

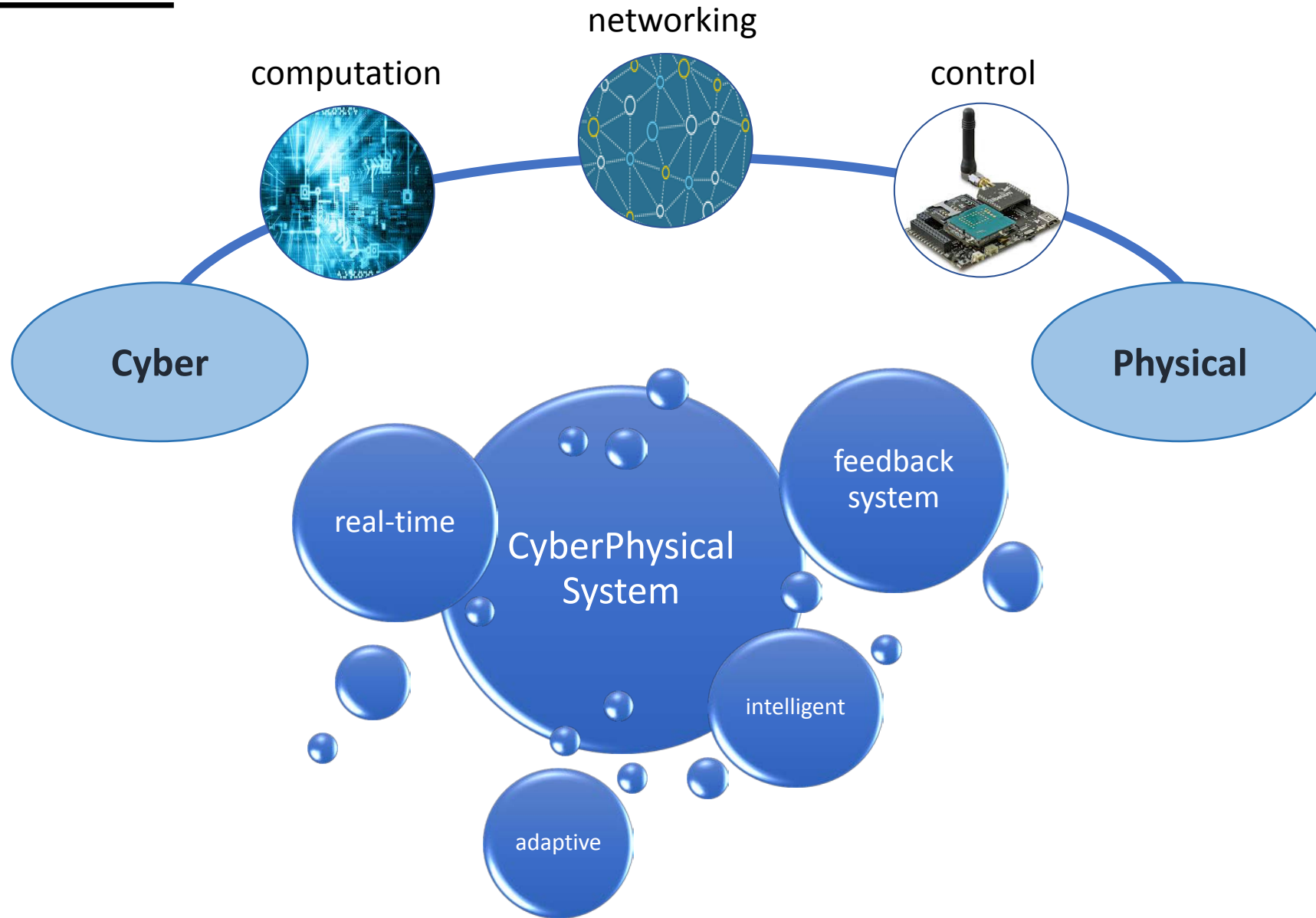


LiDAR Data Processing For Automated Environments

Hannaneh Najdataei

.: October 2017 .:

Introduction



CPS Applications



Health care



Energy

Infrastructure



Manufacturing

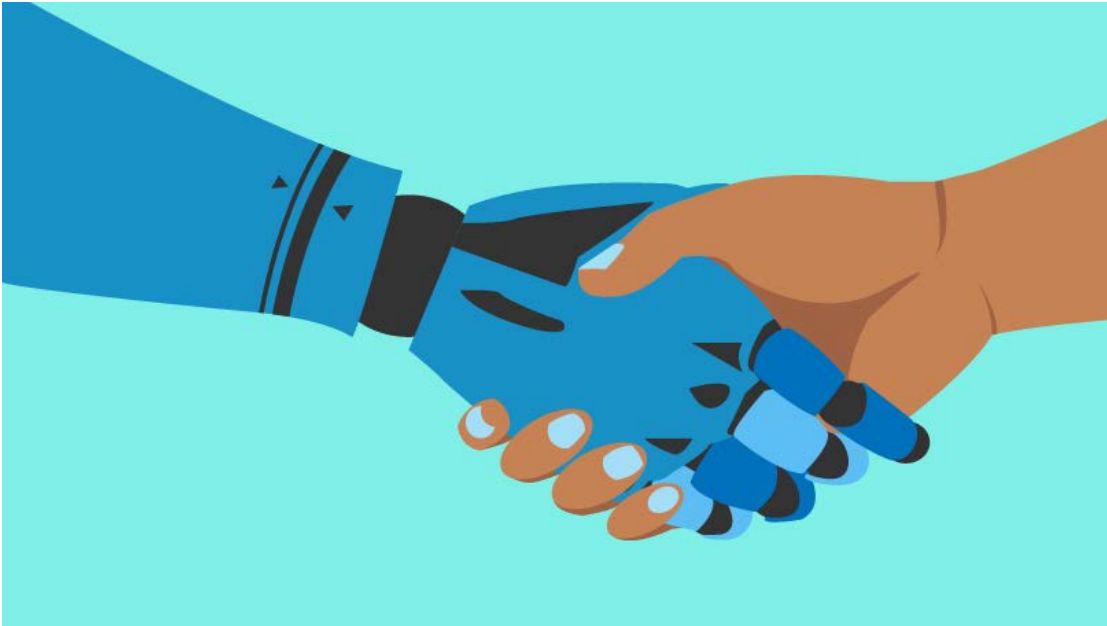


Robotics

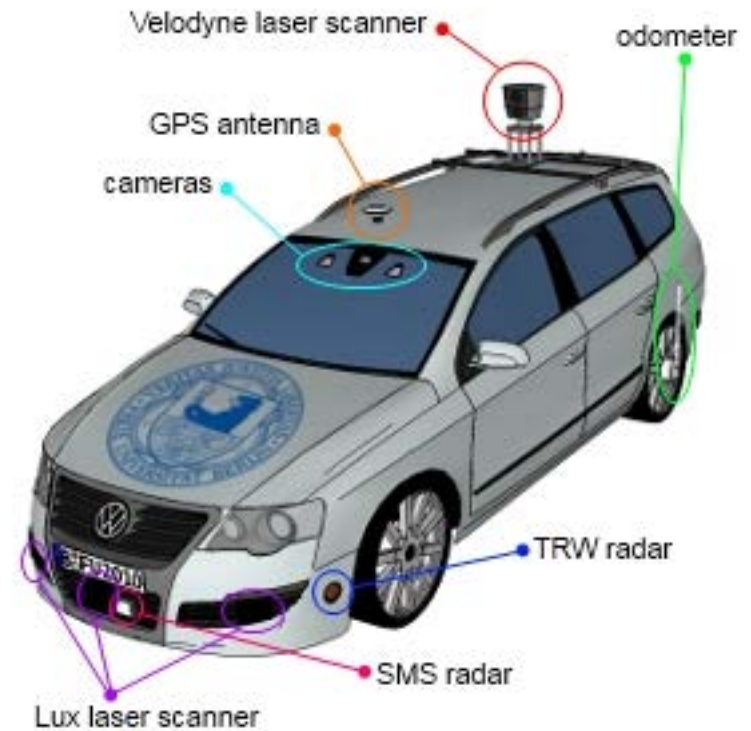


Transportation

Automated Environment



✓ Makes life easier! and safer



Pic: robotland

Outline

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

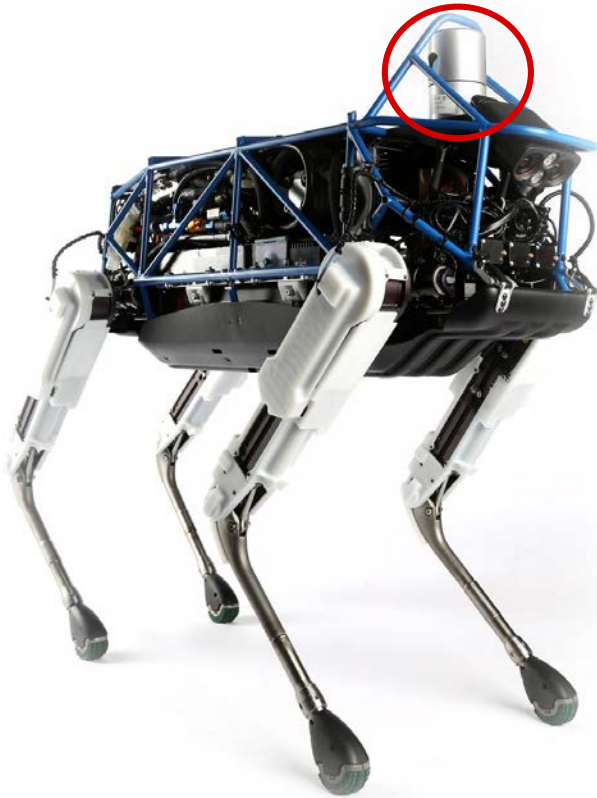


LiDAR and Point Cloud

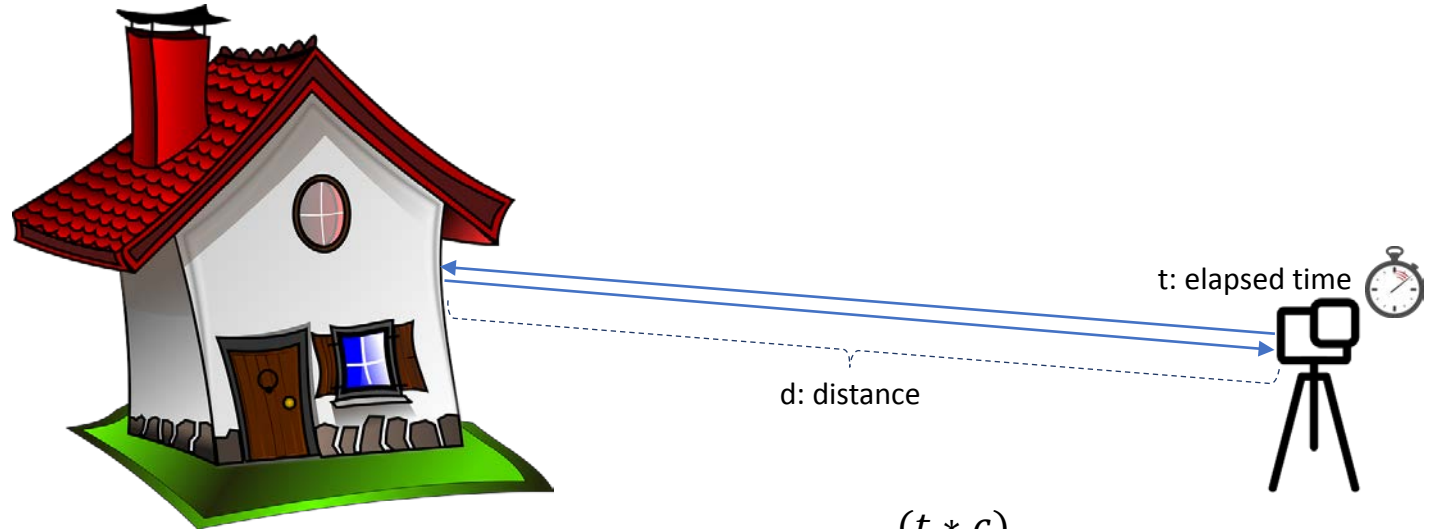
Introducing Spot



LiDAR

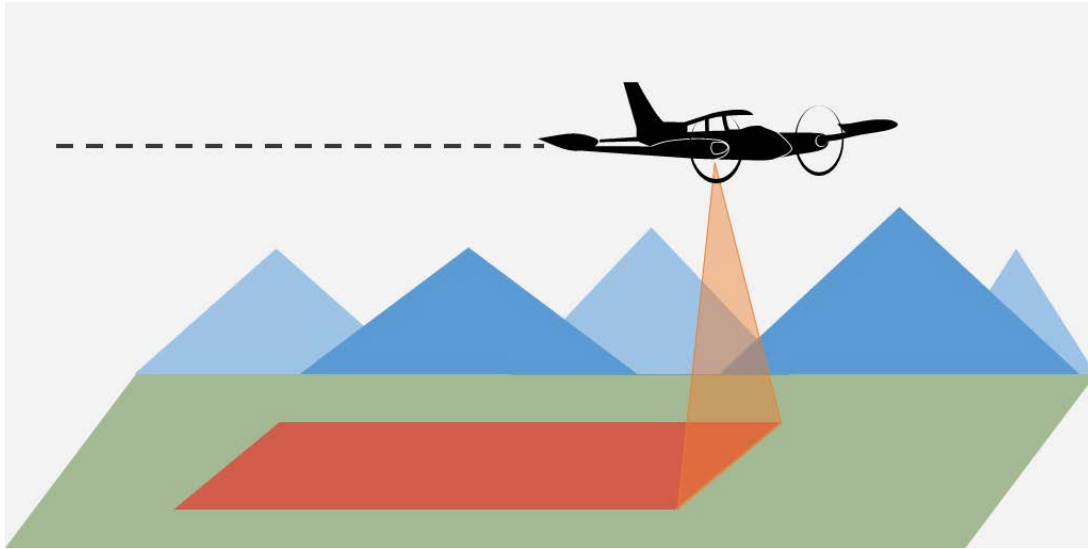


- Portmanteau word of *light* and *radar*
- Stands for Light Detection And Ranging

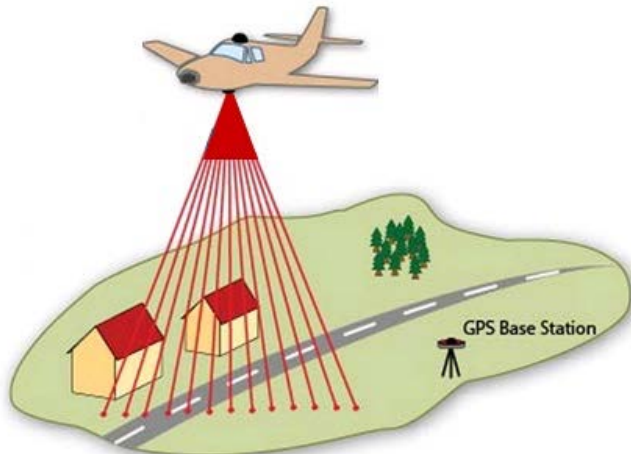


$$d = \frac{(t * c)}{2}$$

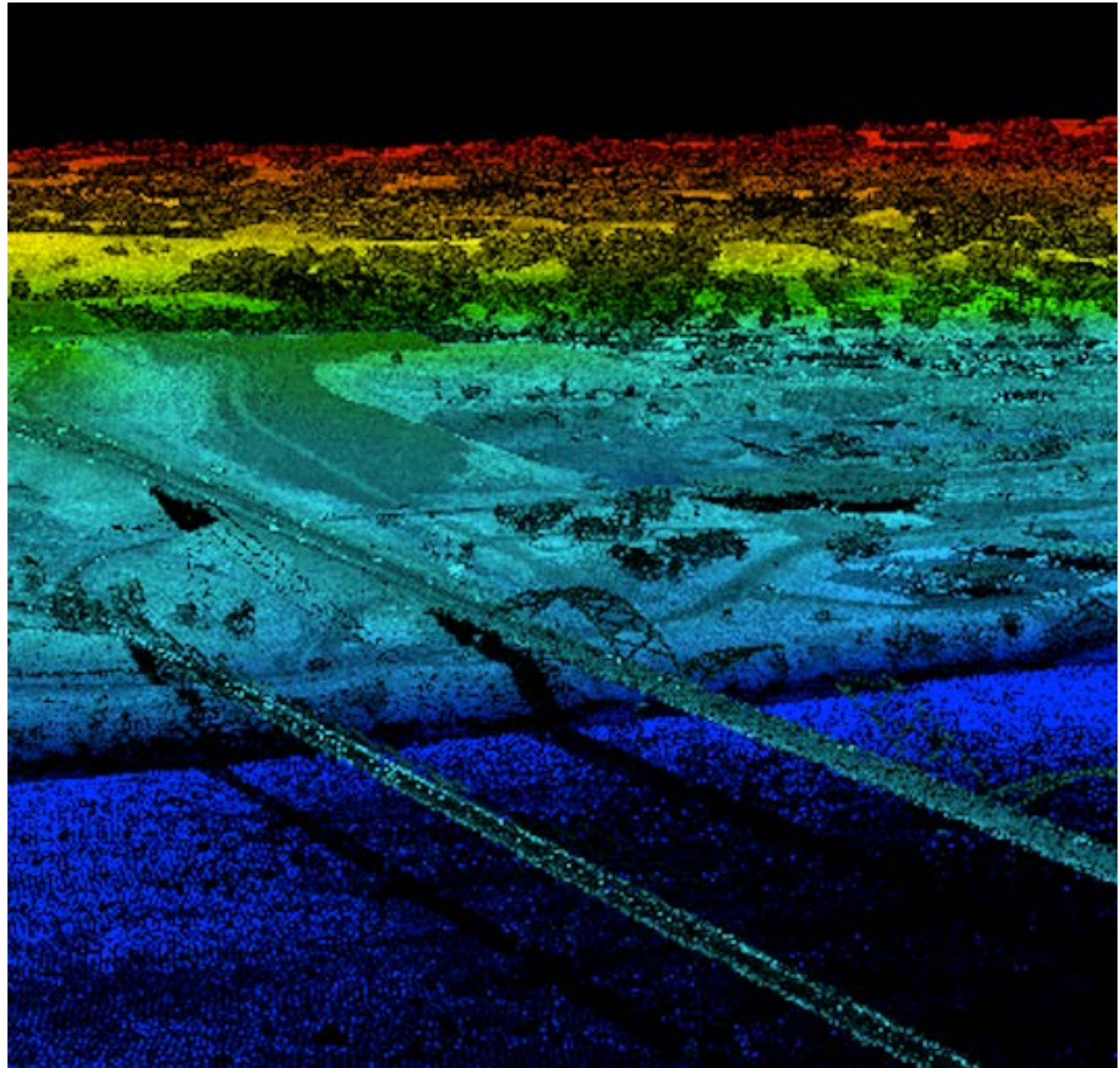
Airborne LiDAR



Airborne LiDAR – 3D mapping



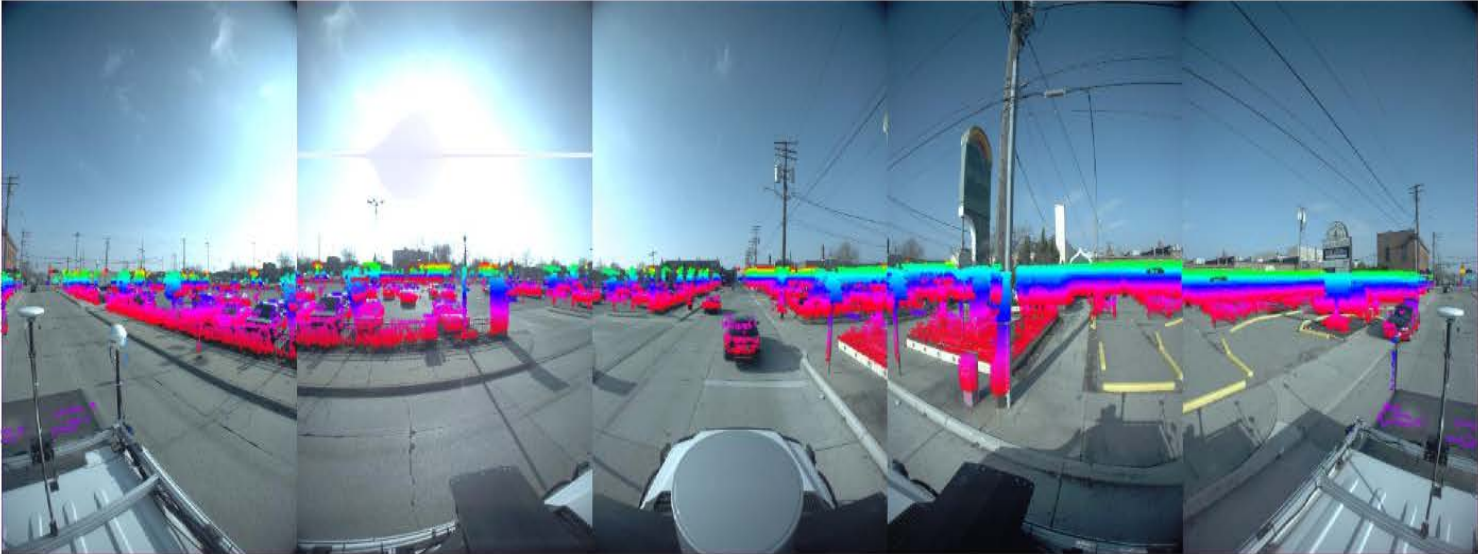
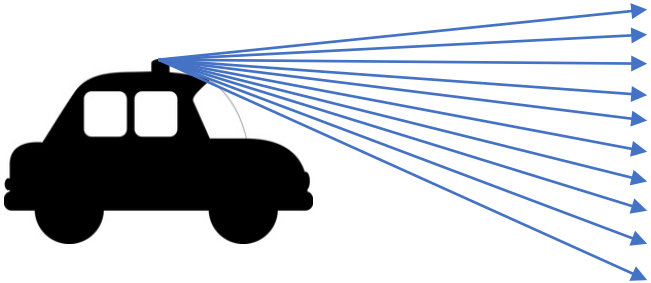
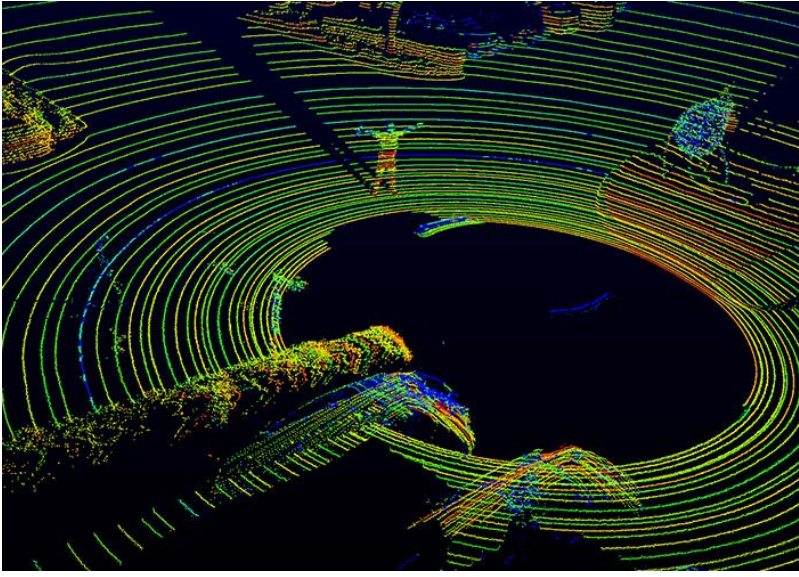
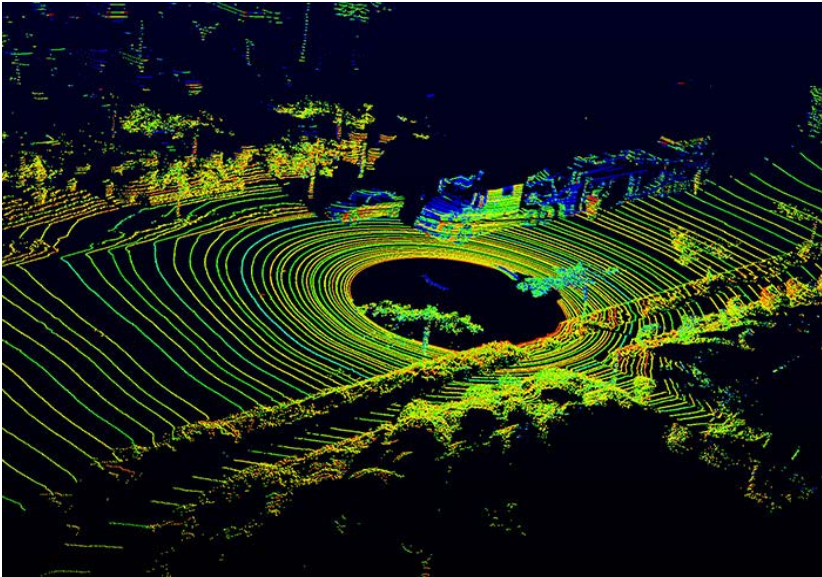
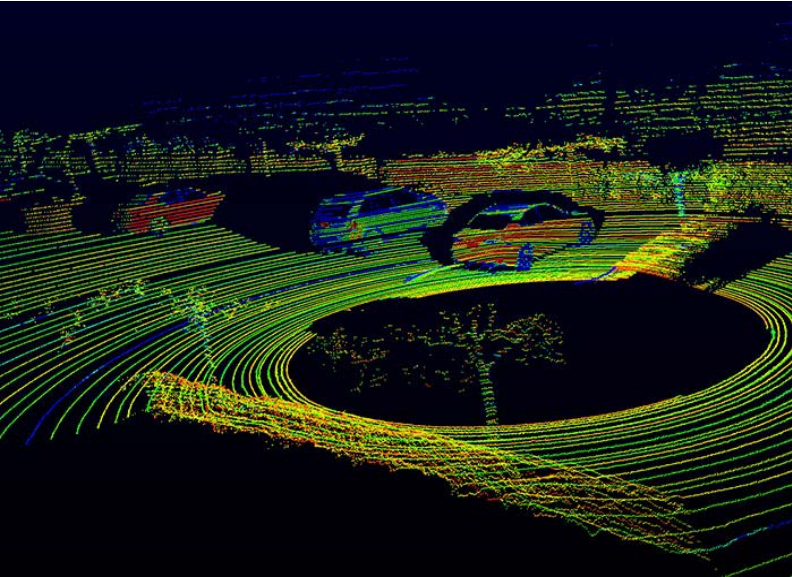
http://amv.cast.usark.edu/wp-content/uploads/2013/03/ALS_schematic.jpg



Pic: airborneimaginginc.com

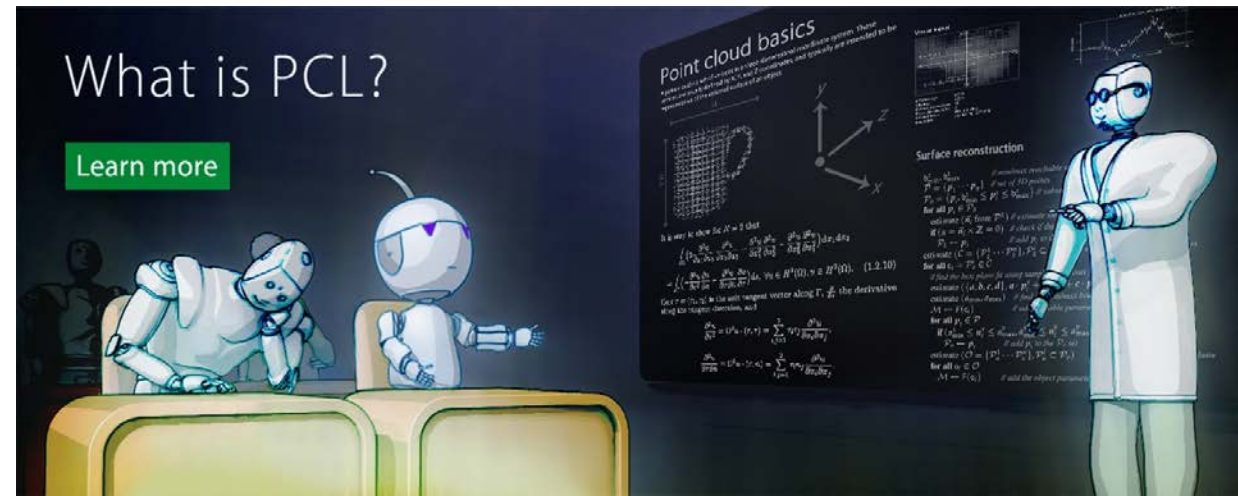
Rotating LiDAR

Pic: point cloud library



The background features a white central area with dark grey and light grey geometric shapes in the top-left and bottom-left corners, and a yellow geometric shape in the bottom-right corner.

Point Cloud Library



PCL features

[Learn more](#)

Initial point cloud data

Filtering

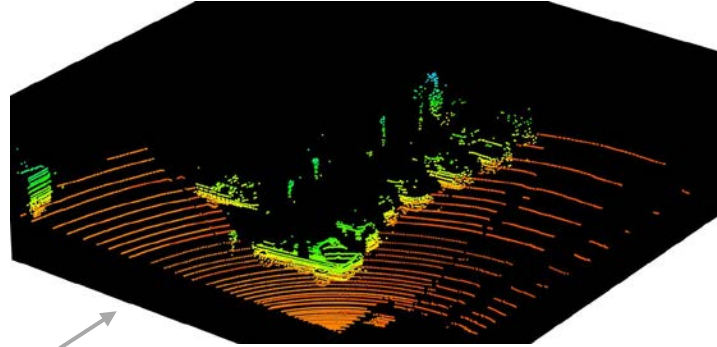
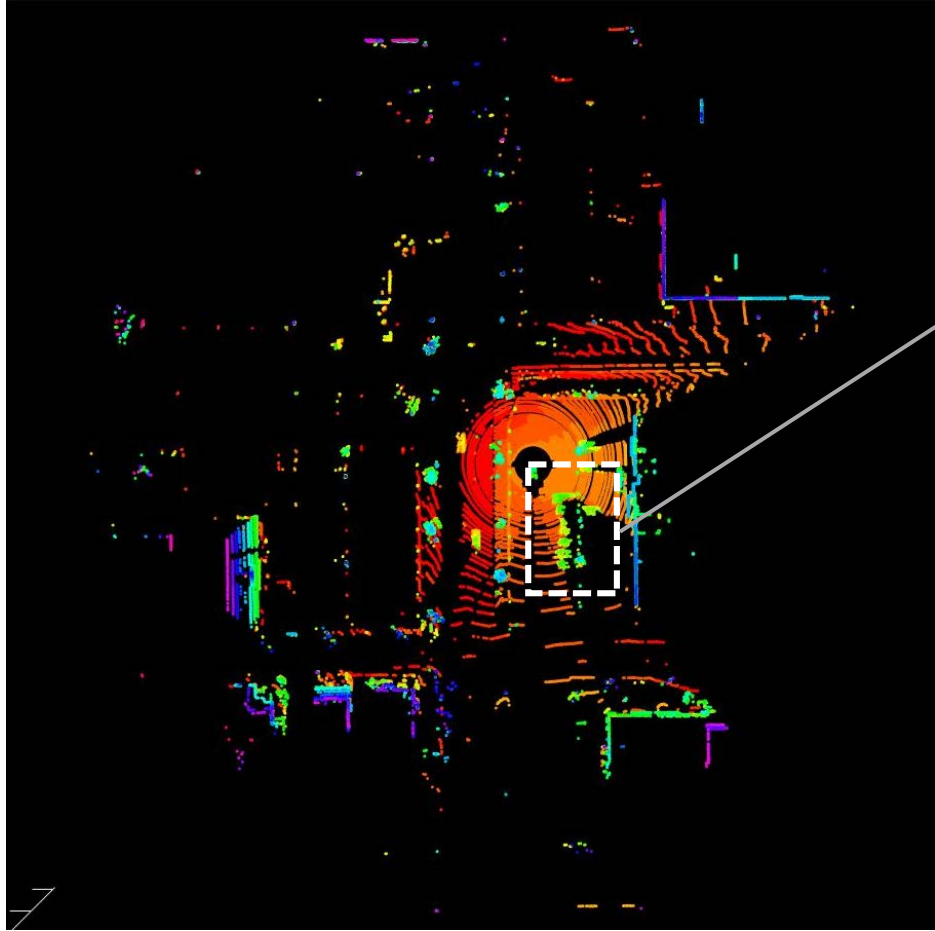
Segmentation

Surface reconstruction

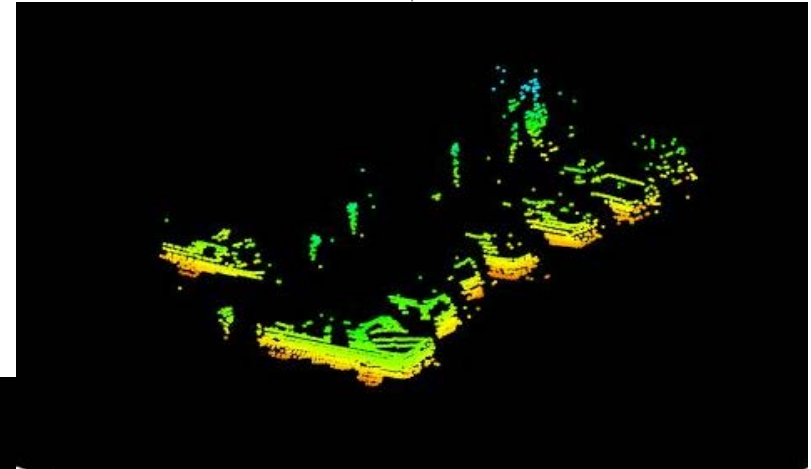
Model fitting



PCL - Segmentation



Removing the Ground



Clustering



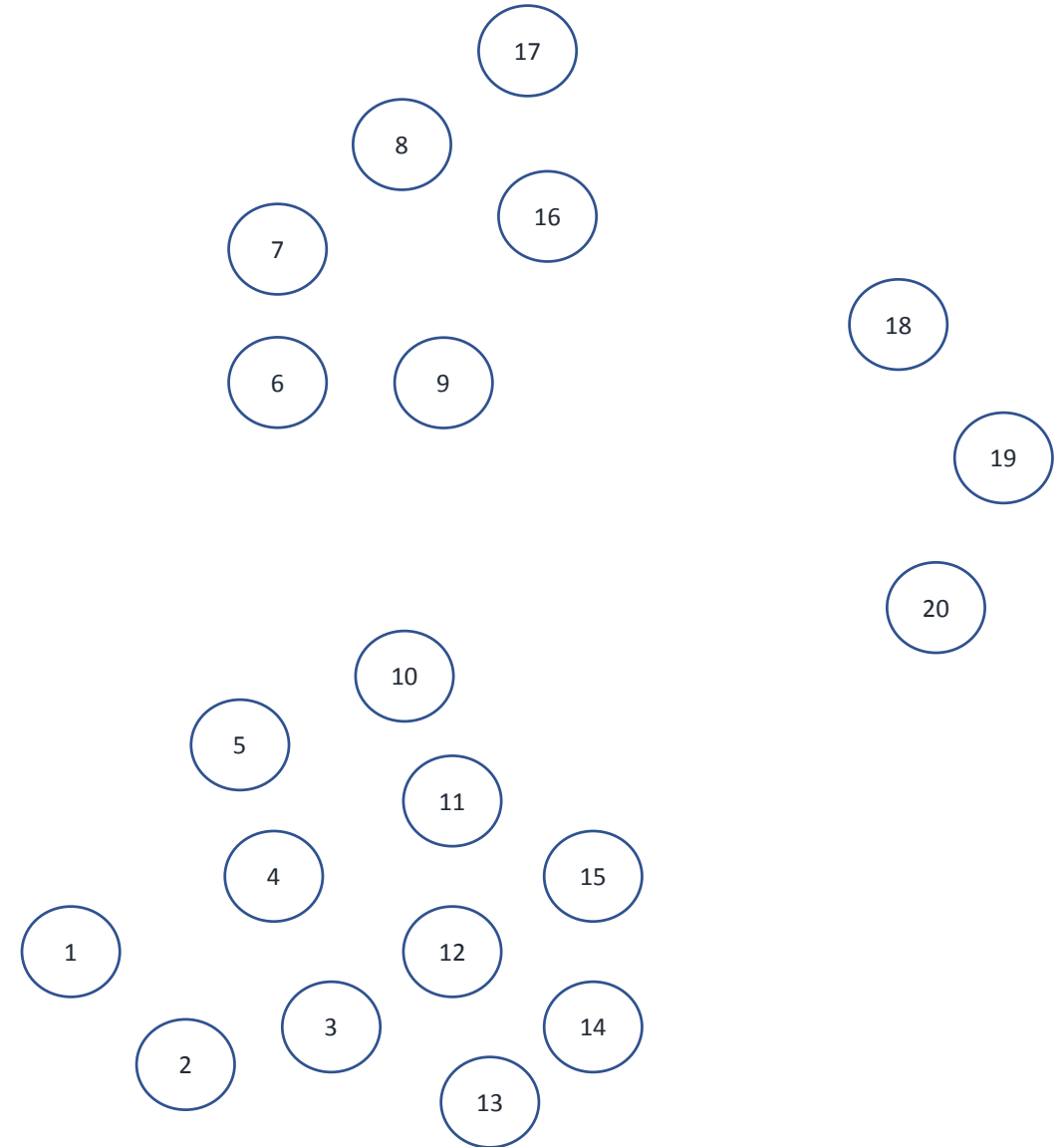
Euclidean Cluster Extraction

PCL – EclideanClusterExtraction(eps, minPts)

```
1: clusters =  $\emptyset$ 
2: for  $p \in P$  do
3:   Q =  $\emptyset$ 
4:   if  $p.status \neq processed$  then
5:     Q.add( $p$ )
6:     for  $q \in Q$  do
7:        $q.status = 'processed'$ 
8:        $N = GetNeighbors(q, \epsilon)$ 
9:       Q.addAll( $N$ )
10:    if  $size(Q) \geq minPts$  then clusters.add(Q)
```

clusters = {}

Q = {}



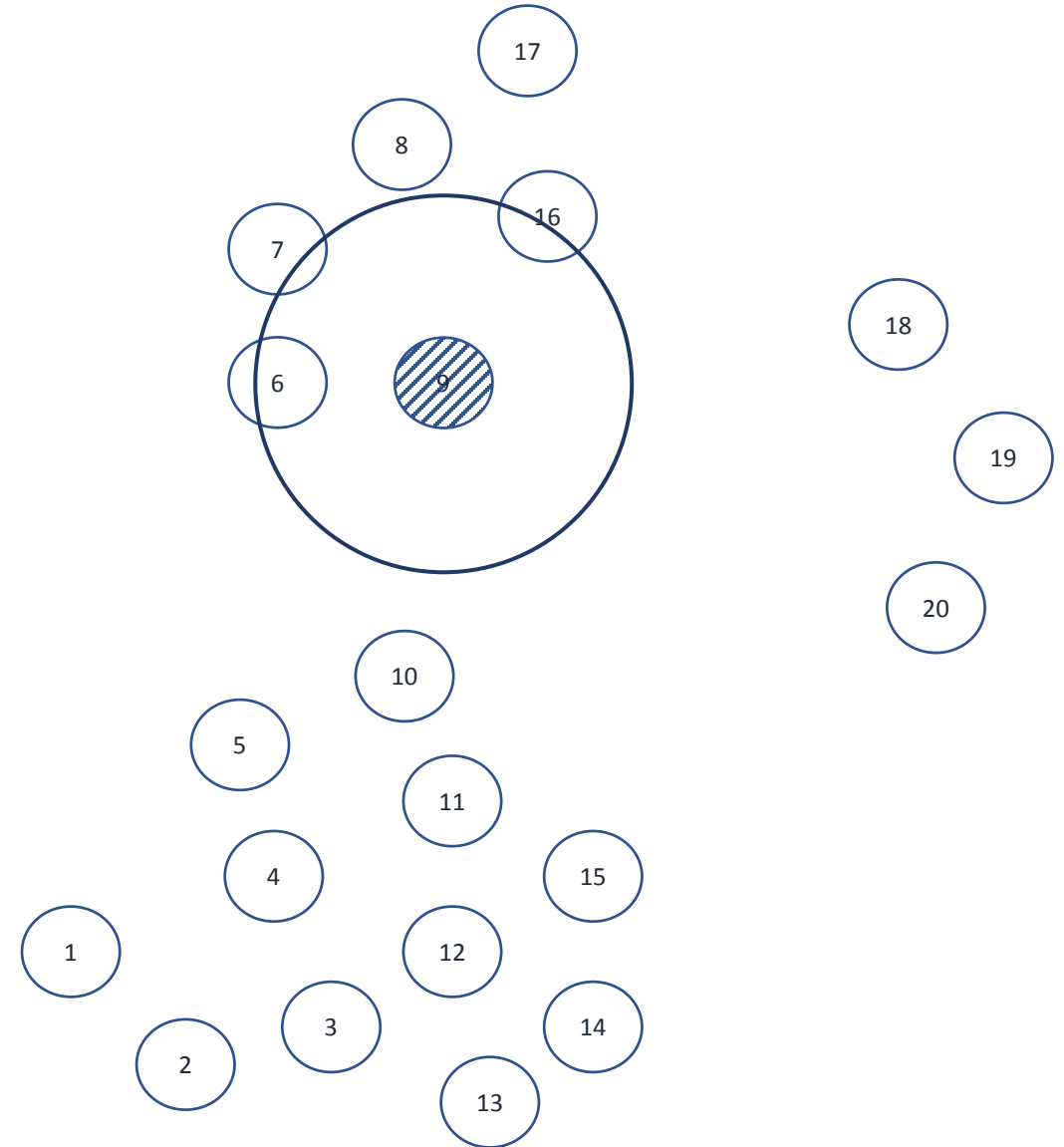
Euclidean Cluster Extraction

PCL – EclideanClusterExtraction(eps, minPts)

```
1: clusters =  $\emptyset$ 
2: for  $p \in P$  do
3:   Q =  $\emptyset$ 
4:   if  $p.status \neq processed$  then
5:     Q.add( $p$ )
6:     for  $q \in Q$  do
7:        $q.status = 'processed'$ 
8:        $N = GetNeighbors(q, \epsilon)$ 
9:       Q.addAll( $N$ )
10:  if  $size(Q) \geq minPts$  then clusters.add(Q)
```

$clusters = \{\}$

$Q = \{\emptyset\}$



Euclidean Cluster Extraction

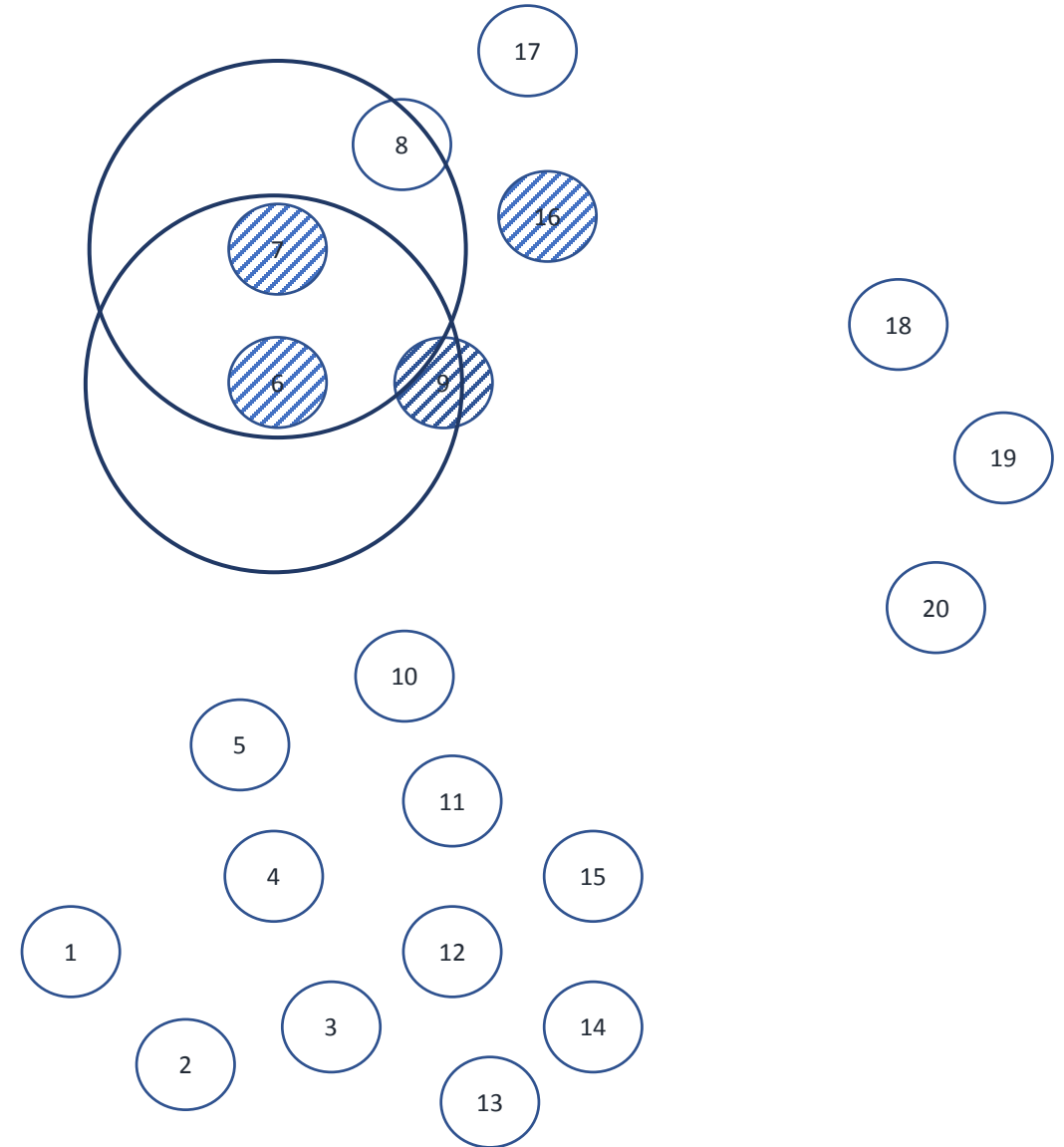
PCL – EclideanClusterExtraction(eps, minPts)

```
1: clusters =  $\emptyset$ 
2: for  $p \in P$  do
3:   Q =  $\emptyset$ 
4:   if  $p.status \neq processed$  then
5:     Q.add( $p$ )
6:     for  $q \in Q$  do
7:        $q.status = 'processed'$ 
8:        $N = GetNeighbors(q, \epsilon)$ 
9:       Q.addAll( $N$ )
10:  if  $size(Q) \geq minPts$  then clusters.add(Q)
```

$clusters = \{ \}$

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

↑↑↑



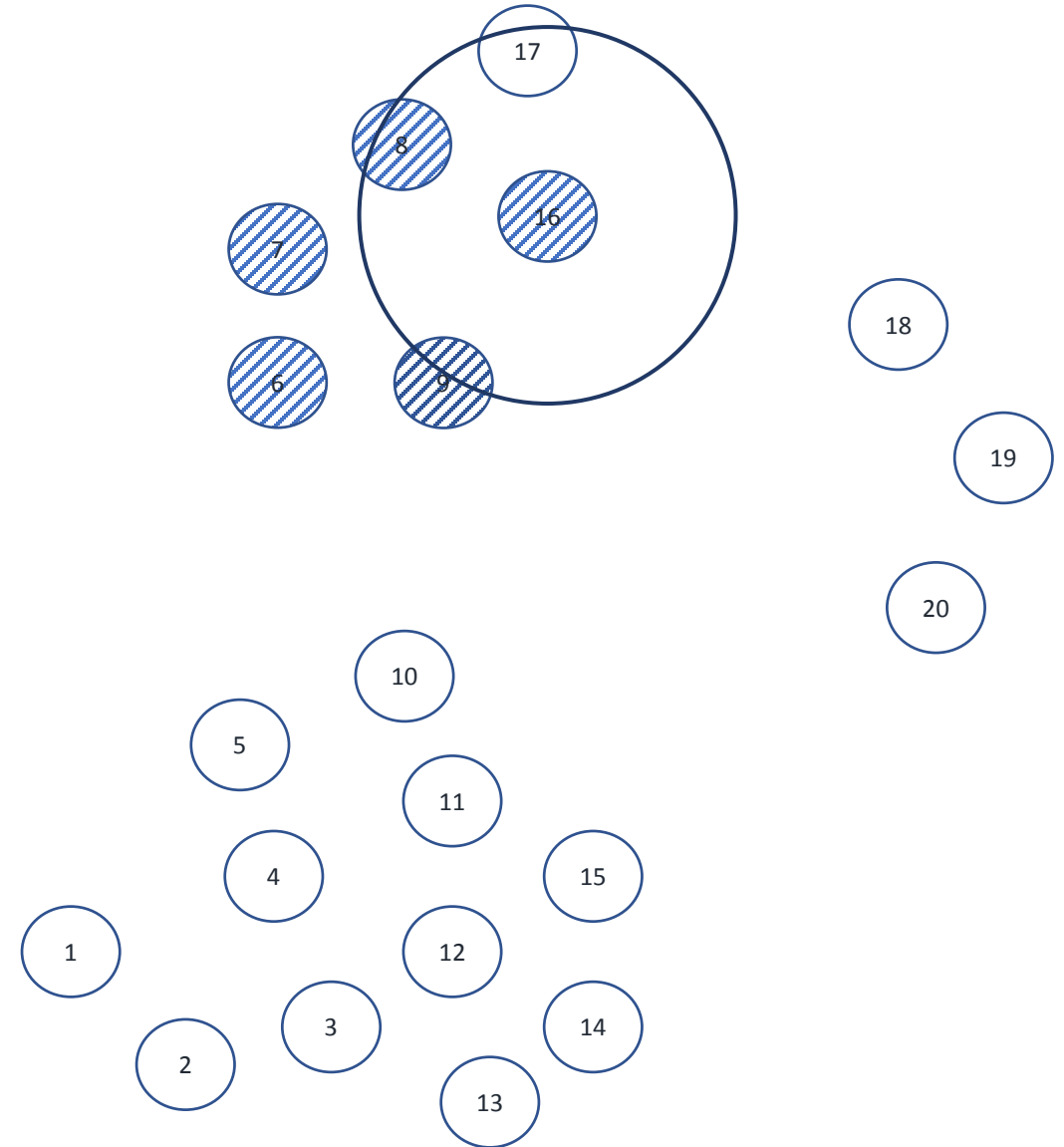
Euclidean Cluster Extraction

PCL – EclideanClusterExtraction(eps, minPts)

```
1: clusters =  $\emptyset$ 
2: for  $p \in P$  do
3:   Q =  $\emptyset$ 
4:   if  $p.status \neq processed$  then
5:     Q.add( $p$ )
6:     for  $q \in Q$  do
7:        $q.status = 'processed'$ 
8:        $N = GetNeighbors(q, \epsilon)$ 
9:       Q.addAll( $N$ )
10:  if  $size(Q) \geq minPts$  then clusters.add(Q)
```

$clusters = \{\}$

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



Euclidean Cluster Extraction

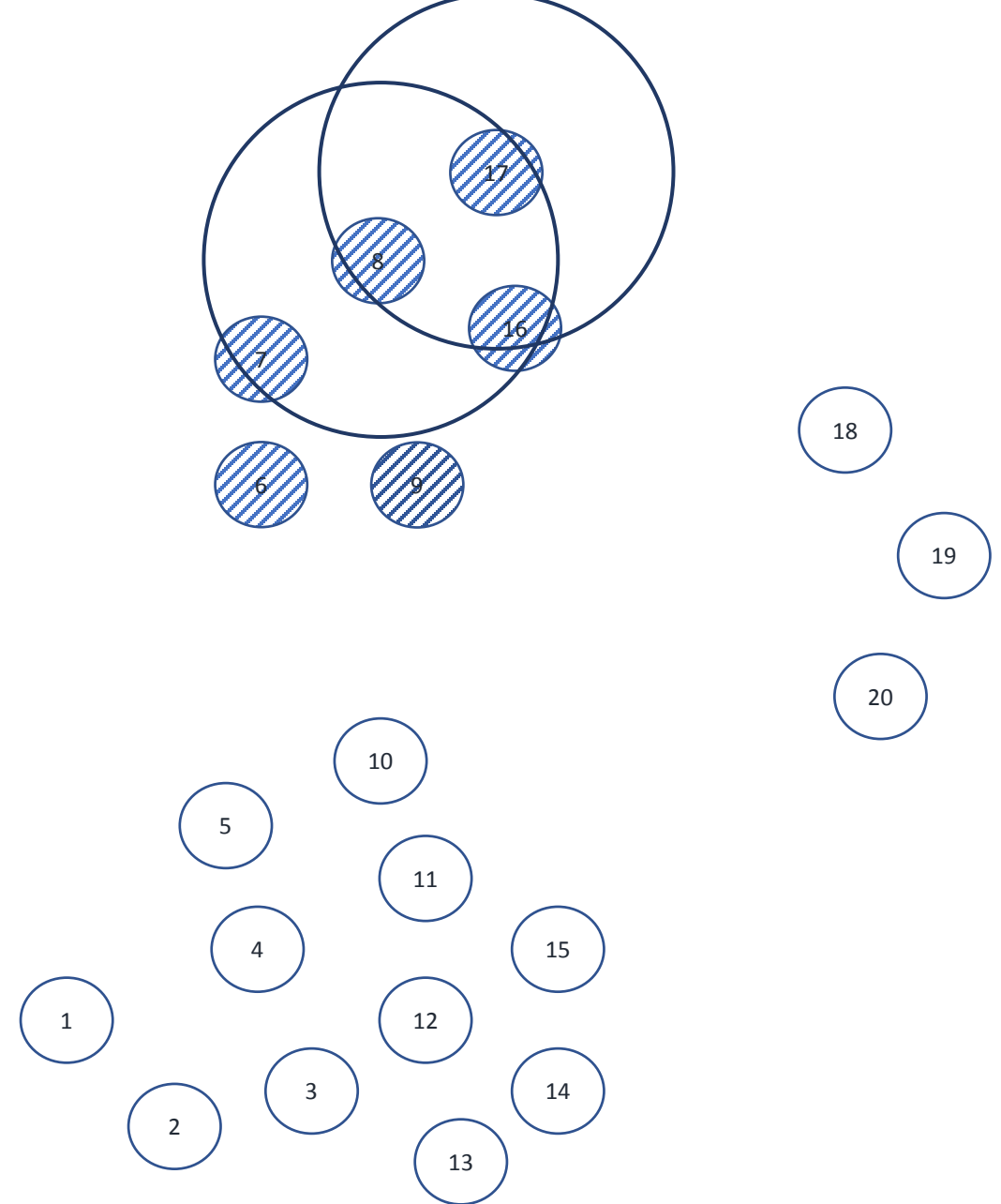
PCL – EclideanClusterExtraction(eps, minPts)

```
1: clusters =  $\emptyset$ 
2: for  $p \in P$  do
3:   Q =  $\emptyset$ 
4:   if  $p.status \neq processed$  then
5:     Q.add( $p$ )
6:     for  $q \in Q$  do
7:        $q.status = 'processed'$ 
8:        $N = GetNeighbors(q, \epsilon)$ 
9:       Q.addAll( $N$ )
10:    if  $size(Q) \geq minPts$  then clusters.add(Q)
```

clusters = {}

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

↑ ↑ ↑



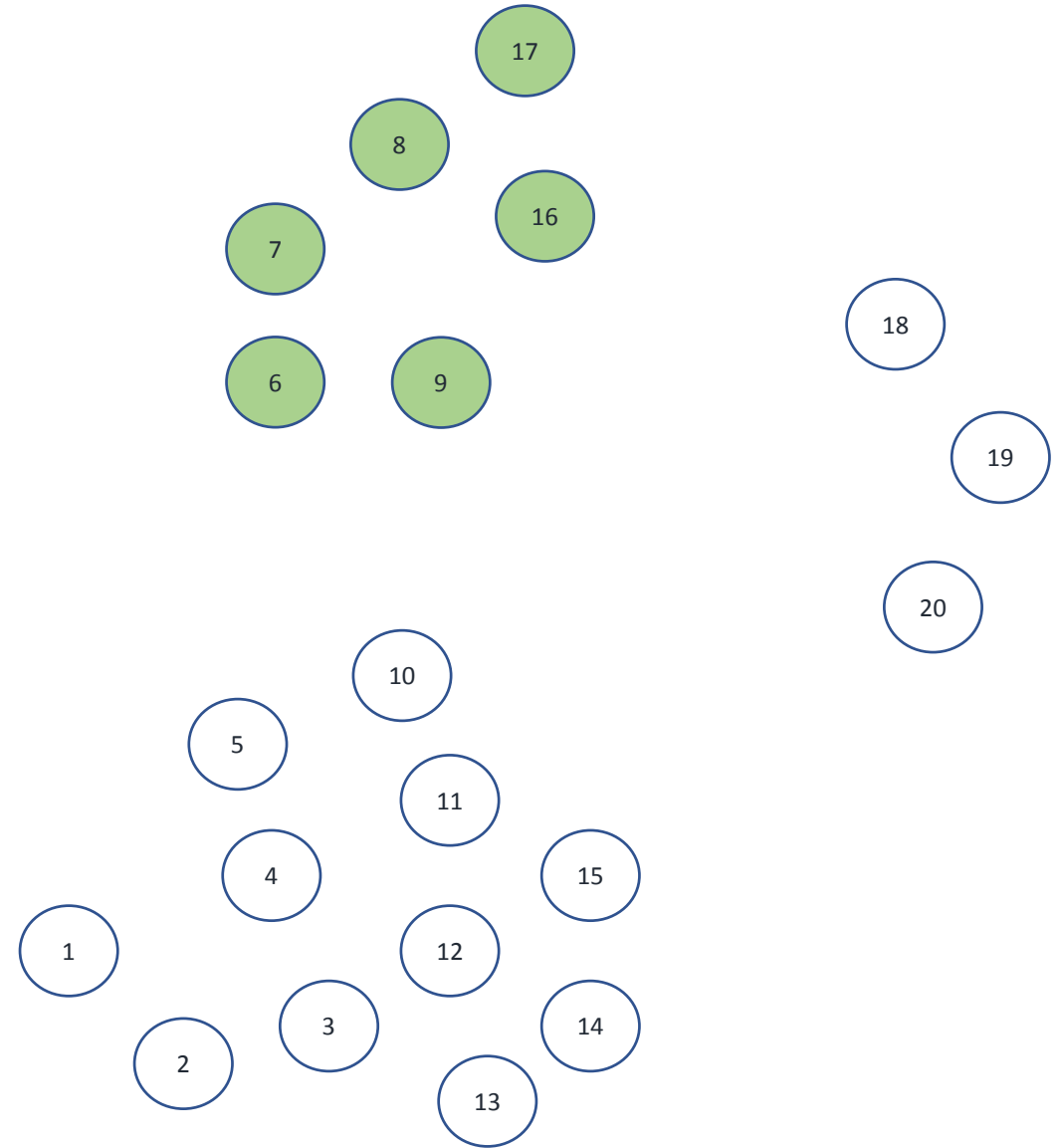
Euclidean Cluster Extraction

PCL – EclideanClusterExtraction(eps, minPts)

```
1: clusters =  $\emptyset$ 
2: for  $p \in P$  do
3:   Q =  $\emptyset$ 
4:   if  $p.status \neq processed$  then
5:     Q.add( $p$ )
6:     for  $q \in Q$  do
7:        $q.status = 'processed'$ 
8:        $N = GetNeighbors(q, \epsilon)$ 
9:       Q.addAll( $N$ )
10:    if  $size(Q) \geq minPts$  then clusters.add(Q)
```

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

$Q = \{\}$



Euclidean Cluster Extraction

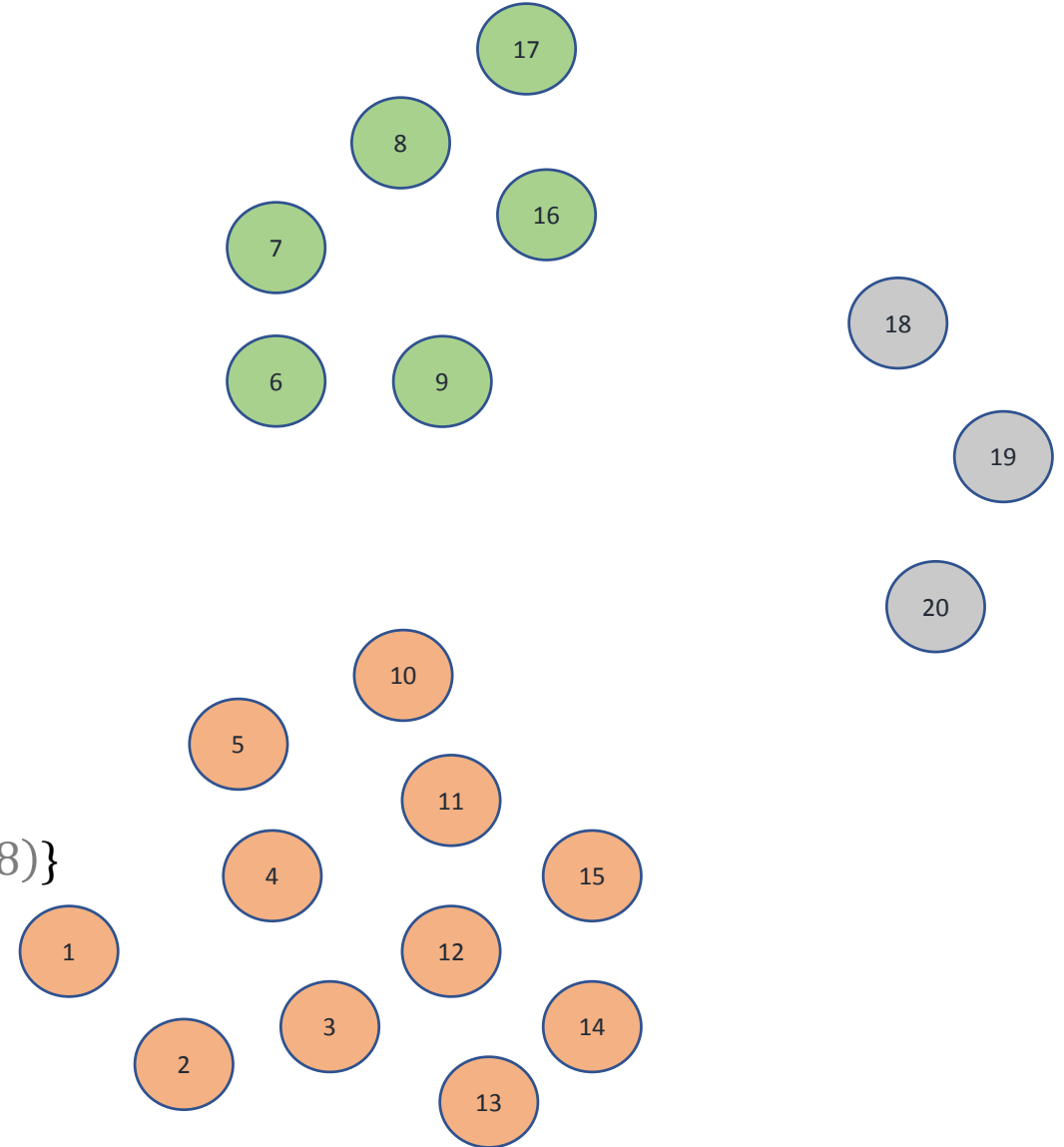
PCL – EclideanClusterExtraction(eps, minPts)

```
1: clusters =  $\emptyset$ 
2: for  $p \in P$  do
3:   Q =  $\emptyset$ 
4:   if  $p.status \neq processed$  then
5:     Q.add( $p$ )
6:     for  $q \in Q$  do
7:        $q.status = 'processed'$ 
8:        $N = GetNeighbors(q, \epsilon)$ 
9:       Q.addAll( $N$ )
10:    if  $size(Q) \geq minPts$  then clusters.add(Q)
```

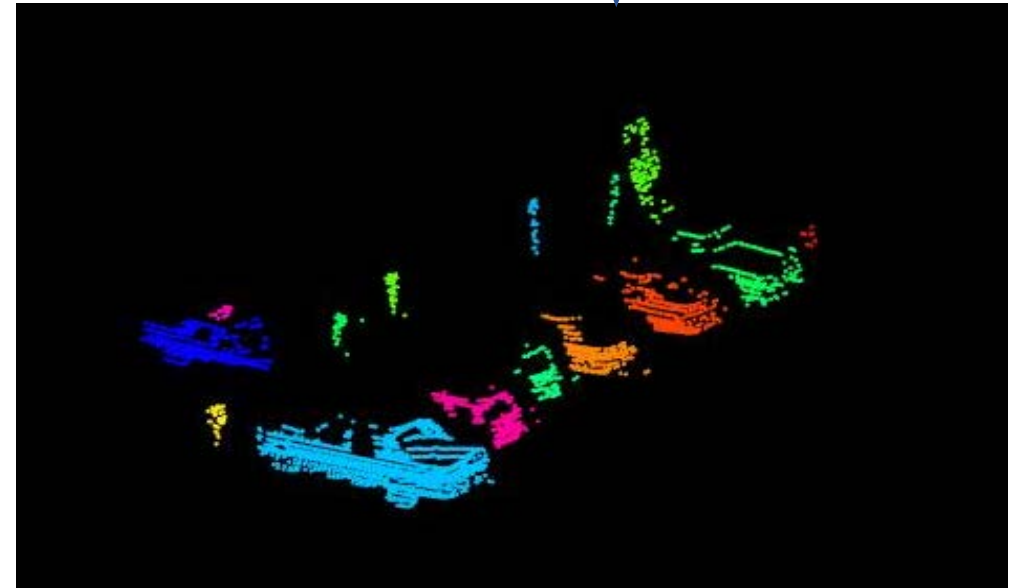
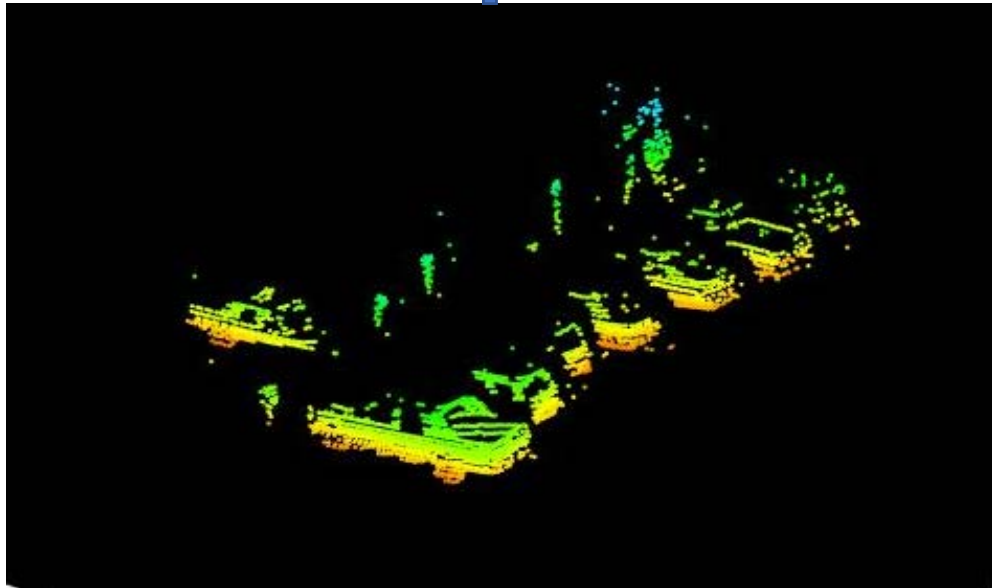
clusters

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

$Q = \{\}$



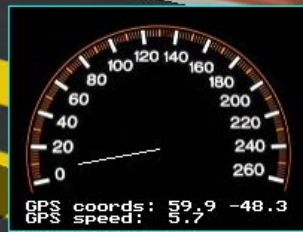
Euclidean Cluster Extraction





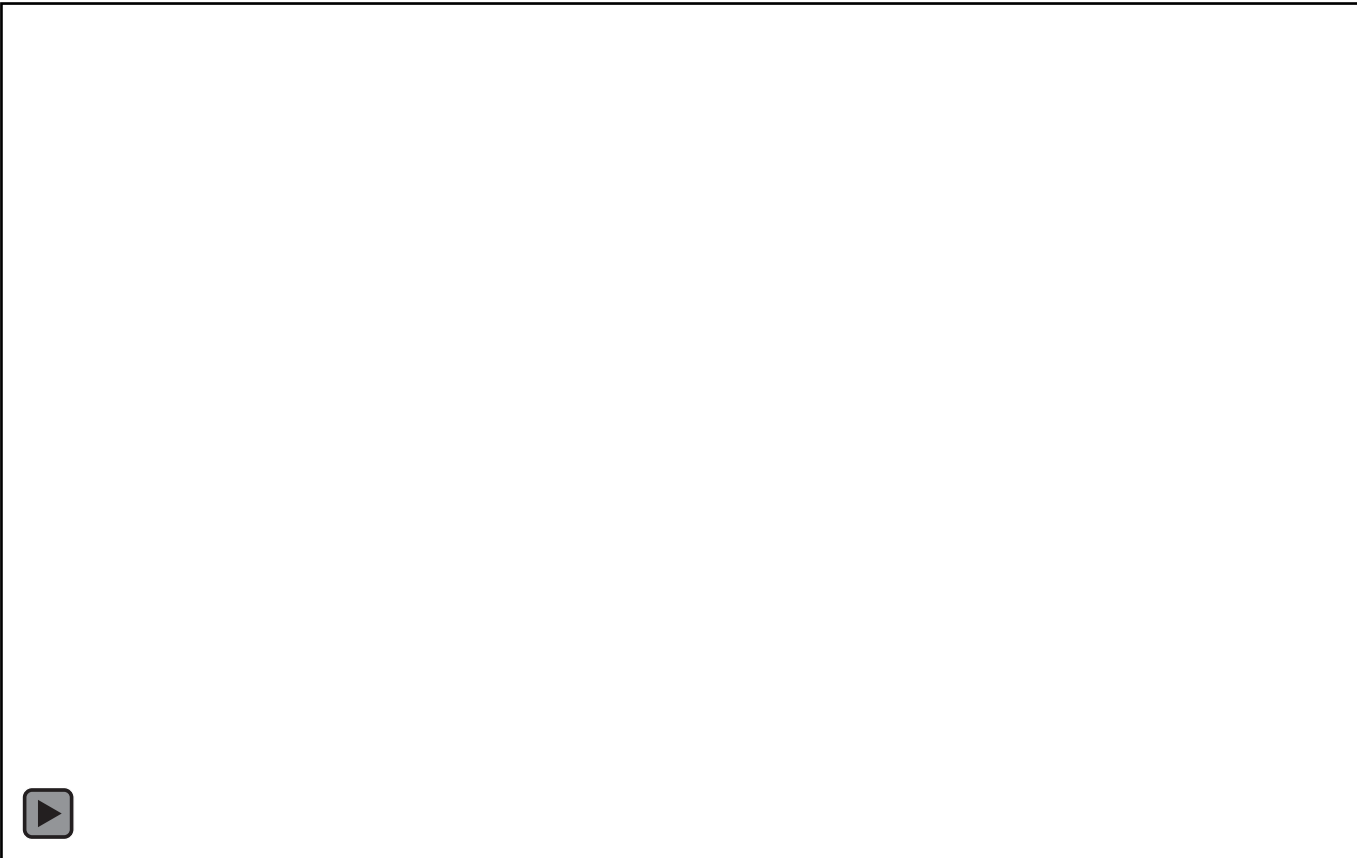
Webots Simulator

Webots Simulator

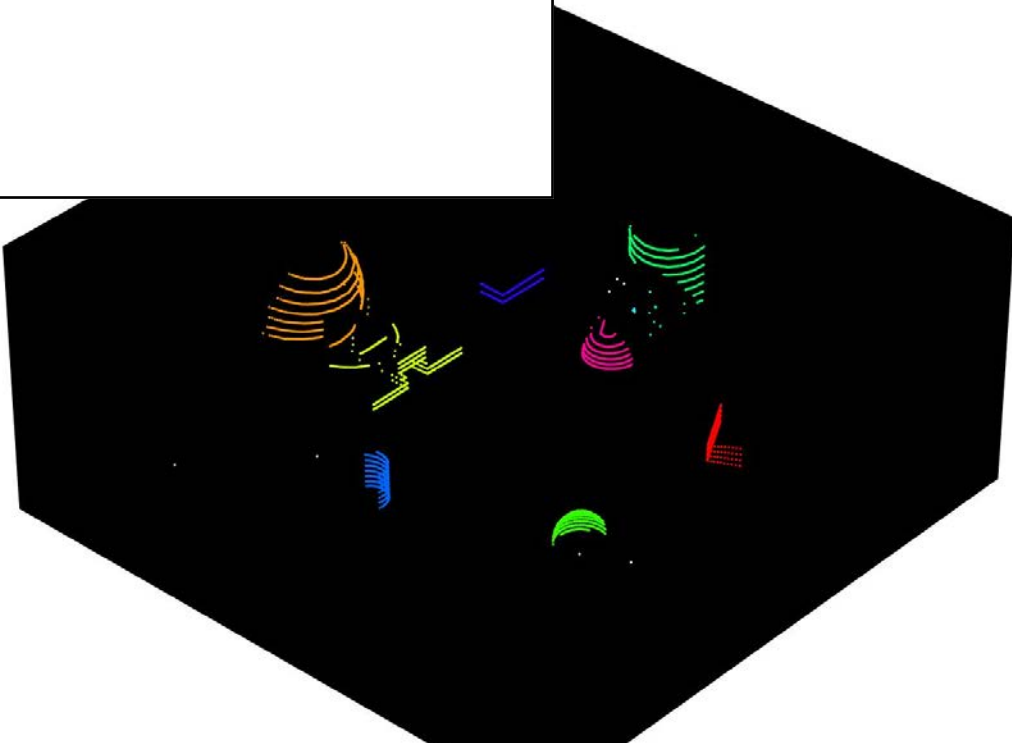
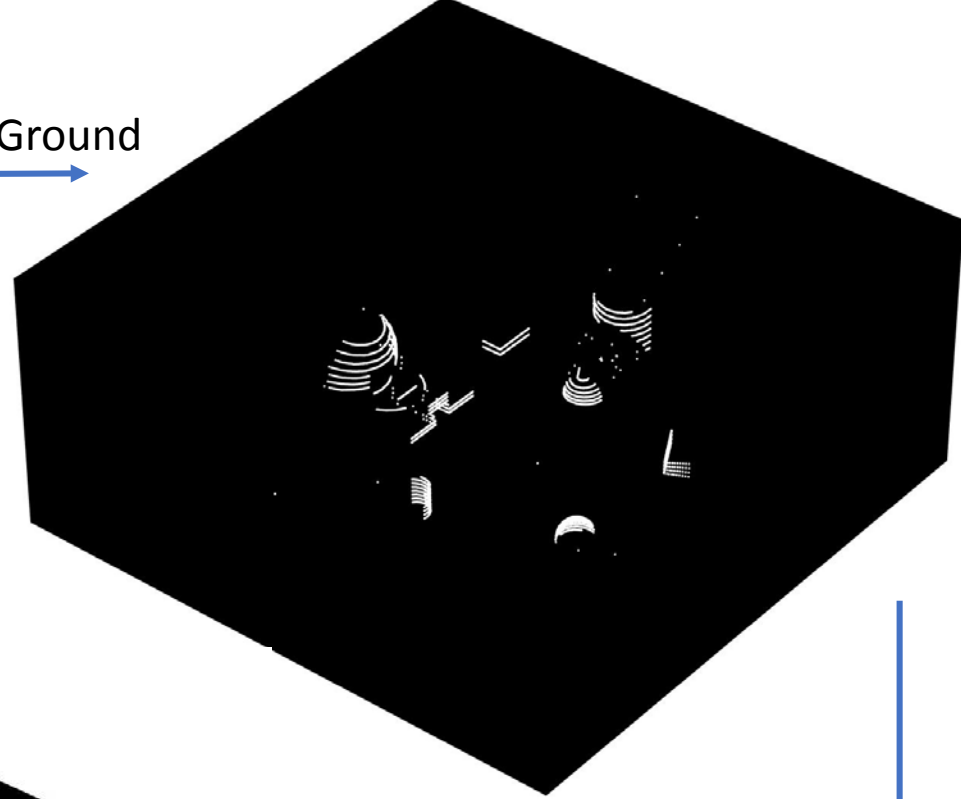


```
192 // right to left using BOTTOM camera
193 for (int i = STEPS - 1; i >= 0; i--) {
194     yawAngle = ((double)i / (STEPS - 1)) * 2.0 - 1.0;
195     headYaw.setPosition(yawAngle);
196     step(SIMULATION_STEP);
197     camera.processImage();
198     if (camera.getBallDirectionAngle() != SimpleCam
199         | return;
200 }
201 // ball was not found: restore head straight posi
202
```

Click to start dragging "robotstadium.wbt" - Webots PRO 6.2 beta1



Remove Ground



Clustering





Use Cases

Use cases

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