# MATLAB EXPO 2019

# LiDAR Processing for Automated Driving

Avinash Nehemiah Principal Product Manager Deep Learning, Computer Vision, Automated Driving

### **Complete Workflow for Lidar Processing in Automated Driving**



MATLAB EXPO 2019

MathWorks





# Why is lidar an essential sensor for automated driving ?

- 1. Accurate depth measurement (currently a function of radar)
- 2. 360-degrees of visibility (require multiple calibrated sensors to achieve)
- 3. Dense data to detect and classify objects (currently function of camera)

\*Automated driving use camera, radar and lidar as complementary sensors \* Robotics application use lidar alone



### **Talk Outline**





### **Talk Outline**





### **Reading Lidar Data from File or Sensor**



pcread PCD and PLY files rosbag ROS bag files

MATLAB EXPO 2019

### velodyneFileReader

Velodyne PCAP files

### **velodynelidar** Live stream from Velodyne LiDAR®

Live Stream from Sensor

A MathWorks

### **Read Input Sequence from PCAP File**

```
% Read PCAP file recorded from HDL32E model sensor
veloReader =
velodyneFileReader('lidarData.pcap', 'HDL32E');
% Read from file and display
player = pcplayer(xlimits, ylimits, zlimits);
while(hasFrame(veloReader))
    ptCloudObj = readFrame(veloReader);
view(player,ptCloudObj.Location,ptCloudObj.Intensity
);
end
```



# - MathWorks<sup>®</sup> R2019a

# Live Acquisition from Velodyne LiDAR<sup>®</sup> Sensors

- Supported models:
  - Ultra Puck (VLP-32C)
  - Puck (VLP-16)
    - Puck Hi-Res
    - Puck LITE
    - Puck
  - HDL-32E

Stream point clouds to data buffer and read into MATLAB on demand







## **Talk Outline**





### pointCloud Object: Lidar Data Container

pointCloud Object: Primary container for Lidar data



findNearestNeighbors	Find nearest neighbors of a point in point cloud
findNeighborsInRadius	Find neighbors within a radius of a point in the point cloud
findPointsInROI	Find points within a region of interest in the point cloud
removeInvalidPoints	Remove invalid points from point cloud
select	Select points in point cloud

Methods of **pointCloud** object



### **Point Cloud Preprocessing**

### **pcdenoise** Denoise point cloud



#### Original: ~ 34,000 points

#### Denoised: ~ 21,000 points



pcdownsample Down sample point cloud

Reduce # of points without compromising structure of scene

Original: ~ 34,000 points

Down sampled: ~ 14,000 points



# **Point Cloud Registration and Stitching**

Create 3D maps with series of Lidar point clouds

Registration techniques supported:

- Iterative closest point (ICP)
- Coherent point drift (CPD)
- Normal distributions transform (NDT)
  - Popular for automated driving applications







## **Talk Outline**





# **Segmenting Ground Plane in Point Cloud Data**

### pcfitplane

### segmentGroundFromLidarData

### **General Geometric Model Fitting**

- Fit model of plane to points
- More general designed for ALL point clouds (Lidar and RGB-D)

### **Specialized Ground Plane Segmentation**

- Specialized for organized point clouds
- Faster execution

MathWorks<sup>®</sup>

### **Ground Plane Segmentation**

```
% Find the ground points.
                                                         File
groundPtsIdx = segmentGroundFromLidarData(ptCloud);
% Map color ground points to green.
colorLabels(groundPtsIdx (:)) = greenIdx;
% Map color nonground points to red.
colorLabels(~groundPtsIdx (:)) = redIdx;
                                                          Z (m)
```





## Why is it important to detect the ground plane ?

- 1. Helps estimate drivable path:
  - Objects that are NOT ground are obstacles

- 2. Improve processing time
  - Fewer points for downstream processing





## **Talk Outline**





### **Lidar Segmentation**



- Assigns labels to colocated points
- Uses distance metric to cluster points
- No object classification
- Fast execution



### Semantic Segmentation (Deep Learning)

- Deep learning to classify each point
- Longer execution time



# **Point Cloud Segmentation with Clustering**

### pcsegdist

Segments point cloud using Euclidean distance (generic) segmentLidar

- Specialized for Lidar (organized point cloud)
- Real-time execution

distThreshold = 0.5;

```
[labels,numClusters] =
pcsegdist(ptCloudWithoutGround,distThreshold);
```







## **Lidar Segmentation with Clustering**

### pcsegdist

Segments point cloud using Euclidean distance (generic)

distThreshold = 0.5;

```
[labels,numClusters] =
pcsegdist(ptCloud,distThreshold);
```



### segmentLidar

- Specialized for Lidar (organized)
- Real-time execution

distThreshold = 0.5;

```
[labels,numClusters] =
```

segmentLidarData(ptCloud,distThreshold);





### Semantic Segmentation of Lidar using Deep Learning





# **Ground Truth Labeling of Lidar Data**



📣 MathWorks<sup>®</sup>

### **Organize Data for Training**

### **Raw Point Cloud Data**

#### 🕂 32x1085 double

	1	2	3	4
1	NaN	0.1761	0.2727	0.3
2	0.0770	0.1932	0.3064	0.4
3	NaN	NaN	0.2822	0.4
4	0.0834	0.2183	0.2875	0.3
5	NaN	NaN	NaN	N
6	0.0791	0.2409	0.4033	0.5
7	0.0745	0.2363	0.3987	0.5
8	0.0651	0.2266	0.3885	0.5
9	0.0559	0.2177	0.3799	0.5
10	0.0512	0.2130	0.3749	0.5
11	0.0419	0.2037	0.3657	0.5
40	0.0211	0 1014	0 2205	∩ <i>4</i> ′

Project to 2D

### **Ground Truth Labels Transformed to Label Mask**





### **Create Network Architecture**





#### MATLAB EXPO 2019

#### Citation

PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation Charles R. Qi, Hao Su, Kaichun Mo, Leonidas J. Guibas



# Veoneer (Autoliv) Lidar Based Sensor Verification



Labeling of Lidar for verification of Radarbased automated driving system

- MATLAB use in project:
  - Ground truth labeling of Lidar
  - Deep learning for Lidar object detection



#### MATLAB EXPO 2019

LiDAR Based Sensor Verification Nathan Kurtz (Autoliv), Balakumar Ragunathan (Autoliv), Arvind Jayaraman (MathWorks) Video: <u>Automated LiDAR Point-Cloud Annotation for Sensor Verification</u>



## **Talk Outline**





# **Detect Objects in Lidar**

📣 MATLAB R	2019a										3 <b>—</b> 3	۵ ×
HOME	PLOTS	APPS	EDITOR PU	BLISH VIEW				a 🛱 <	5 6 6 0	Search Documentation	tion 🔎	💄 Sign In
New Open	Save FILE	Go To 👻 Co Find 👻 NAVIGATE	Insert 🛃 fx 🕞 Ϛ Comment % ‰ ‰ Indent 💽 📲 🚭 EDIT	Insert Insert Property - Method	<ul> <li>Insert State</li> <li>Input / Output ▼</li> <li>Analyze</li> <li>YSTEM OBJECT</li> </ul>	System Block • Breakpoil	nts Continue	Step	Step In Step Out L Run to Cursor	Function Call Stack: HelperBoundingBoxDe.	Vuit Debugging	Ā
< 🔶 🖬 🕻	🔉 河 📙 🕨 C: 🕨 work	k ▶ u <mark>s</mark> erpath ▶	Examples + R2019a +	driving_fusion_visior	<ul> <li>TrackVehiclesUsin</li> </ul>	gLidarExample 🕨						<b>ب</b>
Current Folder	Workspace	Z Editor - I	HelperBoundingBoxDete	ctor.m 💿 ×	igures - Figure 1							⊛ ×
	(VehiclesUsingLidarExamp	ple.m X Help	perBoundingBoxDetecto	r.m × +	Figure 1 🕱							
48	= methods	Access	s = protecte	d) ^								
49	_ fur	nction []	boxDets, obs	tacl								
50	-	& Crop	point cloud									
51 🔵		[pcSurv	vived, surviv	edIn								
52		% Remov	ve ground pl	ane _								
53 -		[pc0bst	tacles,obsta	cleI1 -						4		* 
54		% Form	clusters an	d ge							40	50
55 -		detBBox	ke <u>ś</u> = getBou	ndin					s and a second s	30	40	
56		& Asser	nbl det BBoxes	: 6×10 sind	gle matrix	=	313			20		
57 – 🗭		bboxDet	ts					7 -		10		
58 -	- end	d.	Columns	1 through	4		3	1	X-Cente	er		
59	- end			-			-2		Y-Cente	er		
60	end		12.892	-22.675	-46.8280	-21.1414	0		7 Conto	or		
61 K	_		-3.914	8 -3.723	3 -3.5872	0.0260						
Command W	indow		0.729	0.696	6 -0.6705	0.7558			Length			
K>> 1	pcshow (pc0bst	tacles)	2.874	7 2.6390	0.0816	2.2517			Width			
<i>fx</i> K>>	-		1.751	.0 1.739	0.8562	1.6446	5		Height			
<			1.083	0.591	0.0068	0.5503	3					
IIII + 2 usages	of "bboxDets" found						4		HelperBoun	ndingBoxDetector / steph	mpl Ln 57	Col 13
EXPO 2	2019											



# **Detect Objects in Lidar**

📣 MATLAB R	2019a										3 <b>—</b> 3	۵ ×
HOME	PLOTS	APPS	EDITOR PU	BLISH VIEW				a 🛱 <	5 6 6 0	Search Documentation	tion 🔎	💄 Sign In
New Open	Save FILE	Go To 👻 Co Find 👻 NAVIGATE	Insert 🛃 fx 🕞 Ϛ Comment % ‰ ‰ Indent 💽 📲 🚭 EDIT	Insert Insert Property - Method	<ul> <li>Insert State</li> <li>Input / Output ▼</li> <li>Analyze</li> <li>YSTEM OBJECT</li> </ul>	System Block • Breakpoil	nts Continue	Step	Step In Step Out L Run to Cursor	Function Call Stack: HelperBoundingBoxDe.	Vuit Debugging	Ā
< 🔶 🖬 🕻	🔉 河 📙 🕨 C: 🕨 work	k ▶ u <mark>s</mark> erpath ▶	Examples + R2019a +	driving_fusion_visior	<ul> <li>TrackVehiclesUsin</li> </ul>	gLidarExample 🕨						<b>ب</b>
Current Folder	Workspace	Z Editor - I	HelperBoundingBoxDete	ctor.m 💿 ×	igures - Figure 1							⊛ ×
	(VehiclesUsingLidarExamp	ple.m X Help	perBoundingBoxDetecto	r.m × +	Figure 1 🕱							
48	= methods	(Access	s = protecte	d) ^								
49	_ fur	nction []	boxDets, obs	tacl								
50		& Crop	point cloud									
51 🔵		[pcSurv	vived, surviv	edIn								
52		% Remov	ve ground pl	ane _								
53 -		[pc0bst	tacles,obsta	cleI1 -						4		* 
54		% Form	clusters an	d ge <sup>.</sup>							40	50
55 -		detBBox	ke <u>ś</u> = getBou	ndin					s and a second sec	30	40	
56		& Asser	nbl det BBoxes	: 6×10 sind	gle matrix	=	313			20		
57 – 🗭		bboxDet	ts					7 -		10		
58 -	- end	d.	Columns	1 through	4		3	1	X-Cente	er		
59	- end			-			-2		Y-Cente	er		
60	end		12.892	-22.675	-46.8280	-21.1414	0		7 Conto	or		
61 K	_		-3.914	8 -3.723	3 -3.5872	0.0260						
Command W	indow		0.729	0.696	6 -0.6705	0.7558			Length			
K>> 1	pcshow (pc0bst	tacles)	2.874	7 2.6390	0.0816	2.2517			Width			
<i>fx</i> K>>	-		1.751	.0 1.739	0.8562	1.6446	5		Height			
<			1.083	0.591	0.0068	0.5503	3					
IIII + 2 usages	of "bboxDets" found						4		HelperBoun	ndingBoxDetector / steph	mpl Ln 57	Col 13
EXPO 2	2019											



### **Design Tracker for Lidar Point Cloud**





### **Design Tracker for Lidar Point Cloud**





### Talk Outline



- C++ code generation for lidar detection and tracking
- CUDA code generation for deep learning



### **Generate C/C++ Code for Lidar Detector and Tracker**

		· · · · · · · · · · · · · · · · · · ·			
MAILAB Coder Report Viewer - C:\work	k\userpath\Examples\R2019a\drivi	ng_tusion_vision\TrackVehiclesUsingLidar	Example\codegen\lib\mexLidarTracker\html	l\report.mldatx	- D X
REPORT					
4 4 Go To + 3	A 🗎 🖻				
Back Forward Q Find Trace Ed Code MAT	It In Package Export Report TLAB Code Information				
NAVIGATE TRACE EE	DIT SHARE				Ā
MATLAB SOURCE	Code				0
Function List Call Tree	mexLidarTracker.m	mexLidar	Tracker.m [10-17] mexLidarTrack -		mexLidarTracker.cpp
🖻 👰 mexLidarTracker.m 🔹	+ persistent detector	model tracker detectablemat	ksinput curren	356 //	% measurement noise in detection *
fx mexLidarTracker	5		- 1	357 //	'mexLidarTracker:10' detectorModel = He
🗖 🏟 HelperBoundingBoxDetector.r	7 if isompty(detector	nModel)    isemntv(tracker)	l isempty/dete	358 //	'mexLidarTracker:11'
fr HelperBoundingBoxDetect	g II ISempty (detector	(hoder)    isempty(tracker)	1 Isempty (dete	359 //	'mexLidarTracker:12'
fu assembleDetections	9 % A bounding b	nx detector model		360 //	'mexLidarTracker:13'
fx assembleDetections	10 detectorModel	= HelperBoundingBoxDetector(		361 //	'mexLidarTracker:14'
fx cropPointCloud	11	'XLimits'.[-50 75]	% min	362 //	'mexLidarTracker:15'
<i>fx</i> getBoundingBoxes	12	'YLimits'.[-5 5]	% min	363 //	'mexLidarTracker:16'
fx removeGroundPlane	13	'ZLimits',[-2 5],	% min	364 //	'mexLidarTracker:17'
fx stepImpl	14	'SegmentationMinDistance',1	1.6, % min	365 de	tectorModel.GroundReferenceVector[0] = (
helperCalcDetectability.m	15	'MinDetectionsPerCluster',1	% min	366 de	tectorModel.GroundReferenceVector[1] = [
fr helperCalcDetectability	16	'MeasurementNoise',eye(6),.	% mea	367 de	tectorModel.GroundReferenceVector[2] =
• •	17	'GroundMaxDistance',0.3);	% max	368 de	tectorModel.GroundMaxAngularDistance =
<ul> <li>GENERATED CODE</li> </ul>	18			369 de	tectorModel.Max2Distancecluster = 3.0;
🖻 💼 Source Files 🔹	19 assignmentGate	= [10 100]; % Assignment thr	reshold;	370 de	tectorModel.MinZDistanceciuster =
++ AssignmentCostCalculator	20 confThreshold	= [7 10]; % Confirmation t	threshold for h	371 de	toctorModel icInitialized 0
D AssignmentCostCalculator	21 delThreshold =	[8 10]; % Deletion three	shold for histo	272 ue	ctrl -
EvtendedValmanEilter opn	22 Kr = 1e-5	% False-alarm ra	ate ner unit vo	374 4	
ExtendedKalmanFilter.h	SUMMARY	ALL MESSAGES (0)	BUILD LOGS	CODE INSIGHTS	S (0) VARIABLES
++ HelperBoundingBoxDetect					
D HelperBoundingBoyDetect	🔗 Codo gonor	ation successful			
VEDistance and	Coue gener	ation succession			
THE REDIStance.cpp	Generated on: 24-	Mar-2019 14:59:22			
B KEDIstance.n	Build type: Sta	tic Library			
++ KalmanLikelihood.cpp	Toolchain: Mid	crosoft Visual C++ 2017 v15 0   pmake	(64-bit Windows)		
KalmanLikelihood.h	Build Configuration: East	stor Builde	(01.00.000)		
- ++ ObjectTrack.cpp -	a congulation. Ta				
×					,
14					<u>×</u>



### **Deep Learning Model Deployment**





# **Customer Success Using MATLAB for Lidar Processing**

- Lidar sensor design

Radar verification with Lidar
 Automated 90% of data analysis

Perception with Lidar

"We've used both Python and MATLAB to work with lidar sensor data, and I estimate that **analysis and development was one-and-a-half to two times faster in MATLAB**," MATLAB EXPO 2019





### **Reference Examples on Lidar Processing**

#### Lidar Processing



#### Ground Plane and Obstacle Detection Using Lidar

Detect the ground plane and find nearby obstacles in 3-D lidar data.



#### Track Vehicles Using Lidar: From Point Cloud to Track List

Track vehicles from lidar data using a joint probabilistic data association (JPDA) tracker and an interacting multiple model (IMM) approach.

https://www.mathworks.com/help/driving/examples.html



# **Summary: Lidar Processing for Automated Driving**

