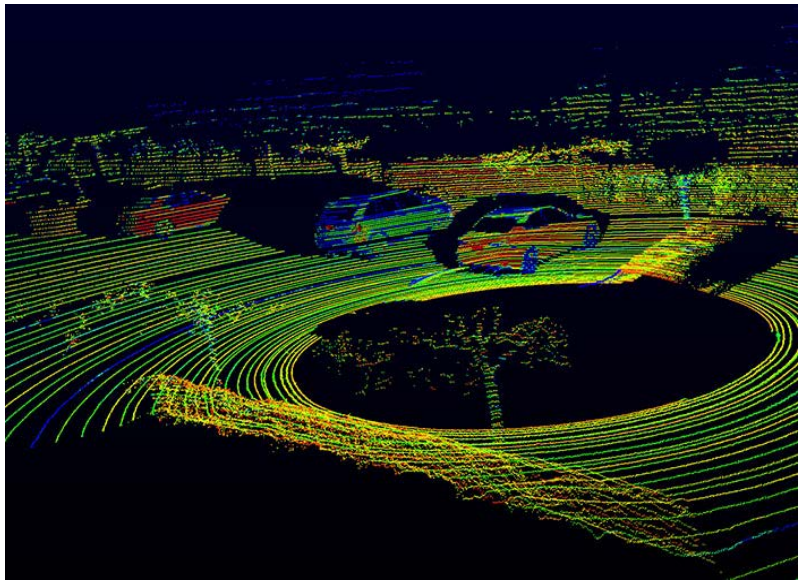
http://emv.cast.uark.edu/wp-content/uploads/2013/01/ALS_scematic.jpg

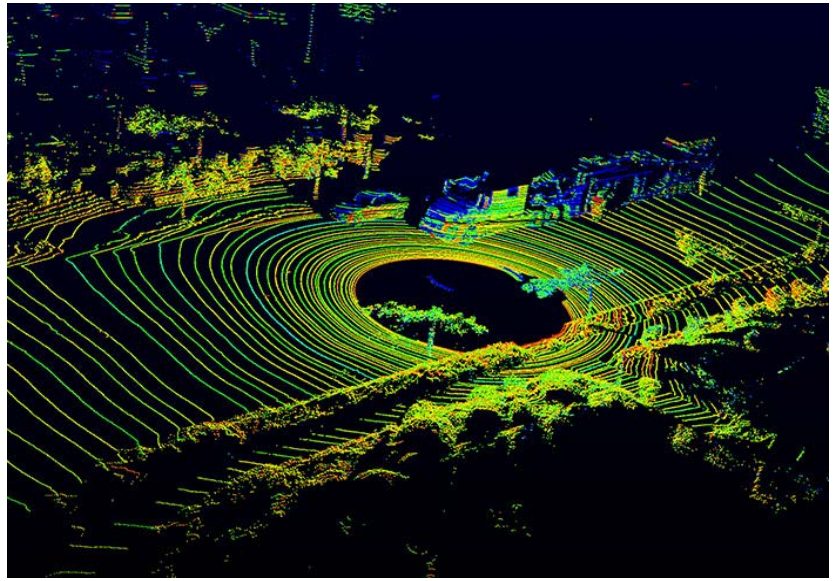


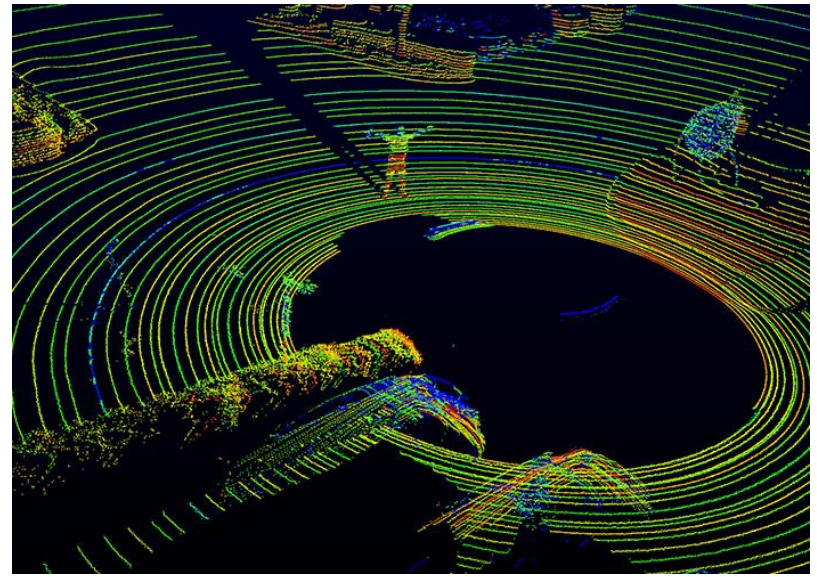
Pic: airborneimaginginc.com

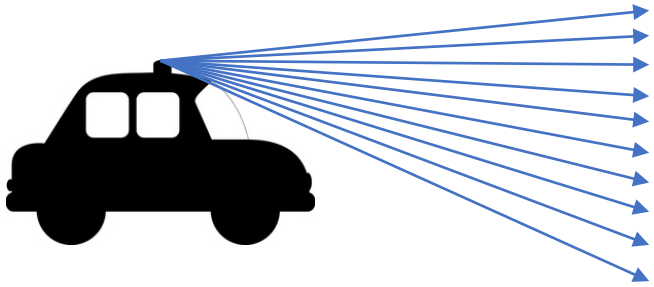
Rotating LiDAR

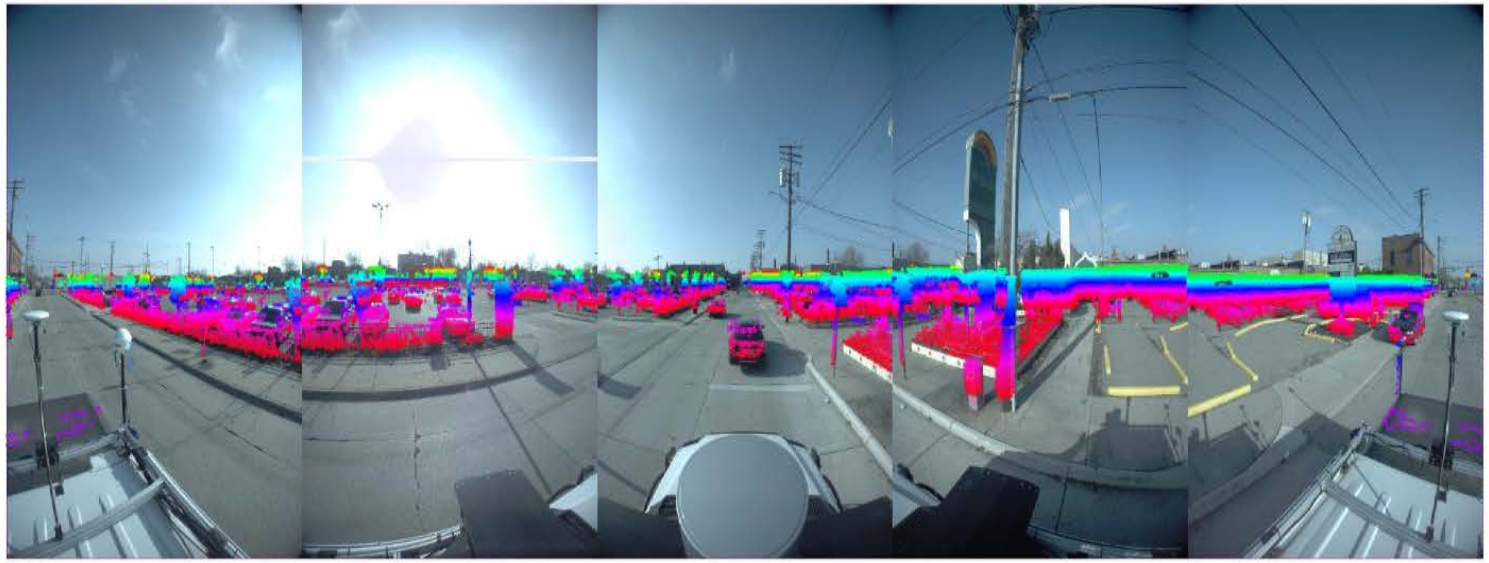
Pic: point cloud library





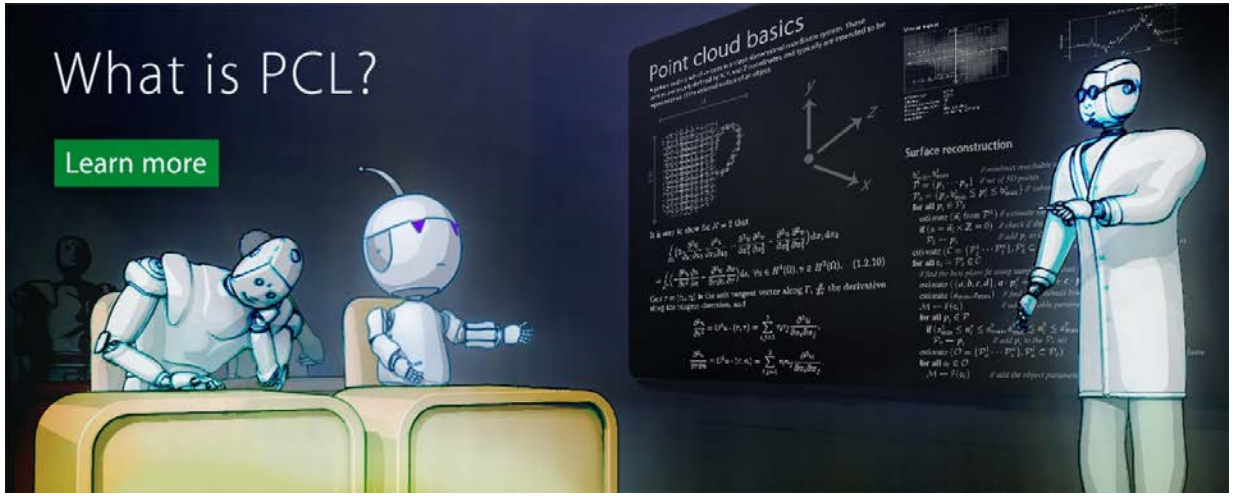


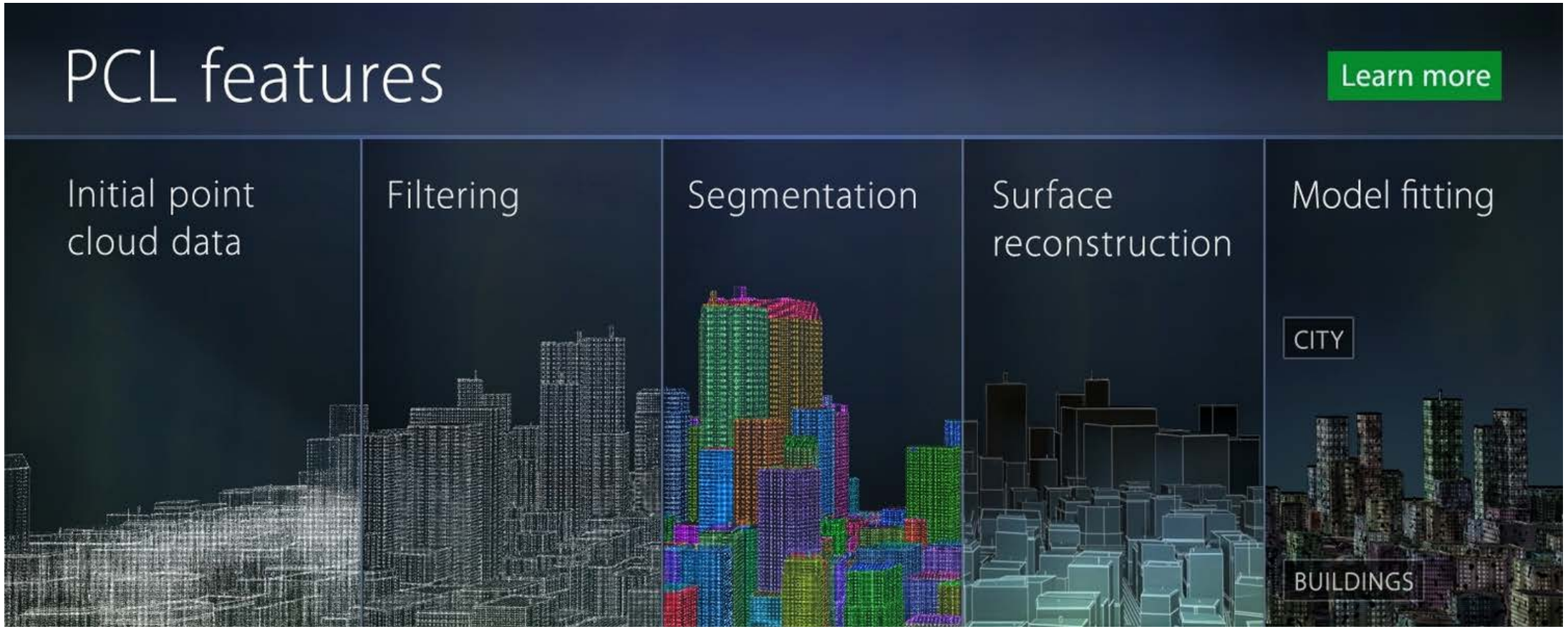




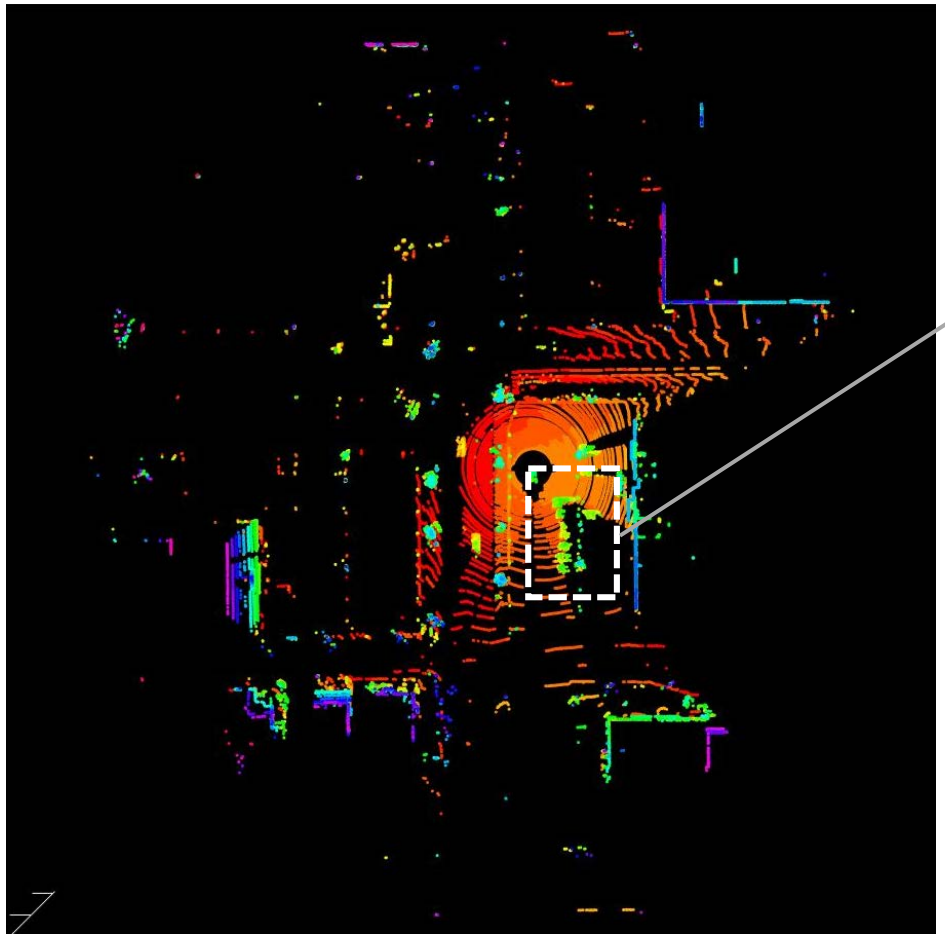
Point Cloud Library

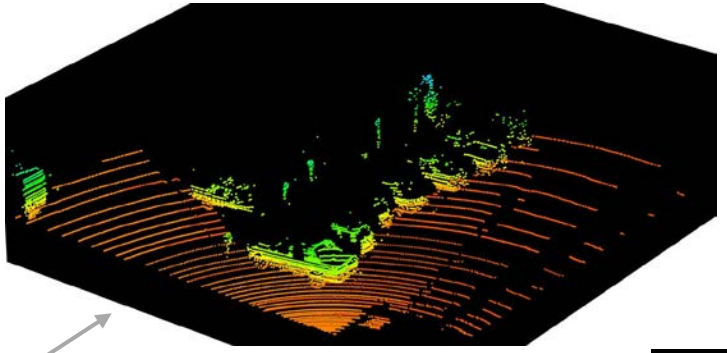
pointcloudlibrary



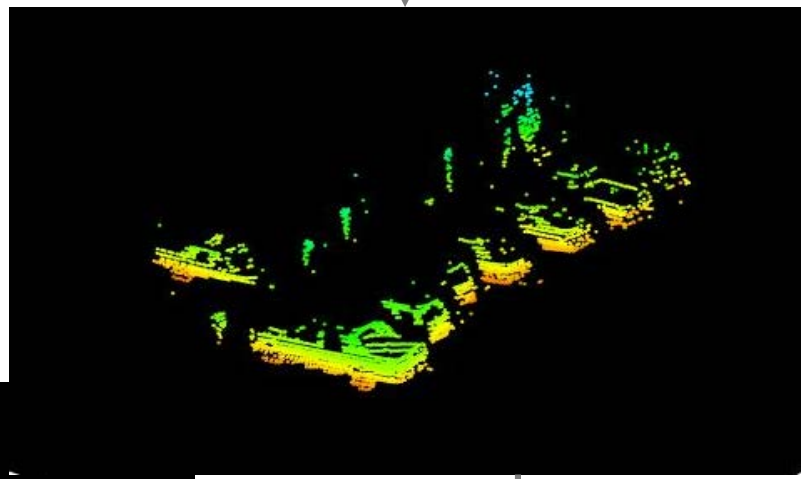


PCL - Segmentation





Removing the Ground



Clustering

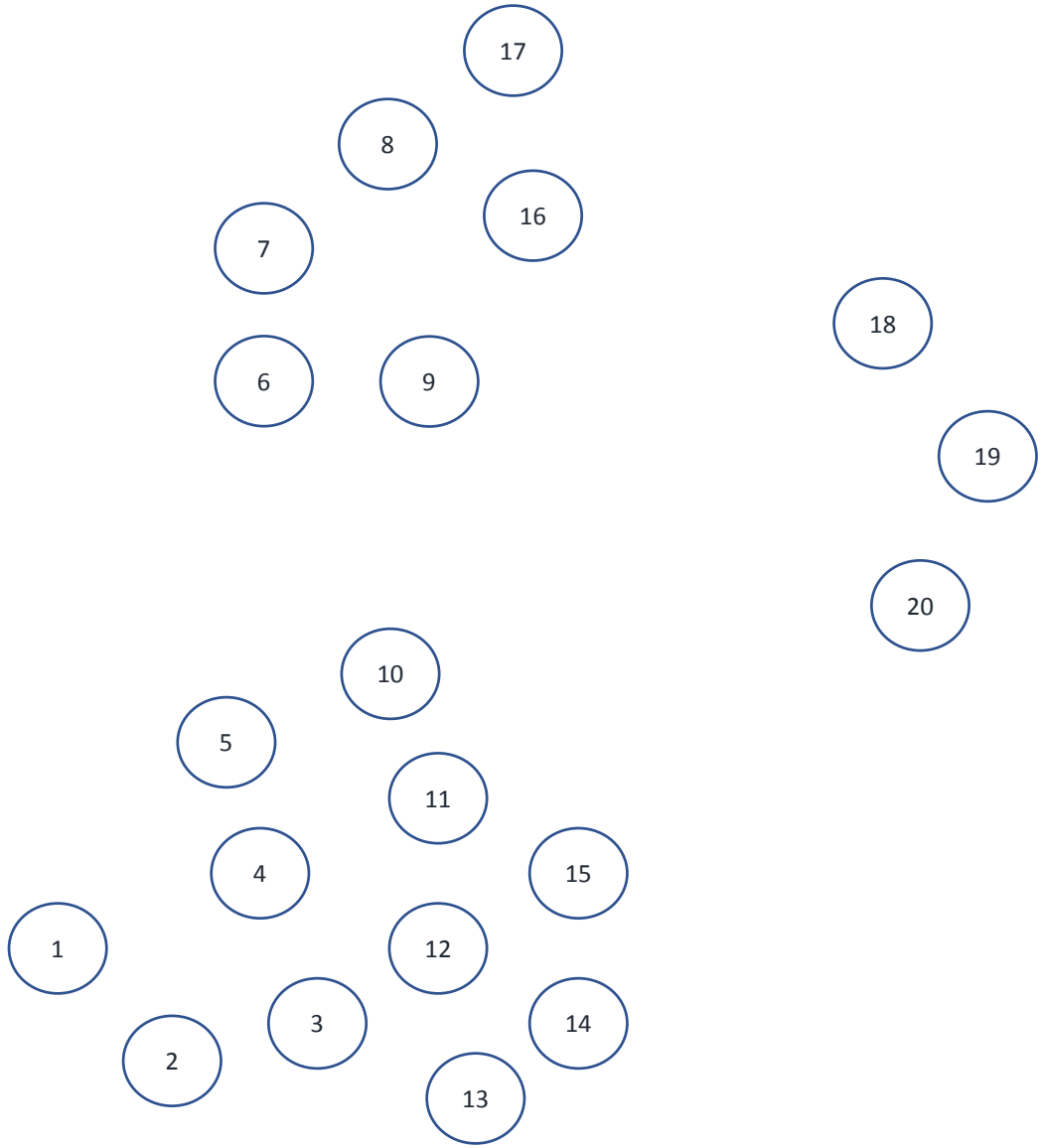
Euclidean Cluster Extraction

$PCL - EclideanClusterExtraction(eps, minPts)$

```
1: clusters = ∅
2: for p ∈ P do
3:     Q = ∅
4:     if p.status ≠ processed then
5:         Q.add(p)
6:         for q ∈ Q do
7:             q.status = 'processed'
8:             N = GetNeighbors(q, ε)
9:             Q.addAll(N)
10:        if size(Q) ≥ minPts then clusters.add(Q)
```

$clusters = \{\}$

$Q = \{\}$



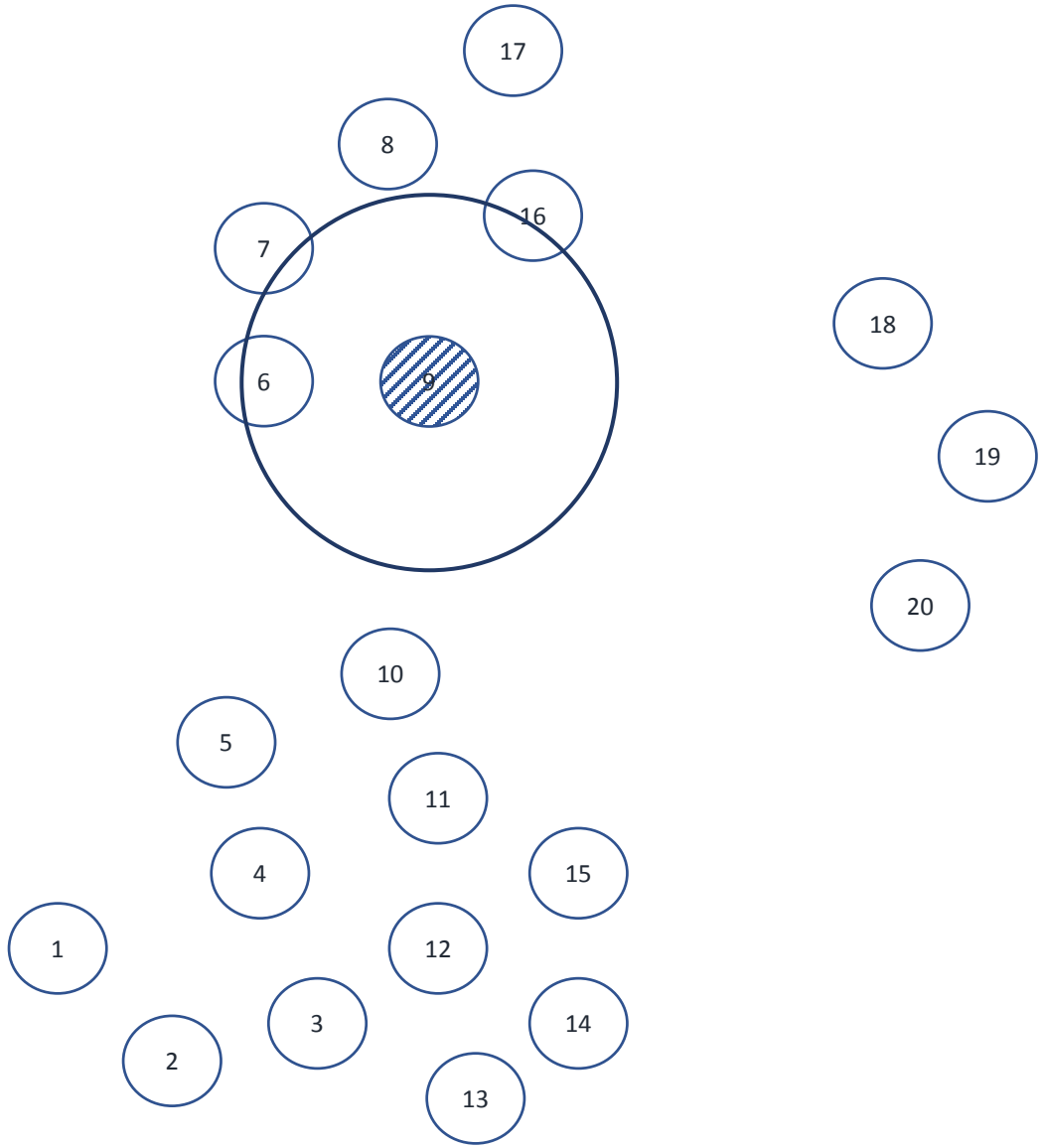
Euclidean Cluster Extraction

$PCL - EclideanClusterExtraction(eps, minPts)$

```
1: clusters = ∅
2: for p ∈ P do
3:     Q = ∅
4:     if p.status ≠ processed then
5:         Q.add(p)
6:         for q ∈ Q do
7:             q.status = 'processed'
8:             N = GetNeighbors(q, ε)
9:             Q.addAll(N)
10:        if size(Q) ≥ minPts then clusters.add(Q)
```

$clusters = \{\}$

$Q = \{9\}$



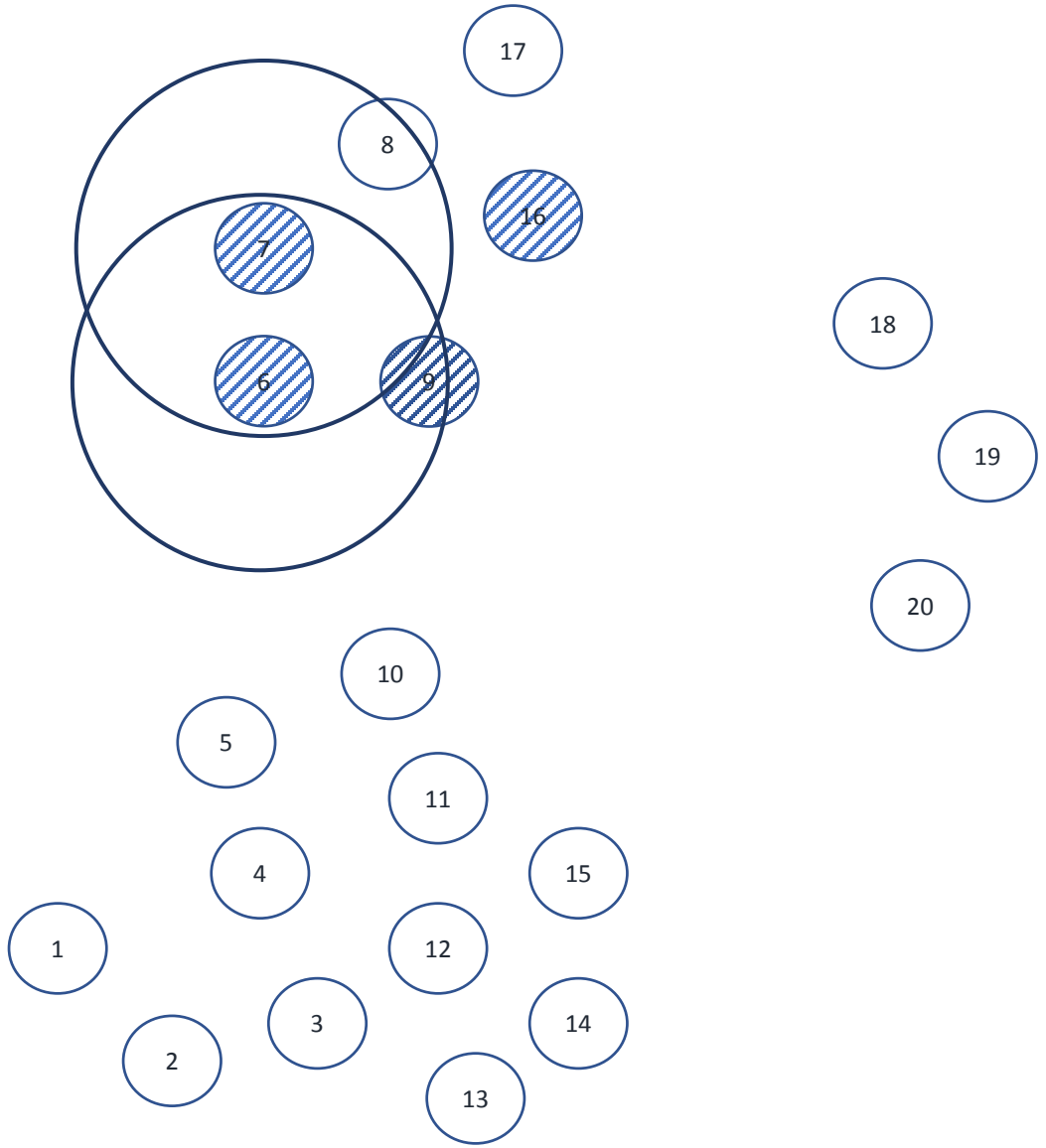
Euclidean Cluster Extraction

$PCL - EclideanClusterExtraction(eps, minPts)$

```
1: clusters = ∅
2: for p ∈ P do
3:     Q = ∅
4:     if p.status ≠ processed then
5:         Q.add(p)
6:         for q ∈ Q do
7:             q.status = 'processed'
8:             N = GetNeighbors(q, ε)
9:             Q.addAll(N)
10:        if size(Q) ≥ minPts then clusters.add(Q)
```

$clusters = \{\}$

$Q = \{9,6,7,16\}$



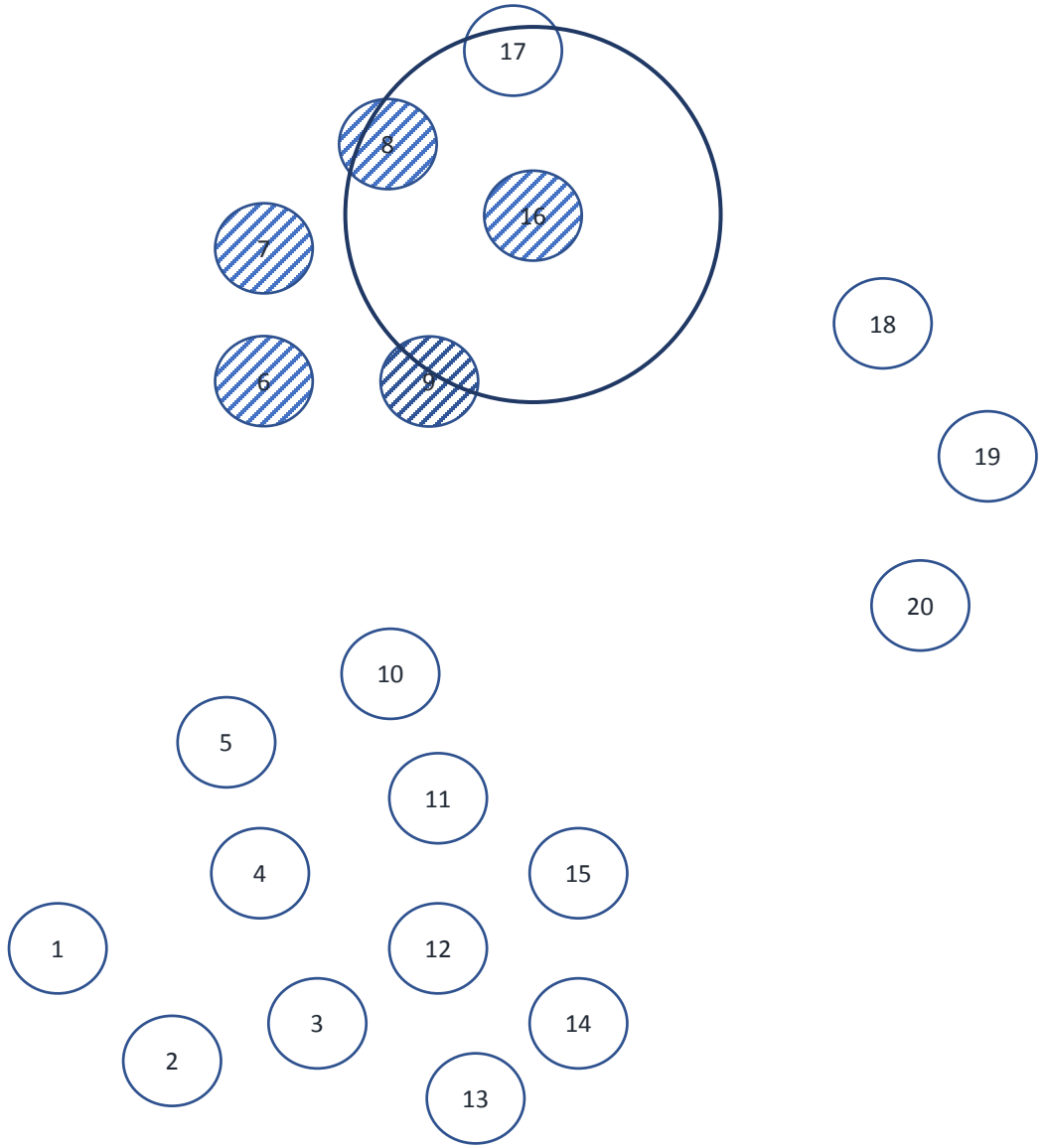
Euclidean Cluster Extraction

$PCL - EclideanClusterExtraction(eps, minPts)$

```
1: clusters = ∅
2: for p ∈ P do
3:     Q = ∅
4:     if p.status ≠ processed then
5:         Q.add(p)
6:         for q ∈ Q do
7:             q.status = 'processed'
8:             N = GetNeighbors(q, ε)
9:             Q.addAll(N)
10:        if size(Q) ≥ minPts then clusters.add(Q)
```

$clusters = \{\}$

$Q = \{9,6,7,16,8\}$



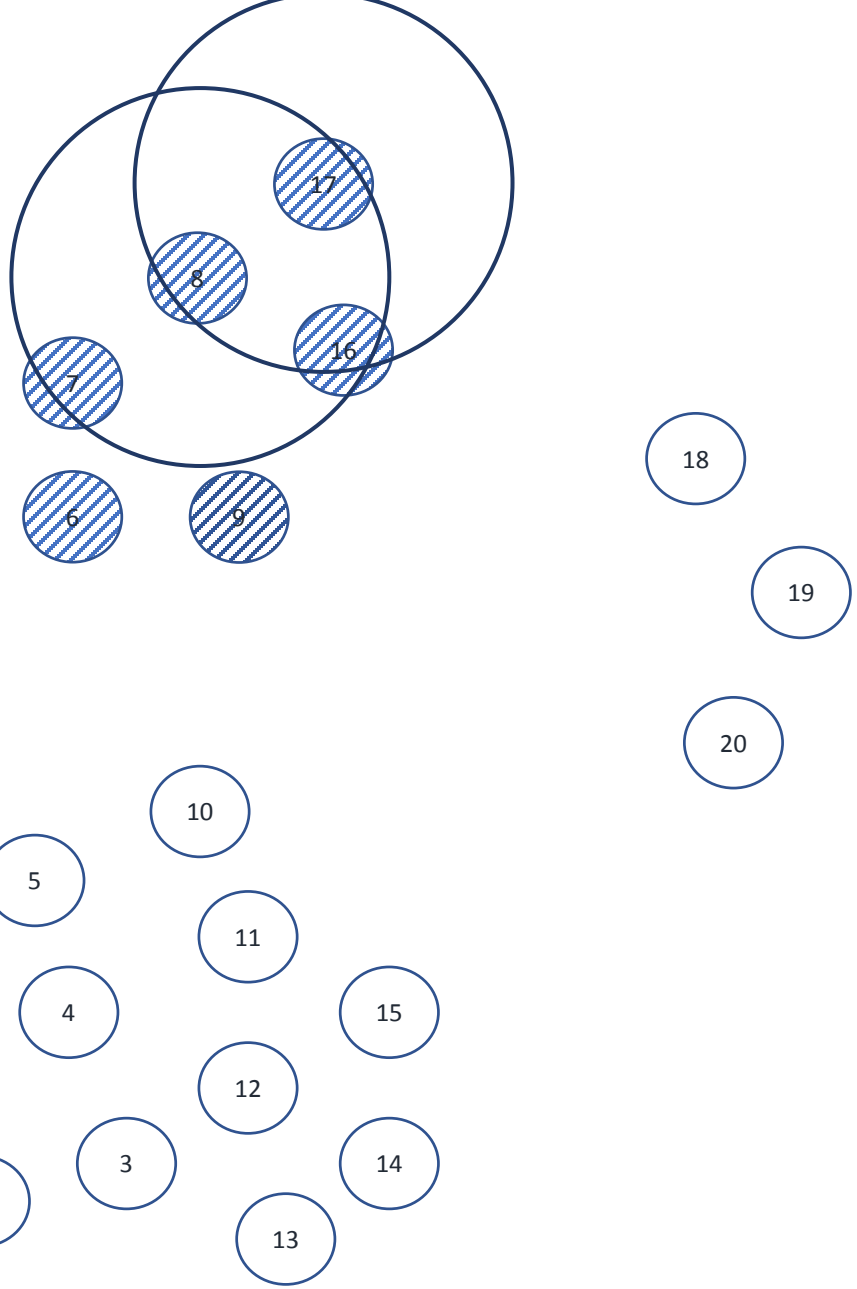
Euclidean Cluster Extraction

$PCL - EclideanClusterExtraction(eps, minPts)$

```
1: clusters = ∅
2: for p ∈ P do
3:     Q = ∅
4:     if p.status ≠ processed then
5:         Q.add(p)
6:         for q ∈ Q do
7:             q.status = 'processed'
8:             N = GetNeighbors(q, ε)
9:             Q.addAll(N)
10:        if size(Q) ≥ minPts then clusters.add(Q)
```

$clusters = \{\}$

$Q = \{9,6,7,16,8,17\}$



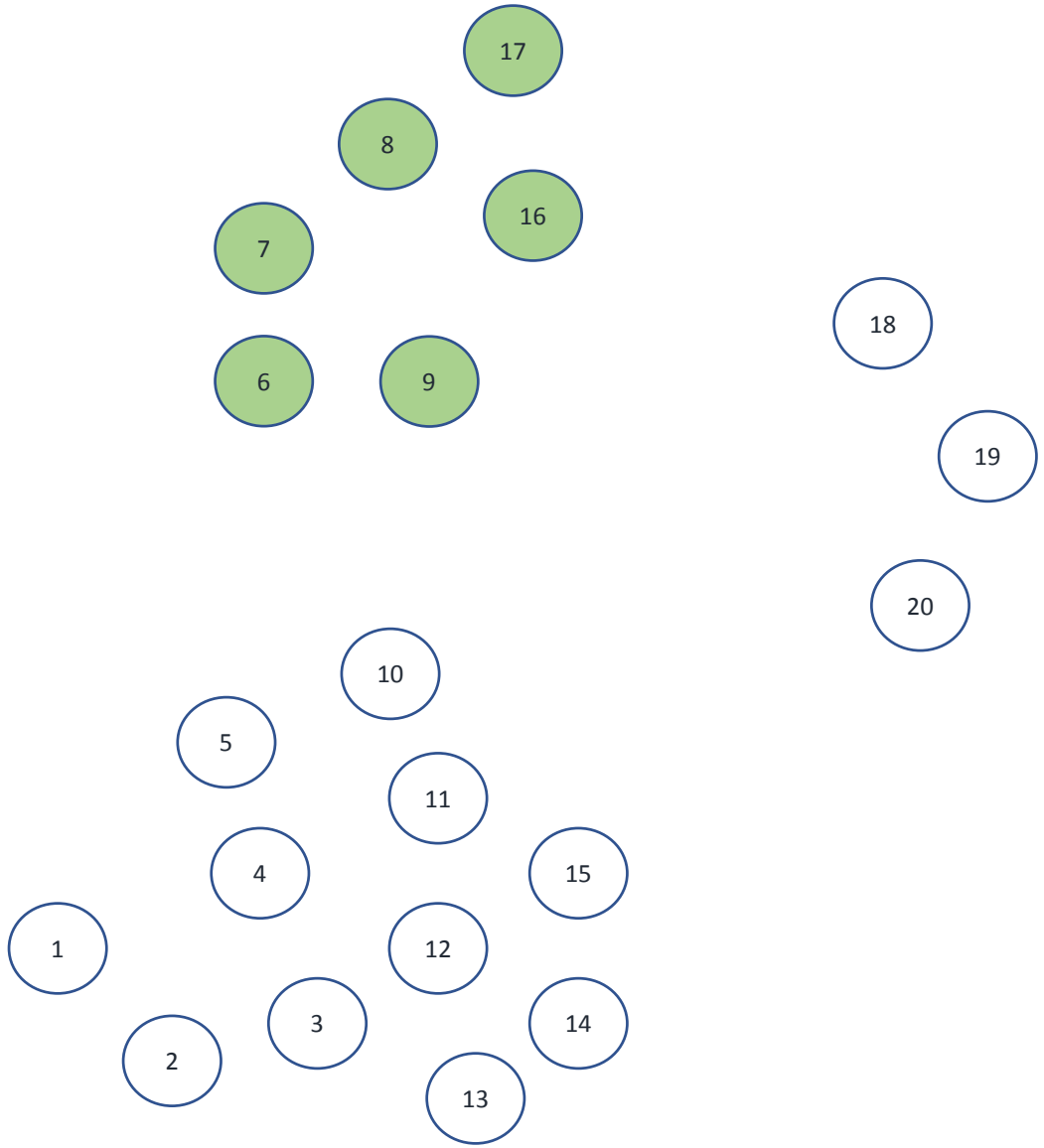
Euclidean Cluster Extraction

$PCL - EclideanClusterExtraction(eps, minPts)$

```
1: clusters = ∅
2: for p ∈ P do
3:     Q = ∅
4:     if p.status ≠ processed then
5:         Q.add(p)
6:         for q ∈ Q do
7:             q.status = 'processed'
8:             N = GetNeighbors(q, ε)
9:             Q.addAll(N)
10:        if size(Q) ≥ minPts then clusters.add(Q)
```

$clusters = \{C_1(9,6,7,16,8,17)\}$

$Q = \{\}$



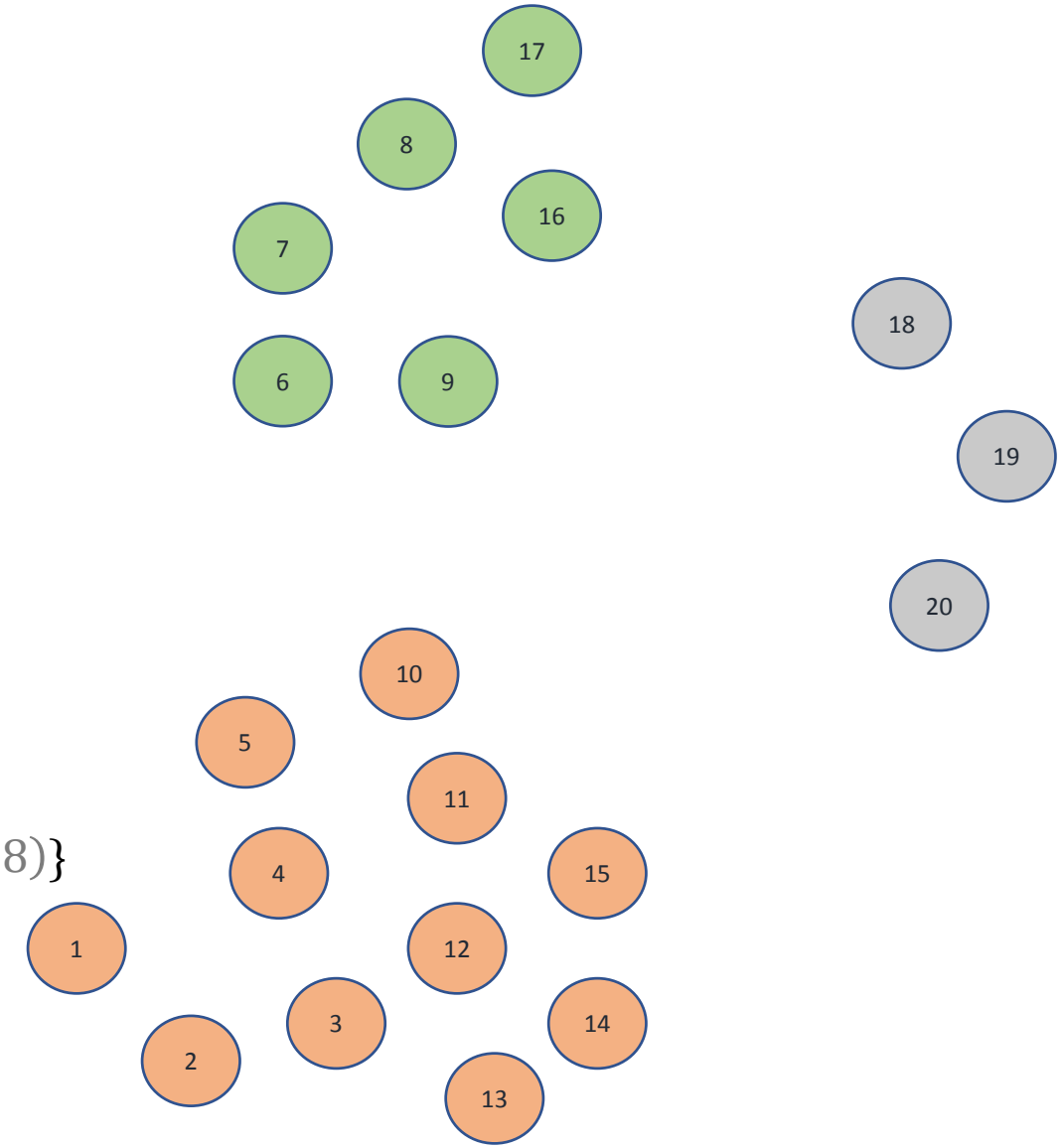
Euclidean Cluster Extraction

$PCL - EclideanClusterExtraction(eps, minPts)$

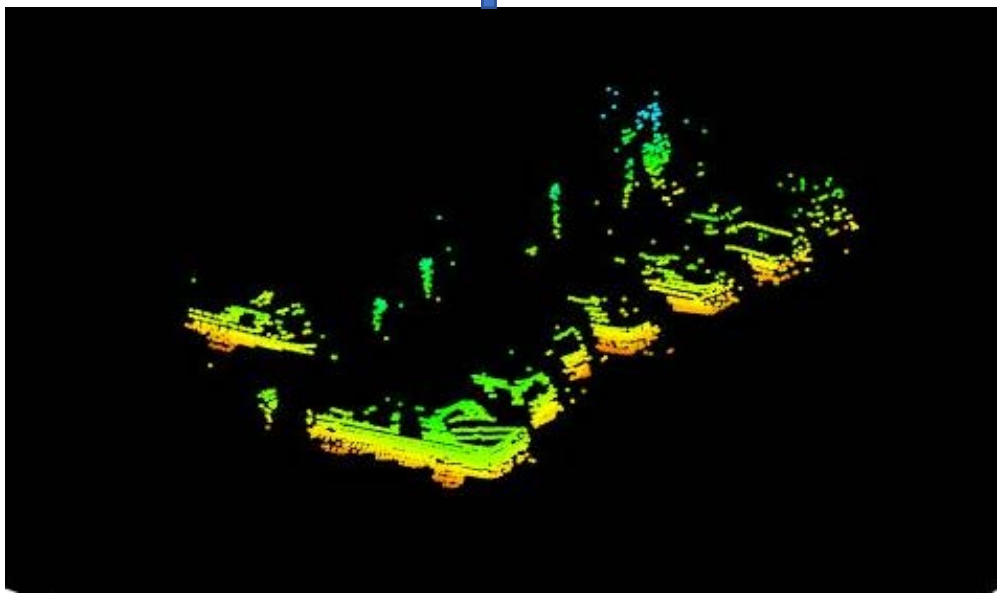
```
1: clusters = ∅
2: for p ∈ P do
3:     Q = ∅
4:     if p.status ≠ processed then
5:         Q.add(p)
6:         for q ∈ Q do
7:             q.status = 'processed'
8:             N = GetNeighbors(q, ε)
9:             Q.addAll(N)
10:        if size(Q) ≥ minPts then clusters.add(Q)
```

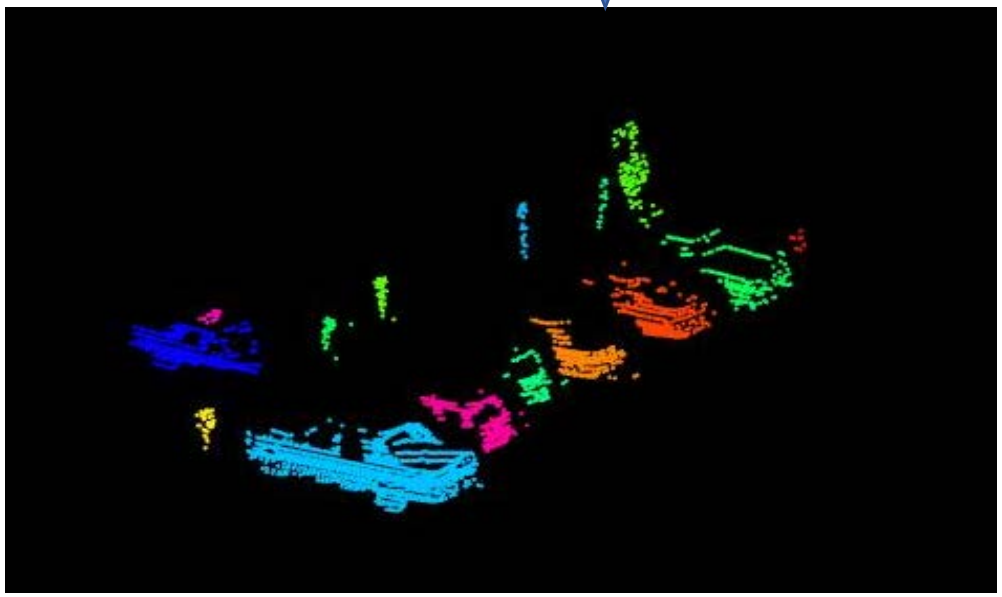
$clusters$
$= \{C_1(9,6,7,16,8,17), C_2(4,5,3,11,12,2,13,10,15,14,1), C_3(19,20,18)\}$

$Q = \{\}$



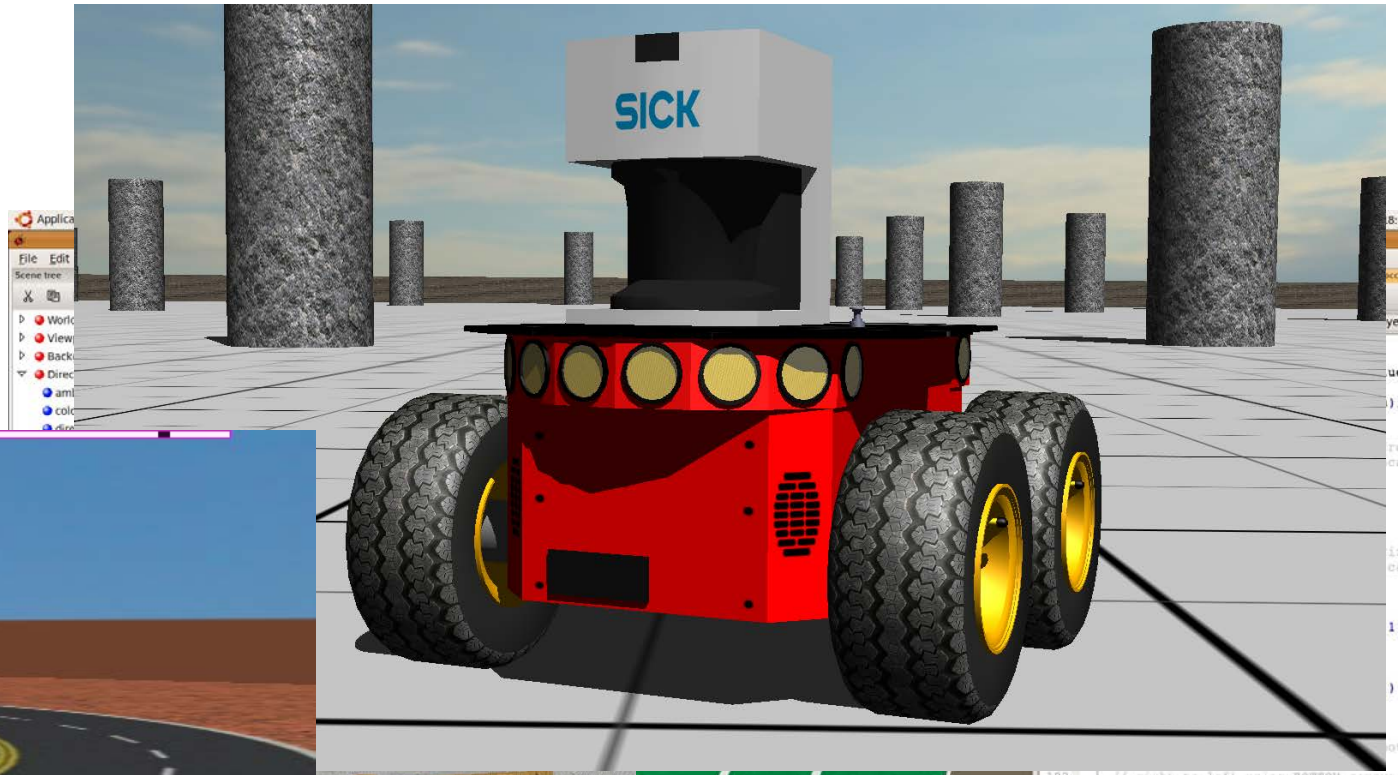
Euclidean Cluster Extraction

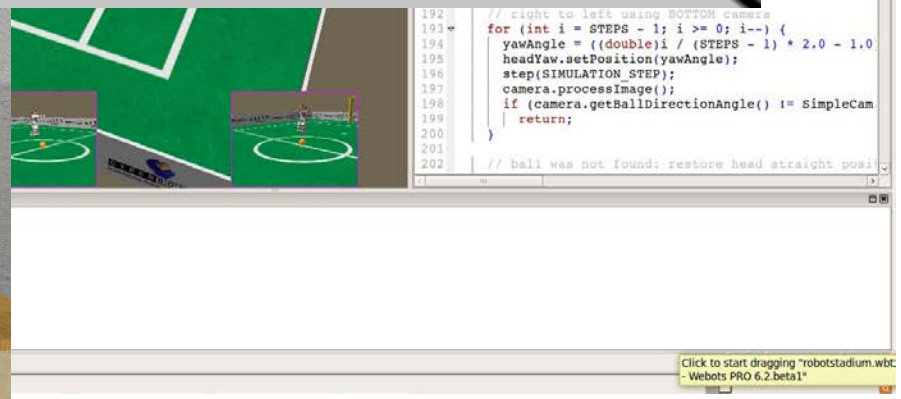


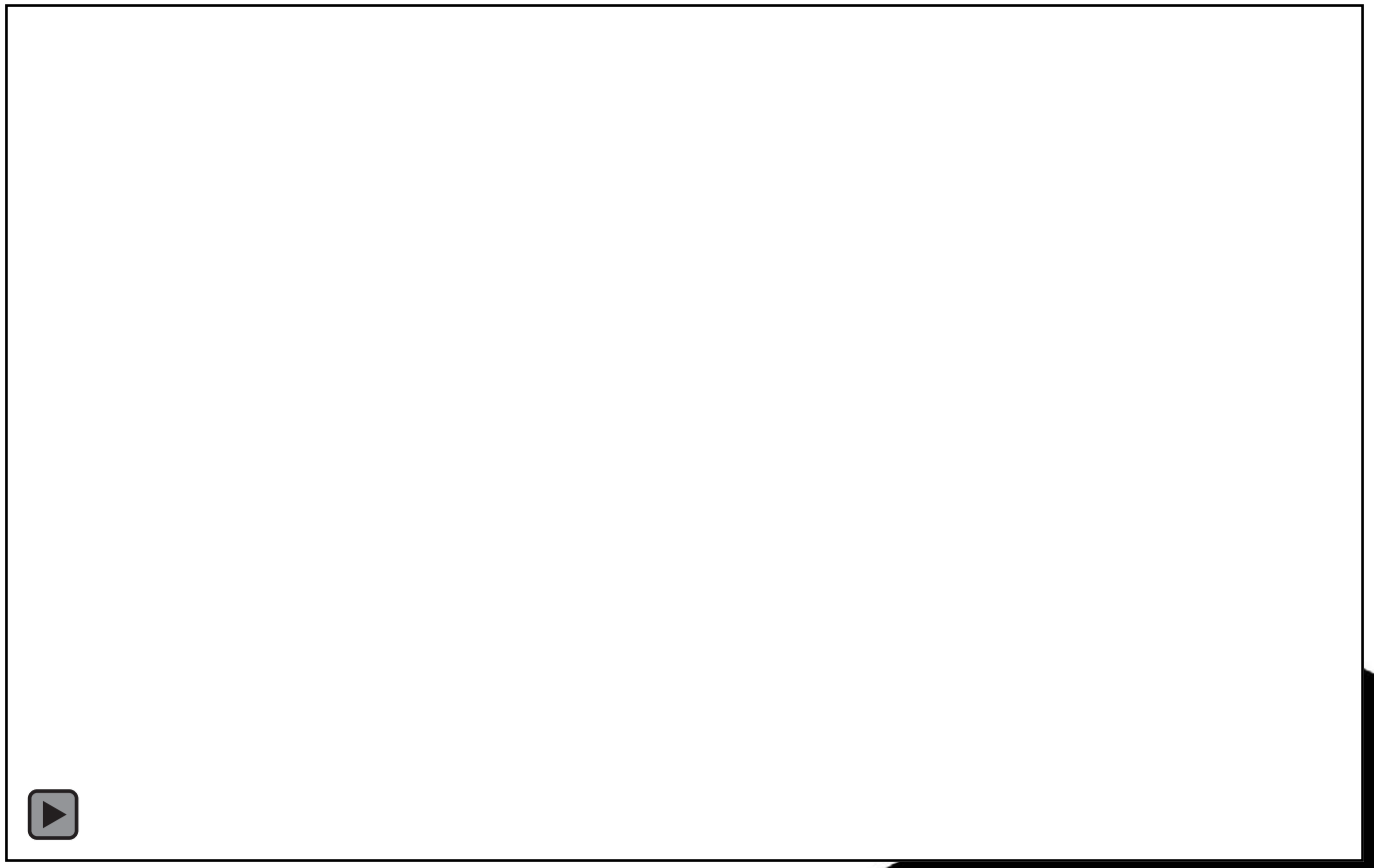


Webots Simulator

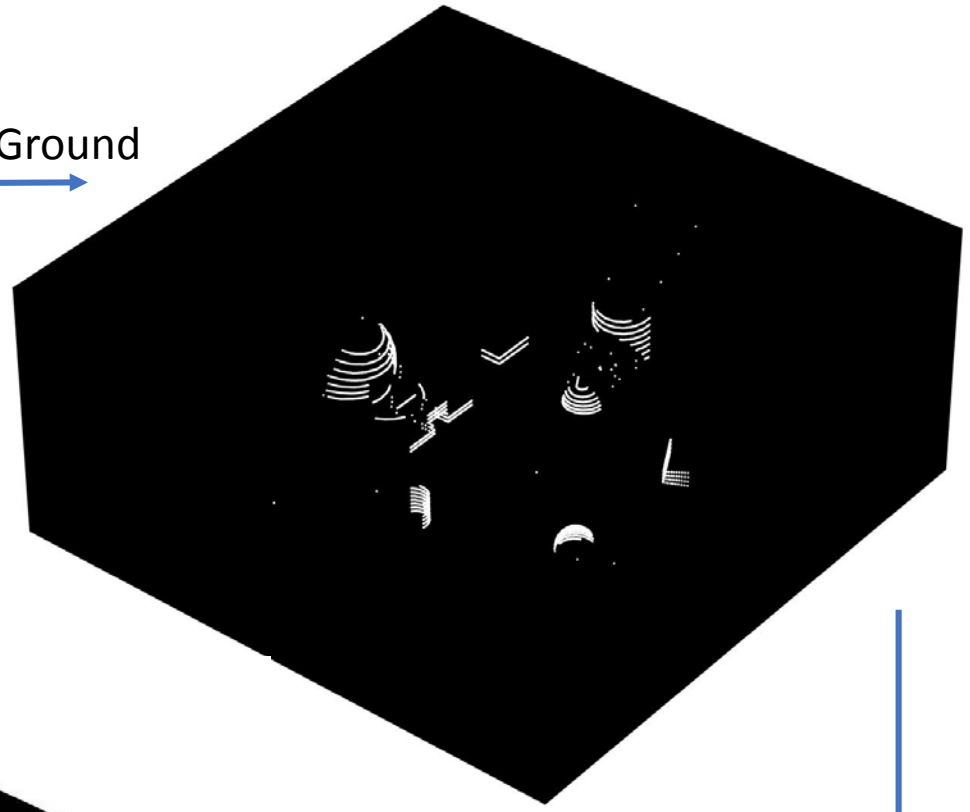
Webots Simulator

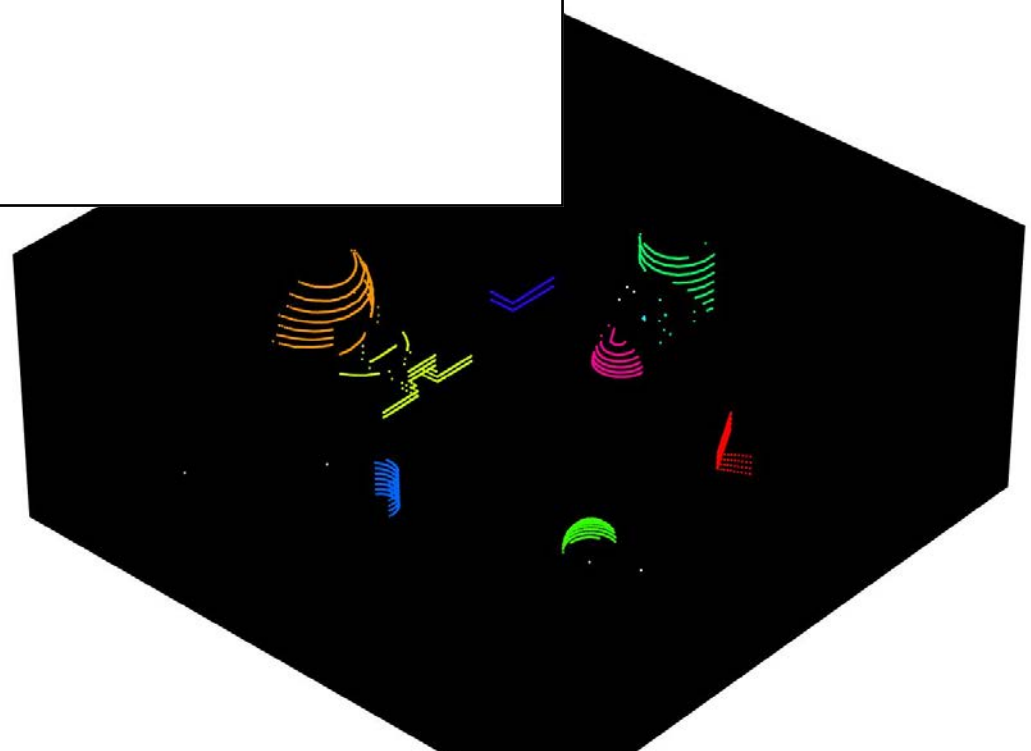






Remove Ground





Clustering

Use Cases

Use cases

- Autonomous vehicles
  - Collision avoidance
  - Object detections
- Manufacturing
  - Protect a prohibited area
  - Alert an emergency situation